



Open Tools from Sybase, Inc.

PowerBuilder

Objects and Controls

Version 6

Power Builder®

AA0794

October 1997

Copyright © 1991-1997 Sybase, Inc. and its subsidiaries.

All rights reserved.

Printed in Ireland.

Information in this manual may change without notice and does not represent a commitment on the part of Sybase, Inc. and its subsidiaries.

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.

No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc. and its subsidiaries.

Sybase, Inc. and its subsidiaries claim copyright in this program and documentation as an unpublished work, revisions of which were first licensed on the date indicated in the foregoing notice.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 other rights of Sybase, Inc. and its subsidiaries.

ClearConnect, Column Design, ComponentPack, InfoMaker, ObjectCycle, PowerBuilder, PowerDesigner, Powersoft, S-Designor, SQL SMART, and Sybase are registered trademarks of Sybase, Inc. and its subsidiaries. Adaptive Component Architecture, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Warehouse, AppModeler, DataArchitect, DataExpress, Data Pipeline, DataWindow, dbQueue, ImpactNow, InstaHelp, Jaguar CTS, jConnect for JDBC, MetaWorks, NetImpact, Optima++, Power++, PowerAMC, PowerBuilder Foundation Class Library, Power J, PowerScript, PowerSite, Powersoft Portfolio, Powersoft Professional, PowerTips, ProcessAnalyst, Runtime Kit for Unicode, SQL Anywhere, The Model For Client/Server Solutions, The Future Is Wide Open, Translation Toolkit, UNIBOM, Unilib, Uninull, Unisep, Unistring, Viewer, WarehouseArchitect, Watcom, Watcom SQL Server, Web.PB, and Web.SQL are trademarks of Sybase, Inc. or its subsidiaries. Certified PowerBuilder Developer and CPD are service marks of Sybase, Inc. or its subsidiaries. DataWindow is a patented proprietary technology of Sybase, Inc. or its subsidiaries.

AccuFonts is a trademark of AccuWare Business Solutions Ltd.

All other trademarks are the property of their respective owners.

Contents

About This Book	xiii	
1	PowerBuilder System Objects and Controls	1
	What are system objects?	2
	System object properties, events, and functions	3
	System object inheritance hierarchy	4
	Base class objects not in this book	4
	Viewing system objects	5
2	System Object Properties, Events, and Functions	7
	Application object	8
	ArrayBounds object	12
	CheckBox control	13
	ClassDefinition object	20
	CommandButton control	24
	Connection	30
	ConnectionInfo	36
	ContextInformation	38
	ContextKeyword	40
	CPlusPlus object	42
	DataStore object	44
	DataWindow control	56
	DataWindowChild	82
	DropDownListBox	94
	DropDownPictureListBox	104
	DynamicDescriptionArea	115
	DynamicStagingArea	119
	EditMask control	121
	EnumerationDefinition object	131
	EnumerationItemDefinition object	133
	Environment object	134
	Error object	136
	Graph object	138

grAxis object.....	147
grDispAttr object.....	152
GroupBox control	156
HScrollBar control	161
Inet	166
InternetResult.....	168
Line control.....	170
ListBox control.....	172
ListView control.....	181
ListViewItem object	195
mailFileDescription.....	197
mailMessage	199
mailRecipient.....	201
mailSession.....	203
MDIClientobject.....	206
Menu	209
MenuCascade	214
Message.....	219
MultiLineEdit.....	222
OLE control	231
OLE custom control (OCX)	240
OLEObject.....	249
OLEStorage	251
OLEStream	253
Oval control	255
Picture control	258
PictureButton control.....	263
PictureListBox control	270
Pipeline object.....	282
ProfileCall object	284
ProfileClass object	286
ProfileLine object.....	287
ProfileRoutine object.....	289
Profiling object.....	292
RadioButton control.....	294
Rectangle control	300
RichTextEdit control	303
RoundRectangle control.....	318
ScriptDefinition object	321
SimpleTypeDefinition object.....	325
SingleLineEdit control	326
StaticText control	334
Tab control	341
Timing	352

TraceActivityNode object	354
TraceBeginEnd object	356
TraceError object	358
TraceESQL object	360
TraceFile object	362
TraceGarbageCollect object	364
TraceLine object	366
TraceObject object	367
TraceRoutine object	369
TraceTree object	371
TraceTreeError object	373
TraceTreeESQL object	375
TraceTreeGarbageCollect object	376
TraceTreeLine object	378
TraceTreeNode object	379
TraceTreeObject object	381
TraceTreeRoutine object	383
TraceTreeUser object	385
TraceUser object	387
Transaction object	389
Transport	392
TreeView control	396
TreeViewItem object	409
TypeDefinition object	412
UserObject	414
VariableCardinalityDefinition object	423
VariableDefinition object	425
VScrollBar control	428
Window object	433

About This Book

Subject

This book lists the properties, events, and related functions for PowerBuilder system objects and controls.

FOR INFO For complete information on the objects and controls you can use in PowerBuilder applications, see the *PowerBuilder User's Guide*.

Audience

This book is for programmers using PowerBuilder to create and maintain applications and is to be used when you are writing scripts.

PowerBuilder System Objects and Controls

About this chapter

This chapter provides overview information about PowerBuilder system objects and controls. This chapter also lists the PowerBuilder system objects not included in this book and explains why they are not.

Contents

Topic	Page
What are system objects?	2
System object properties, events, and functions	3
System object inheritance hierarchy	4
Viewing system objects	5

What are system objects?

- System objects** PowerBuilder **system class objects** are the built-in objects you use to develop your application. PowerBuilder system objects include objects such as windows and menus, as well as graphical controls and predefined entities that you can reference in your application, such as the Message and Error objects.
- Controls** PowerBuilder **controls** are a subset of system objects that you place in windows or user objects. Typically, they are graphical objects that allow users to interact with your application or that you use to enhance the design of your windows.
- System structures** PowerBuilder system **structures** are a subset of system objects that contain collections of variables related to other system objects. For example, the Environment object is a structure that holds information about the computing platform the PowerBuilder Application object is running on.

System object properties, events, and functions

Properties	Each system object has a number of properties associated with it that define its characteristics. For example, the CheckBox control has Height and Width properties that control its size and a BackColor property that controls its background color. You can set the value of object properties within scripts or with the object's Property sheets available within the painters.
Events	<p>PowerBuilder applications are event-driven. For example, when a user clicks a button, chooses an item from a menu, or enters data into an edit box, an event is triggered. You write scripts using PowerScript, the PowerBuilder language, that specify the processing that should happen when the event is triggered. PowerBuilder passes arguments to events, such as the coordinates of the pointer, that help your application figure out what the user did to trigger the event. For most events, you can specify a return code to affect what happens next, such as triggering another event.</p> <p>Controls, with the exception of the GroupBox and the drawing objects (Line, Oval, Rectangle, and RoundRectangle), always have events related to them. Some system objects, such as system structures, do not have any events associated with them.</p>
Functions	PowerScript provides a rich assortment of built-in functions you can use to act upon the objects and controls in your application. For each system object, there is a set of these built-in functions that can act on it. You use these functions in scripts to manipulate the object.

System object inheritance hierarchy

Inheritance	One of the most powerful features of PowerBuilder is inheritance . It enables you to build windows, user objects, and menus that are derived from existing objects. When you build an object that inherits from another object, you create a hierarchy (or tree structure) of ancestor and descendent objects.
Base class object	The object at the top of the hierarchy is a base class object , and the other objects are descendants of this object. Each descendant inherits information from its ancestor. The base class object typically performs generalized processing, and each descendant modifies the inherited processing as needed.
System object hierarchy	The PowerBuilder system objects comprise such a hierarchy. At the top of the hierarchy is the PowerObject , the base class from which all the objects and controls described in this book are descended. The hierarchy also contains other (generic) base class objects that are not typically used in application development but are necessary parts of the logical hierarchy organization.

Base class objects not in this book

This book does not describe base class objects whose primary function is to provide generic properties and functions for descendent objects, since these objects typically are not used in applications. The base class system objects not described in this book are:

- DragObject
- DrawObject
- DWObject
- ExtObject
- Function_Object
- GraphicObject
- NonVisualObject
- OmControl
- OmCustomControl
- OmEmbeddedControl
- OmObject
- OmStorage
- OmStream
- PowerObject
- Structure
- WindowObject

Viewing system objects

Using the Browser

From within PowerBuilder, you can use the PowerBuilder Browser to see a complete list of system objects and their properties, events, and functions.

- ◆ To display the *system objects*, select the System tab of the Browser. The default display is to list the objects alphabetically.
- ◆ To see the *objects displayed hierarchically*, place the cursor in the left pane, press the right mouse button, and select Show Hierarchy.
- ◆ To display a *specific object's properties, events, or functions*, select the object in the left pane and then double-click the Properties, Events, or Functions item in the right pane.

FOR INFO For information about using the PowerBuilder Browser, see the *PowerBuilder User's Guide*.

Using online Help

You can also use PowerBuilder online Help to view more descriptive Help topics about the properties, events, and functions for system objects and controls.

- ◆ If you know the name of the system object or control, use the Search dialog to go directly to the correct topic.
- ◆ To see a list of the system objects for which Help topics exist, select System objects from the Help contents list.
- ◆ To see a list of the system controls for which Help topics exist, select Controls from the Help contents list.

CHAPTER 2

System Object Properties, Events, and Functions

About this chapter

This chapter lists the properties, events, and functions in the definition of PowerBuilder objects and controls. This chapter does not include the base class objects (listed in "Base class objects not in this book" on page 4).

Contents

The objects and controls are listed alphabetically.

Application object

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.

Properties

Application property	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). Platform note This and other DDE functions are available on the Windows platform. On UNIX platforms this and other DDE functions have effect only if the server and client applications are developed using PowerBuilder or compiled using Wind/U from Bristol Technology.
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.

Application property	Data type	Description
DWMessageTitle	String	Specifies the title of the DataApplication message boxes that display during execution. The value of this property is recognized only at the Open event of the Application object.
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.
RightToLeft	Boolean	Specifies that characters should be displayed in right-to-left order in MessageBoxes displayed when you call the MessageBox function. The application must be running on a Hebrew or Arabic version of PowerBuilder under an operating system that supports right-to-left display. Values are: <ul style="list-style-type: none"> ◆ True — Message box text will display in right-to-left order. The text of the MessageBox buttons will only display in the language of the RightToLeft version of Windows (Arabic or Hebrew) if you are running a localized version of PowerBuilder. Otherwise, the text of the MessageBox buttons will display in English. ◆ False — Characters display in left-to-right order
ToolbarFrameTitle	String	Specifies the text that displays as the title for the FrameBar when it is floating.
ToolbarPopupMenuText	String	Allows you to change the toolbar location text (Left, Top, Right, Bottom, Floating) in the Application's toolbar popup menu. Specify the text as a comma-separated list of items.
ToolbarSheetTitle	String	Specifies the text that displays as the title for the SheetBar when it is floating.

Application property	Data type	Description
ToolbarText	Boolean	Specifies whether the text associated with the items in the toolbar displays. Values are: <ul style="list-style-type: none"> ◆ True — Text displays in toolbar bar ◆ False — Text does not display in toolbar
ToolbarTips	Boolean	Specifies whether PowerTips display when text is not displayed on the buttons. Values are: <ul style="list-style-type: none"> ◆ True — Power Tips are displayed ◆ False — Power Tips are not displayed
ToolbarUserControl	Boolean	Specifies whether users can use the toolbar popup menu to hide or show the toolbars, move toolbars, or show text. Values are: <ul style="list-style-type: none"> ◆ True — Users can use popup menu ◆ False — Users cannot use popup menu

Events

Application event	Occurs
Close	When the user closes the application.
ConnectionBegin	When a client application attempts to establish a connection to a server application. This event is triggered only in a server application running in a distributed computing environment. The event returns the connect privilege status of the client.
ConnectionEnd	When a client application's connection is terminated. This event is triggered only in a server application running in a distributed computing environment.
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.

Application event	Occurs
Open	When the user runs the application.
SystemError	<p>When a serious execution time error occurs (such as trying to open a nonexistent application). If there is no script for this event, PowerBuilder displays a message box with the PowerBuilder error number and error message text.</p> <p>For information about error messages, see the <i>PowerBuilder User's Guide</i>.</p>

Functions

Application function	Data type returned	Description
ClassName	String	Returns the class of the Application object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent 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.
SetTransPool	Integer	Sets up a pool of database transactions for an application. SetTransPool allows you to minimize the overhead associated with database connections and also limit the total number of database connections permitted.
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.

ArrayBounds object

A structure that specifies the upper and lower bounds of a single dimension of an array. It is used in the VariableCardinalityDefinition object. ArrayBounds has no events.

Properties

ArrayBounds property	Data type	Description
ClassDefinition	PowerObject	An object of type ClassDefinition describing the ArrayBounds class.
LowerBound	Long	The lower bound of the array dimension. For unbounded arrays, the value is always 0.
UpperBound	Long	The upper bound of the array dimension. For unbounded arrays, the value is always 0.

Functions

ArrayBounds function	Data type returned	Description
ClassName	String	Returns the class of the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object	Returns the type of the object.

CheckBox control

Checkboxes are small square boxes used to set independent options. When they are selected, they display a mark (typically, either an X or a checkmark). 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 filled box.

Properties

CheckBox property	Data type	Description
Automatic	Boolean	Specifies whether the control displays a mark when the user clicks it. Values are: <ul style="list-style-type: none"> ◆ True — Displays mark when clicked ◆ False — Does not display mark when clicked
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215. For more information about color, see the <i>PowerScript Reference</i> .
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order. Values are: <ul style="list-style-type: none"> ◆ True — Move to the top ◆ False — Do not move to the top

CheckBox property	Data type	Description
Checked	Boolean	<p>Specifies whether the control is selected. Values are:</p> <ul style="list-style-type: none"> ◆ True — Control is selected ◆ False — Control is not selected
DragAuto	Boolean	<p>Specifies whether PowerBuilder puts the control automatically into Drag Mode. Values are:</p> <ul style="list-style-type: none"> ◆ 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	<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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be clicked). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control can be clicked ◆ False — Control cannot be clicked
FaceName	String	<p>Specifies the name of the typeface in which the text of the control displays; for example, ARIAL or COURIER.</p>

CheckBox property	Data type	Description
FontCharSet	FontCharSet (enumerated)	Specifies the font character set used for the text in the control. The application must be running on an appropriate version of PowerBuilder under an operating system that supports the selected character set. Values are: ANSI! ChineseBig5! DefaultCharSet! Hangeul! OEM! ShiftJIS! Symbol!
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) used for the text in the control. Values are: AnyFont! Decorative! Modern! Roman! Script! Swiss!
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control. Values are: Default! Fixed! Variable!
Height	Integer	Specifies the height of the control, in PowerBuilder units.
Italic	Boolean	Specifies whether the text in the control is italic. Values are: ◆ True — Text is italic ◆ False — Text is not italic
LeftText	Boolean	Specifies whether the text displays on the left of the control. Values are: ◆ True — Text displays on left ◆ False — Text displays on right Typically, you would set this property to False so the text appears on the right of the control.

CheckBox property	Data type	Description
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
RightToLeft	Boolean	Specifies that characters should be displayed in right-to-left order. The application must be running on a Hebrew or Arabic version of PowerBuilder under an operating system that supports right-to-left display. Values are: <ul style="list-style-type: none"> ◆ True — Characters display in right-to-left order ◆ False — Characters display in left-to-right order
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 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>PowerScript 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 when the control has been defined to have three states. Values are: <ul style="list-style-type: none"> ◆ True — Control is in third state ◆ False — Control is not in third state

CheckBox property	Data type	Description
ThreeState	Boolean	Specifies whether the control has three states. Typically, CheckBox controls have only two states, such as on and off. Values are: <ul style="list-style-type: none"> ◆ True — Control has three states ◆ False — Control does not have three states
Underline	Boolean	Specifies whether the text in the control is underlined. Values are: <ul style="list-style-type: none"> ◆ True — Text is underlined ◆ False — Text is not underlined
Visible	Boolean	Specifies whether the control is visible. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
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

CheckBox 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.

CheckBox event	Occurs
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

CheckBox function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
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.

CheckBox function	Data type returned	Description
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 Properties.
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.

ClassDefinition object

A ClassDefinition object is a PowerBuilder object that provides information about the class definition of a PowerBuilder object. You can examine a class in a PowerBuilder library or the class associated with an instantiated object.

All the properties are read-only. You cannot change the class definition via the ClassDefinition object. The ClassDefinition object has no events.

The ClassDefinition object lets you check:

- ◆ The name of the class
- ◆ The library the class was loaded from
- ◆ The class definition of its ancestor, if any
- ◆ The class definition of its parent or container object, if any
- ◆ Whether the class is autoinstantiated
- ◆ Whether the class is a system class (defined by PowerBuilder) or a user-defined object (defined in a PowerBuilder PBL)
- ◆ The classes the object contains, such as the controls contained in a window
- ◆ The variables and scripts defined in the class

Class names are always reported as lowercase, as you see them in the Browser.

Global functions Call FindFunctionDefinition to get a ScriptDefinition object describing the global function.

Global variables Global variables are included in the VariableList array in the ClassDefinition object for the Application object.

Properties

ClassDefinition property	Data type	Description
Ancestor	ClassDefinition	An object that represents the ancestor class. Ancestor is NULL when the ClassDefinition is describing PowerObject.

ClassDefinition property	Data type	Description
Category	TypeCategory	Specifies if the type is simple, enumerated, or a class or structure. For a class definition, the value is ClassOrStructureType!.
ClassDefinition	PowerObject	An object of type ClassDefinition describing the ClassDefinition class.
DataTypeOf	String	The system class name of the object. DataTypeOf is a string representation of a value of the Object enumerated data type. Values are lowercase with no exclamation point. Sample values include: window string any dropdownlistbox For objects you have defined, the data type is the system class from which your object is inherited.
IsAutoinstantiate	Boolean	Indicates whether the class is an autoinstantiated class.
IsStructure	Boolean	Indicates whether the class is a structure.
IsSystemType	Boolean	Indicates whether the class is a system class, that is one of the classes defined within PowerBuilder as opposed to a class defined in a PBL by a user.
IsVariableLength	Boolean	Specifies whether the data type has a fixed size. Values are: <ul style="list-style-type: none"> ◆ True — The data type is variable length, meaning the data type is a string, any, blob, or unbounded array. ◆ False — The data type is a fixed length.

ClassDefinition property	Data type	Description
IsVisualType	Boolean	Indicates whether the class is a visual (displayable) or non-visual type. Values are: <ul style="list-style-type: none"> ◆ True — The class is visual, for example, a window or a control ◆ False — The class is non-visual, for example, a class user object or a simple data type.
LibraryName	String	The fully qualified name of the library the class was loaded from. Note that the library may no longer contain the class. If a program manipulates the contents of libraries, the class could have been moved or deleted after it was loaded.
Name	String	The name of the class. For a nested class, the name will be returned in the form of <i>libraryEntryName`className</i>
NestedClassList[]	ClassDefinition	An unbounded array of objects representing the nested classes and local structures for the object. The array is empty if there are no nested classes. Call the UpperBound function to find out the number of nested classes.
ParentClass	ClassDefinition	An object that represents the parent class that this class is nested within. The value is NULL if the class is not a nested class.
ScriptList[]	ScriptDefinition	An unbounded array of objects representing the scripts implemented or defined in the collapsed class hierarchy. The array will be empty if there are no scripts. Call the UpperBound function to find out the number of scripts.

ClassDefinition property	Data type	Description
VariableList[]	Variable Definition	An unbounded array of objects representing the properties or shared variables in the collapsed class hierarchy. The array will be empty if there are no variables. Call the UpperBound function to find out the number of variables.

Functions

ClassDefinition function	Data type returned	Description
ClassName	String	Returns the class of the object.
FindMatching Function	ScriptDefinition	Finds a function that matches the specified name and argument list.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object	Returns the type of the object.

CommandButton control

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.

Properties

Command Button property	Data type	Description
BringToTop	Boolean	<p>Specifies whether PowerBuilder moves the control to the top of the front-to-back order of the window. Values are:</p> <ul style="list-style-type: none"> ◆ True — Move to the top ◆ False — Do not move to the top
Cancel	Boolean	<p>Specifies whether the control acts as the Cancel button. (The Cancel button receives a Clicked event if the user presses ESC.) Values are:</p> <ul style="list-style-type: none"> ◆ True — Acts as the Cancel button ◆ False — Does not act as the Cancel button
Default	Boolean	<p>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. Values are:</p> <ul style="list-style-type: none"> ◆ True — Acts as the default ◆ False — Does not act as the default <hr/> <p>Editable controls Default behavior can be affected by editable controls on the window. For more information, see the <i>PowerBuilder User's Guide</i>.</p>

Command Button property	Data type	Description
DragAuto	Boolean	<p>Specifies whether PowerBuilder puts the control automatically into Drag Mode. Values are:</p> <ul style="list-style-type: none"> ◆ 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	<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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be clicked). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control is enabled ◆ False — Control is not enabled
FaceName	String	<p>Specifies the name of the typeface in which the text of the control displays (for example, ARIAL or COURIER).</p>

Command Button property	Data type	Description
FontCharSet	FontCharSet (enumerated)	Specifies the font character set used for the text in the control. The application must be running on an appropriate version of PowerBuilder under an operating system that supports the selected character set. Values are: ANSI! ChineseBig5! DefaultCharSet! Hangeul! OEM! ShiftJIS! Symbol!
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) used for the text in the control. Values are: AnyFont! Decorative! Modern! Roman! Script! Swiss!
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control. Values are: Default! Fixed! Variable!
Height	Integer	Specifies the height of the control, in PowerBuilder units.
Italic	Boolean	Specifies whether the text in the control is italic. Values are: ◆ True — Text is italic ◆ False — Text is not italic
Pointer	String	Specifies the name of the stock pointer of the file containing the pointer that is used for the control.

Command Button property	Data type	Description
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. Values are: <ul style="list-style-type: none"> ◆ True — Text is underlined ◆ False — Text is not underlined
Visible	Boolean	Specifies whether the control is visible. <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
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

CommandButton event	Occurs
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

CommandButton function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
Hide	Integer	Makes the control invisible.

CommandButton function	Data type returned	Description
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 properties.
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.

Connection

The Connection object specifies the parameters that PowerBuilder uses to connect to a server application.

You can customize the Connection object by defining a class user object inherited from the built-in Connection object. The user object has three events: Constructor, Destructor, and Error.

FOR INFO For more information about creating a custom Connection object, see the chapter on user objects in the *PowerBuilder User's Guide*.

Properties

Connection property	Data type	Description
Application	String	<p>Specifies the server application that a client application wants to connect to.</p> <p>For the WinSock driver Specifies either of the following:</p> <ul style="list-style-type: none"> ◆ The port number for the application (for example, 10099). Each server application requires a unique port number on the server machine. If you specify a port number, select a number that is greater than 4096. ◆ The service name for the application. The service name is an indirect reference to the application's port number. The mapping of the service name to the port number is specified in the TCP/IP services file. <p>For the NamedPipes driver Specifies the application portion of the pipe name. The combination of the Location and Application values forms the pipe name. The pipe name is constructed as follows:</p> <p style="padding-left: 40px;"><i>\\location\PIPE\application</i></p> <p>The Application must be unique for the Location you specify.</p> <p>For the Local driver This property is ignored.</p>

Connection property	Data type	Description
ConnectString	String	Text passed to the ConnectionBegin event of the server's Application object at connection time. The text can include application-specific information, such as database connection parameters. PowerBuilder does not evaluate this string.
Driver	String	The communications driver that will be used for the connection. Values are: WinSock NamedPipes Local
ErrCode	Long	Code indicating the success or failure of the most recent operation. Values are: <ul style="list-style-type: none"> ◆ 0 — Success ◆ 50 — Distributed service error ◆ 52 — Distributed communications error ◆ 53 — Requested server not active ◆ 54 — Server not accepting requests ◆ 55 — Request terminated abnormally ◆ 56 — Response to request incomplete ◆ 57 — Not connected ◆ 58 — Object instance does not exist ◆ 62 — Server busy ◆ 75 — Server forced client to disconnect ◆ 80 — Server timed out client connection ◆ 87 — Connection to server has been lost
ErrText	String	Text indicating the success or failure of the most recent operation.

Connection property	Data type	Description
Location	String	<p>Specifies the location of the server application.</p> <p>For the WinSock driver Specifies one of the following values:</p> <ul style="list-style-type: none"> ◆ The host name for the server (as specified in the TCP/IP hosts file) ◆ The IP address (for example, 199.99.99.91) ◆ LocalHost (which tells PowerBuilder that the server application resides on the local machine) <p>For the NamedPipes driver Specifies the location portion of the pipe name. The combination of the Location and Application values forms the pipe name. The pipe name is constructed as follows:</p> <p style="padding-left: 40px;"><i>\\location\PIPE\application</i></p> <p>If no location is specified, a local pipe name is constructed using a dot (.) in the machine name portion.</p> <p>For the Local driver This property is ignored.</p>
Options	String	<p>Specifies one or more communications options. If you specify more than one option, you need to separate the options with commas.</p> <p>For the Local drivers This property is ignored.</p> <p>For a description of each of the options, see "Options" next.</p>
Password	String	<p>The password that will be used to connect to the server. This value is application-specific and is not evaluated by PowerBuilder.</p>
Trace	String	<p>Specifies one or more trace options. The trace options are used to troubleshoot client/server connections. If you specify more than one trace option, you need to separate the options with commas.</p> <p>For a description of each of the trace options, see "Trace options" on page 33.</p>
UserID	String	<p>The name or ID of the user who will connect to the server. This value is application-specific and is not evaluated by PowerBuilder.</p>

Options

Option	Description
BufSize= <i>n</i>	Sets the connection buffer size to the value specified.
MaxRetry= <i>n</i>	Specifies how many times the client application will try to connect when the server application's listening port is busy. Applies to the WinSock driver only.
NoDelay=1	Specifies that each packet be sent without delay. Corresponds to the TCP_NODELAY option. Setting this option may degrade performance significantly. Do not use this option unless you thoroughly understand its implications. Applies to the WinSock driver only.
RawData=1	Specifies that raw data be passed over the network. By default, the WinSock driver obscures the data that is passed over the network. Setting this option to 1 overrides the default behavior. Both the client and server applications must have the same setting. If there is a discrepancy between the client and server settings, the communication fails. Setting this option to 1 may improve performance slightly. Applies to the WinSock driver only.
TimeOut= <i>n</i>	Specifies the amount of time PowerBuilder will wait for a reply from a synchronous request. After the specified time elapses, an error is returned.

Trace options

Trace option	Description
All=1	Logs all activity between the client and the server. This option logs internal information that can be helpful in debugging, including memory usage, an internal call trace, and the types and values for passed parameters
Console=1	Logs activity to a console window (not supported on Windows 3.1).
DumpArgs=1	Logs parameter types and values
Level=1	Enables Console=1, ObjectCalls=1, and ObjectLife=1.

Trace option	Description
Log= <i>filename</i>	Logs activity to the specified file. If the client application uses multiple connection objects, you need to specify a unique log file name for each connection object.
ObjectCalls=1	Logs each object method call and indicates whether the call was successful.
ObjectLife=1	Logs each attempt to create or destroy a remote object and indicates whether the operation was successful.

Events

Connection event	Occurs
Constructor	When the connection object is created.
Destructor	When the connection object is destroyed.
Error	When a client request cannot be satisfied.

Functions

Connection function	Data type returned	Description
ClassName	String	Returns the class of the object.
ConnectToServer	Long	Connects a client application to a server application.
CreateInstance	Long	Creates an instance of a remote object on a server.
DisconnectServer	Long	Disconnects a client application from a server application.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.

Connection function	Data type returned	Description
GetServerInfo	Long	Allows a client application to retrieve information about its connection to a server.
PostEvent	Boolean	Adds an event to the end of the message queue for the object.
RemoteStopConnection	Long	Allows a client application to disconnect another client from a server application.
RemoteStopListening	Long	Allows a client application to instruct a server application to stop listening for client requests.
TriggerEvent	Integer	Triggers a specified event in the object and executes the script for the event.
TypeOf	Object	Returns the type of the object.

ConnectionInfo

The ConnectionInfo object is a structure used to hold information about client connections to a server application. You populate the ConnectionInfo structure by using the GetServerInfo function. The ConnectionInfo object has no events.

Properties

ConnectionInfo property	Data type	Description
Busy	Boolean	Specifies whether the client connection is busy: <ul style="list-style-type: none"> ◆ True — The connection is busy ◆ False — The connection is not busy
CallCount	Long	Total number of calls made by the client.
ClientID	String	The internal ID of the client that made the connection.
ConnectionString	String	The value assigned to the ConnectString property of the Connection object.
ConnectTime	DateTime	The date and time that the client connected to the server.
ConnectUserID	String	The value assigned to the UserID property of the Connection object
LastCallTime	DateTime	The date and time of the last client request.
Location	String	The name of the machine where the client resides.
UserID	String	The operating system user ID of the client that made the connection.

Functions

ConnectionInfo function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.

ConnectionInfo function	Data type returned	Description
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object	Returns the type of the object.

ContextInformation

The ContextInformation object provides information about an application's execution context, including current version information as well as whether it's running in the PowerBuilder execution environment as a Window Plug-in or as a Window ActiveX. Using this information, you can modify display characteristics and application behavior.

Properties

ContextInformation property	Data type	Description
ClassDefinition	PowerObject	An object of type ClassDefinition describing the ContextInformation class.

Events

ContextInformation event	Occurs
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.

Functions

ContextInformation function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetCompanyName	Integer	Returns the company name for the current execution context.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.

ContextInformation function	Data type returned	Description
GetFixesVersion	Integer	Returns the fix level for the current PowerBuilder execution context.
GetHostObject	Integer	Provides a reference to the context's host object.
GetMajorVersion	Integer	Returns the major version for the current PowerBuilder execution context.
GetMinorVersion	Integer	Returns the minor version for the current PowerBuilder execution context.
GetName	Integer	Returns the name for the current execution context.
GetParent	PowerObject	Returns a reference to the name of the parent object.
GetShortName	Integer	Returns the short name for the current PowerBuilder execution context.
GetVersionName	Integer	Returns complete version information for the current PowerBuilder execution context.
PostEvent	Boolean	Adds an event to the end of the message queue for the object.
TriggerEvent	Integer	Triggers a specific event in the object and executes the script for the event.
TypeOf	Object	Returns the type of the object.

ContextKeyword

The ContextKeyword object provides environment information for the current context. In the default environment, the ContextKeyword object provides host workstation environment variables. When running within the Window Plug-in, the ContextKeyword object allows you to access parameters specified in the plug-in's EMBED element.

Properties

ContextKeyword property	Data type	Description
ClassDefinition	PowerObject	An object of type ClassDefinition describing the ContextKeyword class.

Events

ContextKeyword event	Occurs
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.

Functions

ContextKeyword function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextKeywords	Integer	Retrieves one or more values associated with a specified keyword.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetName	Integer	Returns the name for the current execution context.

ContextKeyword function	Data type returned	Description
GetParent	PowerObject	Returns a reference to the name of the parent object.
PostEvent	Boolean	Adds an event to the end of the message queue for the object.
TriggerEvent	Integer	Triggers a specific event in the object and executes the script for the event.
TypeOf	Object	Returns the type of the object.

CPlusPlus object

The CPlusPlus object is a special type of class user object. It serves as a repository for the exported function and data declarations that will make up a C++ class in a DLL that you create.

In order to use this object, you must install the C++ Class Builder as part of your PowerBuilder Enterprise installation. You use the User Object painter to create this object.

Platform Information

The C++ Class Builder is only supported on the Windows Platform.

FOR INFO For more information about using the CPlusPlus object and the C++ Class Builder, see *C++ Class Builder*.

Properties

CPlusPlus property	Data type	Description
LibraryName	String	Specifies the path where the application will try to find the DLL you create.

Events

CPlusPlus event	Occurs
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.

Functions

CPlusPlus function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.

CPlusPlus function	Data type returned	Description
GetParent	PowerObject	Returns a reference to the name of the parent object.
PostEvent	Boolean	Adds an event to the end of the message queue for the control.
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.

DataStore object

A DataStore is a nonvisual DataWindow control. DataStores act just like DataWindow controls except that many of the visual properties associated with DataWindow controls do not apply to DataStores. However, because you can print DataStores, PowerBuilder provides some events and functions for DataStores that pertain to the visual presentation of the data.

Properties

DataStore property	Data type	Description
DataObject	String	Specifies the name of the DataWindow or Report object associated with the control.
Object	DWObject	Used for the direct manipulation of objects within a DataWindow object from a script. These objects could be, for example, columns or text objects.

Events

Some but not all DataStore events have return codes that you can use to determine what action will be taken after the event occurs. You set the return codes in a RETURN statement in the event script.

DataStore event	Occurs
Constructor	Immediately before the Open event occurs in the window.
DBError	When a database error occurs in the DataStore. Return codes: 0 - (Default) Display the error message. 1 - Do not display the error message.
Destructor	Immediately after the Close event occurs in the window.
Error	When an error is found in a data or property expression for a DataWindow object.

DataStore event	Occurs
ItemChanged	<p>When the AcceptText and Update functions are called for the DataStore.</p> <p>Return codes:</p> <ul style="list-style-type: none"> 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 focus to change.
ItemError	<p>When a value imported into a DataStore from a string or file does not pass the validation rules for its column. Can also occur when the AcceptText and Update functions are called for the DataStore.</p> <p>Return codes:</p> <ul style="list-style-type: none"> 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. <p>If the Return code is 0 or 1 (rejects the data), the field with the incorrect data regains the focus.</p>
PrintEnd	<p>When the printing of the DataStore ends.</p>
PrintPage	<p>Before each page of the DataStore is formatted for printing.</p> <p>Return codes:</p> <ul style="list-style-type: none"> 0 - Do not skip a page. 1 - Skip a page.
PrintStart	<p>When the printing of the DataStore starts.</p>
RetrieveEnd	<p>When the retrieval for the DataStore is complete.</p>
RetrieveRow	<p>After a row has been retrieved.</p> <p>Return codes:</p> <ul style="list-style-type: none"> 0 - (Default) Continue. 1 - Stop the retrieval.
RetrieveStart	<p>When the retrieval for the DataStore is about to begin.</p> <p>Return codes:</p> <ul style="list-style-type: none"> 0 - (Default) Continue. 1 - Do not perform the retrieval. 2 - Do not reset the rows and buffers before retrieving the data from the database.

DataStore event	Occurs
SQLPreview	<p>After a Retrieve, Update, or ReselectRow function call and immediately before the SQL statement is submitted to the DBMS.</p> <p>The following return codes specify the action that takes place when the event occurs after an Update function call only:</p> <p>0 - (Default) Continue. 1 - Stop. 2 - Skip this request and execute the next request.</p>
UpdateEnd	<p>When all the updates from the DataStore to the database are complete.</p>
UpdateStart	<p>After an Update function call and just before changes in the DataStore are sent to the database.</p> <p>Return codes:</p> <p>0 - (Default) Continue. 1 - Do not perform the update.</p>

Functions

DataStore function	Data type returned	Description
AcceptText	Integer	Applies the contents of the DataStore's edit control to the current item in the DataStore buffer.
CategoryCount	Integer	Returns the number of categories in the specified graph in the DataStore.
CategoryName	String	Returns the name of the specified category in the specified graph in the DataStore.
ClassName	String	Returns the name assigned to the DataStore.
ClearValues	Integer	Deletes all items from the value list associated with the specified column in the DataStore.
Clipboard	Integer	Copies the specified graph in the DataStore to the clipboard.

DataStore function	Data type returned	Description
CopyRTF	String	Returns the selected text, pictures, and input fields in a DataStore as a string with rich text formatting. Bitmaps and input fields are included in the string.
Create	Integer	Creates a DataWindow object using the specified source code and replaces the DataWindow object in the specified DataStore with the new DataWindow object.
DataCount	Long	Returns the number of data points in the specified series in the specified graph in the DataStore.
DBCcancel	Integer	Cancels a database retrieval in progress.
DeletedCount	Long	Returns the number of rows that have been deleted from the DataStore but have not yet been updated in the associated database table.
DeleteRow	Integer	Deletes the specified row from the DataStore.
Describe	String	Returns requested information about the structure of the DataStore.
Filter	Integer	Moves rows that do not meet the current filter criteria to the filter buffer.
FilteredCount	Integer	Returns the number of rows that do not meet the current filter criteria.
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 DataStore.
FindCategory	Integer	Returns the number of the specified category in the specified graph in the DataStore.
FindGroupChange	Long	Searches starting at a specified row for the first break for the specified group in the DataStore.

DataStore function	Data type returned	Description
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 DataStore.
GetBorderStyle	Border (enumerated)	Returns a Border enumerated data type indicating the border style of the specified column in the DataStore. Border enumerated data types are: Box! Lowered! NoBorder! Raised! ResizeBorder! ShadowBox! Underline!
GetChild	Integer	Stores in the specified variable the name of the child DataWindow in the specified column.
GetColumn	Integer	Returns the number of the current column in the DataStore.
GetColumnName	String	Returns the name of the current column in the DataStore.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetData	Double	Returns the value of the specified data in the specified series in the specified graph in the DataStore. See also GetDataValue.
GetDataPieExplode	Integer	Reports the percentage that a pie slice is exploded in a pie graph.
GetDataStyle	Integer	Finds out the appearance of a data point in a graph. Each data point in a series can have individual appearance settings. There are different syntaxes, depending on what settings you want to check.

DataStore function	Data type returned	Description
GetDataValue	Integer	Returns the value of the specified data in the specified series in the specified graph in the DataStore.
GetFormat	String	Returns the format used for display in the specified column of the DataStore.
GetItemDate	Date	Returns the date data in the specified row and column of the DataStore.
GetItemDateTime	DateTime	Returns the datetime data in the specified row and column of the DataStore.
GetItemDecimal	Decimal	Returns the decimal data in the specified row and column of the DataStore.
GetItemNumber	Double	Returns the numeric data in the specified row and column of the DataStore.
GetItemStatus	dwItemStatus (enumerated)	Returns the status of the item at the specified row and column location in the specified buffer. Values are: DataModified! New! NewModified! NotModified!
GetItemString	String	Returns the string data in the specified row and column of the DataStore.
GetItemTime	Time	Returns the time data in the specified row and column of the DataStore.
GetNextModified	Long	Returns the number of the first row that was modified in the specified buffer in the specified DataStore after the specified row.
GetParent	PowerObject	Returns a reference to the name of the parent object.
GetRow	Long	Returns an integer containing the number of the current row in the DataStore.

DataStore function	Data type returned	Description
GetSelectedRow	Integer	Returns the number of the first selected row after the specified row number in the DataStore.
GetSeriesStyle	Integer	Finds out the appearance of a series in a graph. There are several syntaxes, depending on what settings you want.
GetSQLSelect	String	Returns the current SELECT statement for the DataStore.
GetText	String	Returns the text in the edit control over the current row and column of the DataStore.
GetTrans	Integer	Returns the values in the DataStore transaction object.
GetValidate	String	Returns the validation rule used in the specified column of the DataStore.
GetValue	String	Returns the specified item in the value list for the specified column.
GroupCalc	Integer	Recalculates the breaks in the groups in the DataStore.
ImportClipboard	Long	Copies data from the clipboard to the DataStore.
ImportFile	Long	Copies data from a file to the DataStore.
ImportString	Long	Copies data from a string to the DataStore.
InsertDocument	Integer	<p>Inserts a rich text format or plain text file into a DataStore. You use a function parameter to specify how the new content is added:</p> <ul style="list-style-type: none"> ◆ It can be inserted at the insertion point ◆ It can replace all existing content <p>This function only applies to DataStores whose content has the RichText presentation style.</p>

DataStore function	Data type returned	Description
InsertRow	Long	Inserts a new initialized row before the specified row in the DataStore.
IsSelected	Boolean	Returns True if the specified row in the DataStore is selected; returns False if the row is not selected or is greater than the number of rows in the DataStore.
ModifiedCount	Long	Returns the number of rows that have been modified in the DataStore but have not yet been updated in the associated database table.
Modify	String	Uses the specification contained in a string to modify the DataStore.
PasteRTF	Long	Pastes rich text data from a string into a DataStore whose contents has the RichText presentation style.
PostEvent	Boolean	Adds an event to the end of the message queue for the DataStore.
Print	Integer	Sends the contents of the DataStore to the current printer. This function has several syntaxes.
PrintCancel	Integer	<p>Cancels the print job and deletes the spool file (if any) when the contents of the DataStore is sent to print.</p> <p>This function has two syntaxes. Use Syntax 1 when Syntax 1 of the Print function was used to send it to print.</p>
ReselectRow	Integer	Accesses the database to reselect all columns that can be updated and refreshes all timestamp columns in a row in the DataStore.

DataStore function	Data type returned	Description
Reset	Integer	Syntax 1: Clears all the data from a DataStore. Syntax 2: Does not apply to a DataStore object. Syntax 3: Deletes the data, the categories, or the series from a graph within a DataWindow object with an external data source.
ResetDataColors	Integer	Resets the color of a data point to the color specified for the series.
ResetTransObject	Integer	Stops the DataStore from using a programmer-defined Transaction object (thereafter, the DataStore uses its internal transaction object).
ResetUpdate	Integer	Resets the update flags for the DataStore.
Retrieve	Long	Causes the DataStore to retrieve rows from the database.
RowCount	Long	Returns the number of rows currently available in the DataStore (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 DataStore to another DataStore (or DataWindow control) or from one buffer to another within a single DataStore.
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 DataStore and inserts the rows in another DataStore (or DataWindow control) or another buffer of the same DataStore.

DataStore function	Data type returned	Description
SaveAs	Integer	Saves the data represented in the specified graph in the DataStore to the specified filename in the specified format.
SaveAs	Integer	<p>Syntax 1: Saves the contents of the DataStore to the specified file, in the specified format, with or without column headings at the beginning</p> <p>Syntax 2: Saves the data represented in the specified graph in the DataStore to the specified file, in the specified format.</p> <hr/> <p>Platform note For information on the file formats each platform supports, see the <i>PowerScript Reference</i>.</p> <hr/>
SelectRow	Integer	Selects or deselects the specified row of the DataStore.
SeriesCount	Integer	Returns the number of series in the specified graph in the DataStore.
SeriesName	String	Returns the name of the specified series in the specified graph in the DataStore.
SetBorderStyle	Integer	Sets the border style of the specified column in the DataStore.
SetColumn	Integer	Makes the specified column the current column in the DataStore.
SetDataPieExplode	Integer	Explodes a pie slice in a pie graph.
SetDataStyle	Integer	<p>For the specified data point in the specified series in the specified graph in the DataStore:</p> <p>Syntax 1: Sets the data point's color.</p> <p>Syntax 2: Sets the line style and width for the data point.</p> <p>Syntax 3: Sets the fill pattern or symbol for the data point.</p>

DataStore function	Data type returned	Description
SetDetailHeight	Integer	Sets the height of each row in a specified range.
SetFilter	Integer	Defines the filter criteria for the DataStore. The actual filtering is performed by the Filter function.
SetFormat	Integer	Sets the display format for the specified column of the DataStore.
SetItem	Integer	Sets the value of the specified row and column of the specified DataStore.
SetItemStatus	Integer	Sets the status of a row in a specified column of the DataStore in the specified buffer.
SetPosition	Integer	Moves an object within the DataStore to another band or changes the front-to-back order of objects within a band.
SetRow	Integer	Makes the specified row the current row in the DataStore.
SetSeriesStyle	Integer	For the specified series in the specified graph in the DataStore: Syntax 1: Sets the series' color. Syntax 2: Sets the linestyle and width. Syntax 3: Sets the fill pattern or symbol for data markers in the series. Syntax 4: Specify that the series is an overlay.
SetSort	Integer	Defines the sort criteria for the DataStore. The actual sorting is performed by the Sort function.
SetSQLPreview	Integer	Sets the current SQL statement for the DataStore.
SetSQLSelect	Integer	Changes the current SELECT statement for the DataStore.
SetText	Integer	Replaces the text in the edit control at the current row and column of the DataStore with the specified text.

DataStore function	Data type returned	Description
SetTrans	Integer	Sets values in the DataStore's internal transaction object.
SetTransObject	Integer	Sets the transaction object for the DataStore 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 DataStore.
SetValue	Integer	Sets the value of the specified item in the value list or the code table of the specified column of the DataStore.
ShareData	Integer	Shares data between a primary DataStore (or DataWindow control) and a secondary DataStore (or DataWindow control).
ShareDataOff	Integer	Turns off sharing for the DataStore. If the DataStore is primary, all secondary DataStores (or DataWindow controls) are disconnected and their DataWindow objects no longer contain data.
Sort	Integer	Sorts the rows of the DataStore based on its current sort criteria.
TriggerEvent	Integer	Triggers a specified event in the DataStore and executes the script for the event.
TypeOf	Object	Returns the type of the DataStore.
Update	Integer	Sends to the database all inserts, deletes, and updates of the DataStore.

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.

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

FOR INFO For information about DataWindow objects, see the *PowerBuilder User's Guide*.

Obsolete functions

Several DataWindow control functions are described as obsolete, which means that although the function operates as usual in PowerBuilder 6.0, it will be removed in a future release. You should replace all use of these functions as soon as possible.

Properties

DataWindow property	Data type	Description
Border	Boolean	Specifies whether the control has a border. Values are: <ul style="list-style-type: none">◆ True — Control has a border◆ False — Control does not have a border
BorderStyle	BorderStyle (enumerated)	Specifies the border style of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order.

DataWindow property	Data type	Description
ControlMenu	Boolean	<p>Specifies whether the Control Menu box displays in the control title bar. Values are:</p> <ul style="list-style-type: none"> ◆ True — Control Menu box displays in the control title bar ◆ False — Control Menu box does not display in the control title bar
DataObject	String	<p>Specifies the name of the DataWindow object or Report object associated with the control.</p>
DragAuto	Boolean	<p>Specifies whether PowerBuilder puts the control automatically into Drag Mode. DragAuto has these boolean values:</p> <ul style="list-style-type: none"> ◆ 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	<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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control is enabled ◆ False — Control is not enabled

DataWindow property	Data type	Description
Height	Integer	Specifies the height of the DataWindow control, in PowerBuilder units.
HScrollBar	Boolean	Specifies whether a horizontal scrollbar displays in the control when all the data cannot be displayed at one time. Values are: <ul style="list-style-type: none"> ◆ True — Horizontal scrollbar is displayed ◆ False — Horizontal scrollbar is not displayed
HSplitScroll	Boolean	Specifies whether the split bar displays in the control. Values are: <ul style="list-style-type: none"> ◆ True — Split bar is displayed ◆ False — Split bar is not displayed
Icon	String	Specifies the name of the ICO file that contains the icon that displays when the DataWindow control is minimized. <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
LiveScroll	Boolean	Scrolls the rows in the DataWindow control while the user is moving the scrollbar.
MaxBox	Boolean	Specifies whether a Maximize Box displays in the DataWindow control title bar. Values are: <ul style="list-style-type: none"> ◆ True — Maximize Box displays ◆ False — Maximize Box does not display
MinBox	Boolean	Specifies whether a Minimize Box displays in the DataWindow control title bar. Values are: <ul style="list-style-type: none"> ◆ True — Minimize Box displays ◆ False — Minimize Box does not display
Object	DWObject	Used for the direct manipulation of objects within a DataWindow object from a script. These objects could be, for example, columns or text objects.

DataWindow property	Data type	Description
Resizable	Boolean	Specifies whether the DataWindow control is resizable. Values are: <ul style="list-style-type: none"> ◆ True — DataWindow is resizable ◆ False — DataWindow is not resizable
RightToLeft	Boolean	Specifies that characters should be displayed in right-to-left order. The application must be running on a Hebrew or Arabic version of PowerBuilder under an operating system that supports right-to-left display. Values are: <ul style="list-style-type: none"> ◆ True — Characters display in right-to-left order ◆ False — Characters display in left-to-right order
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 control title bar.
TitleBar	Boolean	Specifies whether a title bar displays in the DataWindow control. The user can move the DataWindow control only if it has a title bar. Values are: <ul style="list-style-type: none"> ◆ True — Title bar is displayed in control ◆ False — No title bar is displayed in control
Visible	Boolean	Specifies whether the DataWindow control is visible. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
VScrollBar	Boolean	Specifies whether a vertical scrollbar displays in the control when not all the data can be displayed at one time. Values are: <ul style="list-style-type: none"> ◆ True — Vertical scrollbar is displayed ◆ False — Vertical scrollbar is not displayed

DataWindow property	Data type	Description
Width	Integer	Specifies the width of the DataWindow 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 edge of the window), in PowerBuilder units.

Events

Some but not all DataWindow events have return codes that you can use to determine what action will be taken after the event occurs. You set the return codes in a RETURN statement in the event script.

DataWindow event	Occurs
Clicked	When the user clicks a noneditable field or between fields in the DataWindow control. Return 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 control. Return 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 control. For a RichText presentation style DataWindow, when the user double-clicks in the text.
DragDrop	When a dragged control is dropped on the DataWindow control.
DragEnter	When a dragged control enters the DataWindow control.

DataWindow 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 control.
Error	When an error is found in a data or property expression for a DataWindow object.
GetFocus	Just before the DataWindow control receives focus (before it is selected and becomes active).
ItemChanged	<p>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 <i>within the DataWindow</i>).</p> <p>Return codes:</p> <ul style="list-style-type: none"> 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 focus to change.
ItemError	<p>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.</p> <p>Return codes:</p> <ul style="list-style-type: none"> 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. <p>If the Return code is 0 or 1 (rejects the data), the field with the incorrect data regains the focus.</p>
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.

DataWindow event	Occurs
PrintPage	<p>Before each page of the DataWindow is formatted for printing.</p> <p>Return codes:</p> <ul style="list-style-type: none"> 0 - Do not skip a page. 1 - Skip a page.
PrintStart	When the printing of the DataWindow starts.
RButtonDown	<p>When the right mouse button is pressed on the control.</p> <p>For a RichText presentation style DataWindow, if PopUp Menu has been turned on, this event will not be triggered when the right mouse button is pressed.</p>
Resize	When the user or a script resizes a DataWindow control.
RetrieveEnd	When the retrieval for the DataWindow is complete.
RetrieveRow	<p>After a row has been retrieved.</p> <p>Return codes:</p> <ul style="list-style-type: none"> 0 - (Default) Continue. 1 - Stop the retrieval.
RetrieveStart	<p>When the retrieval for the DataWindow is about to begin.</p> <p>Return codes:</p> <ul style="list-style-type: none"> 0 - (Default) Continue. 1 - Do not perform the retrieval. 2 - Do not reset the rows and buffers before retrieving the data from the database.
RowFocusChanged	When the current row changes in the DataWindow.
ScrollHorizontal	When the user scrolls right or left in the DataWindow control with the TAB or arrow keys or the scrollbar.
ScrollVertical	When the user scrolls up or down in the DataWindow control with the TAB or arrow keys or the scrollbar.
SQLPreview	<p>After a Retrieve, Update, or ReselectRow function call and immediately before the SQL statement is submitted to the DBMS.</p> <p>The following return codes specify the action that takes place when the event occurs after an Update function call only:</p> <ul style="list-style-type: none"> 0 - (Default) Continue. 1 - Stop. 2 - Skip this request and execute the next request.

DataWindow event	Occurs
UpdateEnd	When all the updates from the DataWindow to the database are complete.
UpdateStart	After an Update function call and just before changes in the DataWindow are sent to the database. Return codes: 0 - (Default) Continue. 1 - Do not perform the update.

Functions

DataWindow function	Data type returned	Description
AcceptText	Integer	Applies the contents of the DataWindow control's edit control to the current item in the DataWindow buffer.
CanUndo	Boolean	Specifies whether the last edit can be undone with the Undo function. Applies to the edit control over the current row and column. Values are: ◆ True — Last edit can be undone ◆ False — Last edit cannot be undone
CategoryCount	Integer	Returns the number of categories in the specified graph in the DataWindow control.
CategoryName	String	Returns the name of the specified category in the specified graph in the DataWindow control.
ClassName	String	Returns the name assigned to the DataWindow control.
Clear	Integer	Clears (deletes) the selected text in the edit control of the DataWindow control. For a RichText presentation style DataWindow, clears the selected text in the DataWindow.

DataWindow function	Data type returned	Description
ClearValues	Integer	Deletes all items from the value list associated with the specified column in the DataWindow control.
Clipboard	Integer	Copies the specified graph in the DataWindow control to the clipboard.
Copy	Integer	Copies the selected text in the edit control over the current row and column of the DataWindow control to the clipboard. For a RichText presentation style DataWindow, copies the selected text in the DataWindow control.
CopyRTF	String	Returns the selected text, pictures, and input fields in a DataWindow control as a string with rich text formatting. Bitmaps and input fields are included in the string.
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 object during execution.
Cut	Integer	Cuts the selected text from the edit control over the current row and column of the DataWindow and stores it in the clipboard. For a RichText presentation style DataWindow, cuts the selected text in the DataWindow control.
DataCount	Long	Returns the number of data points in the specified series in the specified graph in the DataWindow control.

DataWindow function	Data type returned	Description
DBCcancel	Integer	Cancels a database retrieval in progress.
DBErrorCode	Long	Returns the error code (number) generated by a database error. Obsolete function DBErrorCode is an obsolete function and will be discontinued in the near future. Database error codes are now available as event arguments.
DBErrorMessage	String	Returns a string containing the text of the error message generated by a database error. Obsolete function DBErrorMessage is an obsolete function and will be discontinued in the near future. Database error messages are now available as event arguments.
DeletedCount	Long	Returns the number of rows that have been deleted from the DataWindow control but have not yet been updated in the associated database table.
DeleteRow	Integer	Deletes the specified row from the DataWindow control.
Describe	String	Returns requested information about the structure of the DataWindow control.
Drag	Integer	Starts or ends the dragging of the DataWindow control.
Filter	Integer	Displays specific rows of the DataWindow control based on its current filter.

DataWindow function	Data type returned	Description
FilteredCount	Integer	Returns the number of rows that are not visible because of the current filter.
Find	Long	Syntax 1: Finds the next row in a DataWindow control in which data meets a specified condition. Syntax 2: For Rich Text presentation style Data Windows, finds the specified text in the control and highlights the text if found. You can specify search direction and whether to match whole words and case.
FindCategory	Integer	Returns the number of the specified category in the specified graph in the DataWindow control.
FindGroupChange	Long	Searches starting at a specified row for the first break for the specified group in the DataWindow control.
FindNext	Integer	Finds the next occurrence of text in the control and highlights it, using criteria set up in a previous call of the Find function. This function only applies to DataWindow controls whose content has the RichText presentation style.
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 control.
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.

DataWindow function	Data type returned	Description
GetBorderStyle	Border (enumerated)	Returns a Border enumerated data type indicating the border style of the specified column in the DataWindow control. Values are: Box! Lowered! NoBorder! Raised! ResizeBorder! ShadowBox! Underline!
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 control that the user clicked or double-clicked.
GetClickedRow	Long	Returns the number of the row in the DataWindow control that the user clicked or double-clicked.
GetCurrentColumn	Integer	Returns the number of the current column in the DataWindow control.
GetCurrentColumnName	String	Returns the name of the current column in the DataWindow control.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetData	Double	Returns the value of the specified data in the specified series in the specified graph in the DataWindow control. See also GetDataValue.
GetDataPieExplode	Integer	Reports the percentage that a pie slice is exploded in a pie graph.
GetDataStyle	Integer	Finds out the appearance of a data point in a graph. Each data point in a series can have individual appearance settings. There are different syntaxes, depending on what settings you want to check.

DataWindow function	Data type returned	Description
GetDataValue	Integer	Returns the value of the specified data in the specified series in the specified graph in the DataWindow control.
GetFormat	String	Returns the format used for display in the specified column of the DataWindow control.
GetItemDate	Date	Returns the date data in the specified row and column of the DataWindow control.
GetItemDateTime	DateTime	Returns the datetime data in the specified row and column of the DataWindow control.
GetItemDecimal	Decimal	Returns the decimal data in the specified row and column of the DataWindow control.
GetItemNumber	Double	Returns the numeric data in the specified row and column of the DataWindow control.
GetItemStatus	dwItemStatus (enumerated)	Returns the status of the item at the specified row and column location in the specified buffer. Values are: DataModified! New! NewModified! NotModified!
GetItemString	String	Returns the string data in the specified row and column of the DataWindow control.
GetItemTime	Time	Returns the time data in the specified row and column of the DataWindow control.

DataWindow function	Data type returned	Description
GetMessageText	String	Obtains the message text generated by a crosstab DataWindow object in the DataWindow control. Obsolete function GetMessageText is an obsolete function and will be discontinued in the near future. Message text is now available as an argument in a user-defined DataWindow event associated with the token <code>pbm_dwnmessagetext</code> .
GetNextModified	Long	Returns the number of the first row that was modified in the specified buffer in the specified DataWindow control after the specified row.
GetObjectAtPointer	String	Returns the string containing the name of the column or graphic control under the pointer in the DataWindow control, then a tab character (~t), and then the row number.
GetParent	PowerObject	Returns a reference to the name of the parent object.
GetRow	Long	Returns an integer containing the number of the current row in the DataWindow control.
GetSelectedRow	Integer	Returns the number of the first selected row after the specified row number in the DataWindow control.
GetSeriesStyle	Integer	Finds out the appearance of a series in a graph. There are several syntaxes, depending on what settings you want.

DataWindow function	Data type returned	Description
GetSQLPreview	String	Returns the current SQL statement the DataWindow control is submitting to the database. Obsolete function GetSQLPreview is an obsolete function and will be discontinued in the near future. SQL syntax is now available as an event argument.
GetSQLSelect	String	Returns the current SELECT statement for the DataWindow control.
GetText	String	Returns the text in the edit control over the current row and column of the DataWindow control.
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. Obsolete function GetUpdateStatus is an obsolete function and will be discontinued in the near future. Update status is now available as an argument in the DataWindow control DBError and SQLPreview events.
GetValidate	String	Returns the validation rule used in the specified column of the DataWindow control.
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 control.

DataWindow function	Data type returned	Description
Hide	Integer	Makes the control invisible.
ImportClipboard	Long	Copies data from the clipboard to the DataWindow control.
ImportFile	Long	Copies data from a file to the DataWindow control.
ImportString	Long	Copies data from a string to the DataWindow control.
InsertDocument	Integer	<p>Inserts a rich text format or plain text file into a DataWindow control. You use a function parameter to specify how the new content is added:</p> <ul style="list-style-type: none"> ◆ It can be inserted at the insertion point ◆ It can replace all existing content <p>This function only applies to DataWindow controls whose content has the RichText presentation style.</p>
InsertRow	Long	Inserts a new initialized row before the specified row in the DataWindow control.
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 control.
LineCount	Integer	Determines the number of lines in an edit control that allows multiple lines.
ModifiedCount	Long	Returns the number of rows that have been modified in the DataWindow control but have not yet been updated in the associated database table.

DataWindow function	Data type returned	Description
Modify	String	Uses the specification contained in a string to modify the DataWindow control.
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 control and identifies the object type.
OLEActivate	Integer	<p>Activates OLE for the OLE object in the specified row and column of the DataWindow control.</p> <hr/> <p>Platform note This and other OLE functions are only available on the Windows platform.</p> <hr/>
Paste	Integer	Inserts the contents of the clipboard into the edit control over the current row and column in the DataWindow control.
PasteRTF	Long	Pastes rich text data from a string into a DataWindow control whose contents has the RichText presentation style.
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.

DataWindow function	Data type returned	Description
Position	Integer	<p>Syntax 1: Reports the insertion point's position in the edit control over the current row and column of the DataWindow control.</p> <p>Syntax 2: Reports the line and column position of the insertion point or the start and end of selected text in a DataWindow control whose contents has the RichText presentation style.</p>
PostEvent	Boolean	Adds an event to the end of the message queue for the DataWindow control.
Print	Integer	Sends the contents of the DataWindow control to the current printer. This function has several syntaxes.
PrintCancel	Integer	<p>Cancels the print job and deletes the spool file (if any) when the contents of the DataWindow control is sent to print.</p> <p>This function has two syntaxes. Use Syntax 1 when Syntax 1 of the Print function was used to send it to print.</p>
ReplaceText	Integer	Replaces the selected text in the edit control 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 control.
Reset	Integer	<p>Syntax 1: Clears all the data from a DataWindow control.</p> <p>Syntax 2: Does not apply to a DataWindow object.</p> <p>Syntax 3: Deletes the data, the categories, or the series from a graph within a DataWindow object with an external data source.</p>

DataWindow function	Data type returned	Description
ResetDataColors	Integer	Resets the color of a data point to the color specified for the series.
ResetTransObject	Integer	Stops the DataWindow control from using a programmer-defined transaction object (thereafter, the DataWindow uses its internal transaction object).
ResetUpdate	Integer	Resets the update flags for the DataWindow control.
Resize	Integer	Changes the width and height of the DataWindow control.
Retrieve	Long	Retrieves rows from the database for the DataWindow control.
RowCount	Long	Returns the number of rows currently available in the DataWindow control (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.

DataWindow function	Data type returned	Description
SaveAs	Integer	<p>Syntax 1: Saves the contents of the DataWindow control to the specified file, in the specified format, with or without column headings at the beginning</p> <p>Syntax 2: Saves the data represented in the specified graph in the DataWindow control to the specified file, in the specified format.</p> <hr/> <p>Platform note For information on the file formats which each platform supports, see the <i>PowerScript Reference</i>.</p>
Scroll	Integer	Scrolls the edit control of a DataWindow control the specified direction the specified number of lines.
ScrollNextPage	Long	<p>Syntax 1: Scrolls forward by the number of rows showing in the DataWindow (when the DataWindow control contents does not have the RichText presentation style).</p> <p>Syntax 2: Scrolls to the next page of the document in a DataWindow control whose contents has the RichText presentation style.</p>
ScrollNextRow	Long	Scrolls the DataWindow control to the next row. ScrollNextRow changes the current row but does not change the current column.

DataWindow function	Data type returned	Description
ScrollPriorPage	Long	<p>Syntax 1: Scrolls backward by the number of rows showing in the DataWindow (when the DataWindow control contents does not have the RichText presentation style).</p> <p>Syntax 2: Scrolls to the prior page of the document in a DataWindow control whose contents has the RichText presentation style.</p>
ScrollPriorRow	Long	Scrolls to the previous row. The ScrollPriorRow function changes the current row in the DataWindow control 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 control but does not change the current column.
SelectedLength	Integer	Reports the total number of characters and spaces (length) in the selected text in the edit control over the current row and column.
SelectedLine	Integer	Reports the line number in the edit control over the current row and column.
SelectedStart	Integer	Reports the starting position in the edit control over the current row and column.
SelectedText	String	Reports what text (if any) is selected in the edit control over the current row and column of the DataWindow control.
SelectRow	Integer	Selects or deselects the specified row in the DataWindow control.

DataWindow function	Data type returned	Description
SelectText	Integer	<p>Syntax 1: Selects text in the edit control of a DataWindow control (other than one whose contents is in the RichText presentation style). You specify where the selection begins and how many characters to select.</p> <p>Syntax 2: Selects text beginning and ending at the specified line and character positions in a DataWindow control whose contents is in the RichText presentation style.</p>
SelectTextAll	Integer	Selects all the contents of a DataWindow control with the RichText presentation style.
SelectTextLine	Integer	Selects the line containing the insertion point in a DataWindow control with the RichText presentation style.
SelectTextWord	Integer	Selects the word containing the insertion point in a DataWindow control with the RichText presentation style.
SeriesCount	Integer	Returns the number of series in the specified graph in the DataWindow control.
SeriesName	String	Returns the name of the specified series in the specified graph in the DataWindow control.
SetActionCode	Integer	<p>Defines the action a DataWindow control takes following an event.</p> <hr/> <p>Obsolete function SetActionCode is an obsolete function and will be discontinued in the near future. You now set return codes in a return statement in the event script.</p>

DataWindow function	Data type returned	Description
SetBorderStyle	Integer	Sets the border style of the specified column in the DataWindow control.
SetColumn	Integer	Makes the specified column the current column in the DataWindow control.
SetDataPieExplode	Integer	Explodes a pie slice in a pie graph.
SetDataStyle	Integer	For the specified data point in the specified series in the specified graph in the DataWindow control: Syntax 1: Sets the data point's color. Syntax 2: Sets the line style and width for the data point. Syntax 3: Sets the fill pattern or symbol for the data point.
SetDetailHeight	Integer	Sets the height of each row in a specified range.
SetFilter	Integer	Defines the filter criteria for the DataWindow control. 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 control.
SetItem	Integer	Sets the value of the specified row and column of the specified DataWindow control.
SetItemStatus	Integer	Sets the status of a row in a specified column of the DataWindow control in the specified buffer.

DataWindow function	Data type returned	Description
SetPosition	Integer	Syntax 1: Specifies whether the DataWindow control always displays on top in the front-to-back order within the window. Syntax 2: Moves an object within the DataWindow to another band or changes the front-to-back order of objects within a band.
SetRedraw	Integer	Controls automatic redrawing of the DataWindow control after each change in its properties or contents.
SetRow	Integer	Makes the specified row the current row in the DataWindow control.
SetRowFocusIndicator	Integer	Sets the current row indicator for the DataWindow control.
SetSeriesStyle	Integer	For the specified series in the specified graph in the DataWindow control: Syntax 1: Sets the series' color. Syntax 2: Sets the linestyle and width. Syntax 3: Sets the fill pattern or symbol for data markers in the series. Syntax 4: Specify that the series is an overlay.
SetSort	Integer	Defines the sort criteria for the DataWindow control. The actual sorting is performed by the Sort function.
SetSQLPreview	Integer	Sets the current SQL statement for the DataWindow control.
SetSQLSelect	Integer	Changes the current SELECT statement for the DataWindow control.
SetTabOrder	Integer	Changes the tab value of the specified column in the DataWindow control.

DataWindow function	Data type returned	Description
SetText	Integer	Replaces the text in the edit control at the current row and column of the DataWindow control with the specified text.
SetTrans	Integer	Sets values in the DataWindow control's internal transaction object.
SetTransObject	Integer	Sets the transaction object for the DataWindow control 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 control.
SetValue	Integer	Sets the value of the specified item in the value list or the code table of the specified column of the DataWindow control.
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 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.
ShowHeadFoot	Integer	In a RichText presentation style DataWindow control, displays the panels for editing the header and footer or hides the panels and returns to editing the main text.
Sort	Integer	Sorts the rows of the DataWindow control based on its current sort criteria.

DataWindow function	Data type returned	Description
TextLine	String	Reports information about the edit control over the current row and column.
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 edit control over the current row and column.
Update	Integer	Sends to the database all inserts, deletes, and updates of the DataWindow control.

DataWindowChild

A DataWindowChild object is a nested report or a DropDownDataWindow within a DataWindow object. For example, a DataWindow object that populates a column having the DropDownDataWindow edit style is a DataWindowChild object.

The DataWindowChild object is used for accessing DataWindow objects independently from DataWindow functionality, and it inherits from the system Structure object because it needs storage and autoinstantiation.

A DataWindowChild object has no properties and no events.

Obsolete functions

Several DataWindowChild functions are described as obsolete, which means that although the function operates as usual in PowerBuilder 6.0, it will be removed in a future release. You should replace all use of these functions as soon as possible.

Functions

DataWindowChild function	Data type returned	Description
AcceptText	Integer	Applies the contents of the edit control to the current item in the DataWindowChild buffer.
ClassName	String	Returns the name assigned to the DataWindowChild.
ClearValues	Integer	Deletes all items from the value list associated with the specified column in the DataWindowChild.
CrosstabDialog	Integer	Displays the Crosstab Definition dialog box so the user can modify the definition of a crosstab DataWindow object during execution.
DBCcancel	Integer	Cancels a database retrieval in progress.

DataWindowChild function	Data type returned	Description
DBErrorCode	Long	Returns the error code (number) generated by a database error. Obsolete function DBErrorCode is an obsolete function and will be discontinued in the near future. Database error codes are now available as event arguments.
DBErrorMessage	String	Returns a string containing the text of the error message generated by a database error. Obsolete function DBErrorMessage is an obsolete function and will be discontinued in the near future. Database error messages are now available as event arguments.
DeletedCount	Long	Returns the number of rows that have been deleted from the DataWindowChild but have not yet been updated in the associated database table.
DeleteRow	Integer	Deletes the specified row from the DataWindowChild.
Describe	String	Returns requested information about the structure of the DataWindowChild.
Filter	Integer	Displays specific rows of the DataWindowChild based on its current filter.
FilteredCount	Integer	Returns the number of rows that are not visible because of the DataWindowChild's current filter.

DataWindowChild function	Data type returned	Description
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 DataWindowChild.
FindGroupChange	Long	Searches starting at a specified row for the first break for the specified group in the DataWindowChild.
FindRequired	Integer	Identifies the required columns that the user has not filled.
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 DataWindowChild. Values are: Box! Lowered! NoBorder! Raised! ResizeBorder! ShadowBox! Underline!
GetClickedColumn	Integer	Returns the number of the column in the DataWindowChild that the user clicked or double-clicked.
GetClickedRow	Long	Returns the number of the row in the DataWindowChild that the user clicked or double-clicked.
GetColumn	Integer	Returns the number of the current column in the DataWindowChild.
GetColumnName	String	Returns the name of the current column in the DataWindowChild.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.

DataWindowChild function	Data type returned	Description
GetFormat	String	Returns the format used for display in the specified column of the DataWindowChild.
GetItemDate	Date	Returns the date data in the specified row and column of the DataWindowChild.
GetItemDateTime	DateTime	Returns the datetime data in the specified row and column of the DataWindowChild.
GetItemDecimal	Decimal	Returns the decimal data in the specified row and column of the DataWindowChild.
GetItemNumber	Double	Returns the numeric data in the specified row and column of the DataWindowChild.
GetItemStatus	dwItemStatus (enumerated)	Returns the status of the item at the specified row and column location in the specified buffer. Values are: DataModified! New! NewModified! NotModified!
GetItemString	String	Returns the string data in the specified row and column of the DataWindowChild.
GetItemTime	Time	Returns the time data in the specified row and column of the DataWindowChild.
GetNextModified	Long	Returns the number of the first row that was modified in the specified buffer in the specified DataWindowChild after the specified row.

DataWindowChild function	Data type returned	Description
GetObjectAtPointer	String	Returns the string containing the name of the DataWindowChild column or graphic control under the pointer in the DataWindowChild, then a tab character (~t), and then the row number.
GetParent	PowerObject	Returns a reference to the name of the parent object.
GetRow	Long	Returns an integer containing the number of the current row in the DataWindowChild.
GetSelectedRow	Integer	Returns the number of the first selected row after the specified row number in the DataWindowChild.
GetSQLPreview	String	Returns the current SQL statement the DataWindowChild is submitting to the database. Obsolete function GetSQLPreview is an obsolete function and will be discontinued in the near future. SQL syntax is now available as an event argument.
GetSQLSelect	String	Returns the current SELECT statement for the DataWindowChild.
GetText	String	Returns the text in the edit control over the current row and column of the DataWindowChild.
GetTrans	Integer	Returns the values in the DataWindowChild transaction object.

DataWindowChild function	Data type returned	Description
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. Obsolete function GetUpdateStatus is an obsolete function and will be discontinued in the near future. Update status is now available as an argument in the DataWindow DBError and SQLPreview events.
GetValidate	String	Returns the validation rule used in the specified column of the DataWindowChild.
GetValue	String	Returns the specified item in the value list for the specified column.
GroupCalc	Integer	Recalculates the breaks in the groups in the DataWindowChild.
ImportClipboard	Long	Copies data from the clipboard to the DataWindowChild.
ImportFile	Long	Copies data from a file to the DataWindowChild.
ImportString	Long	Copies data from a string to the DataWindowChild.
InsertRow	Long	Inserts a new initialized row before the specified row in the DataWindowChild.
IsSelected	Boolean	Returns True if the specified row in the DataWindowChild is selected; returns False if the row is not selected or is greater than the number of rows in the DataWindowChild.

DataWindowChild function	Data type returned	Description
ModifiedCount	Long	Returns the number of rows that have been modified in the DataWindowChild but have not yet been updated in the associated database table.
Modify	String	Uses the specification contained in a string to modify the DataWindowChild.
OLEActivate	Integer	Activates OLE for the OLE object in the specified row and column of the DataWindowChild. Platform note This and other OLE functions are only available on the Windows platform.
Print	Integer	Sends the contents of the DataWindowChild to the current printer. This function has several syntaxes.
PrintCancel	Integer	Cancels the print job and deletes the spool file (if any) when the contents of the DataWindowChild is sent to print. This function has 2 syntaxes. Use Syntax 1 when Syntax 1 of the Print function was used to send it to print.
ReselectRow	Integer	Accesses the database to reselect all columns that can be updated and refreshes all timestamp columns in a row in the DataWindowChild.

DataWindowChild function	Data type returned	Description
Reset	Integer	Syntax 1: Clears all the data from a DataWindowChild. Syntax 2: Does not apply to a DataWindowChild. Syntax 3: Deletes the data, the categories, or the series from a graph within a DataWindow object with an external data source.
ResetTransObject	Integer	Stops the DataWindowChild from using a programmer-defined transaction object (thereafter, the DataWindow uses its internal transaction object).
ResetUpdate	Integer	Resets the update flags for the DataWindowChild.
Resize	Integer	Changes the width and height of the DataWindowChild.
Retrieve	Long	Causes the DataWindowChild to retrieve rows from the database.
RowCount	Long	Returns the number of rows currently available in the DataWindowChild (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 DataWindowChild to another or from one buffer to another within a single DataWindowChild.
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 DataWindowChild and inserts the rows in another DataWindowChild or another buffer of the same DataWindowChild.

DataWindowChild function	Data type returned	Description
SaveAs	Integer	<p>Syntax 1: Saves the contents of the DataWindowChild to a file in the specified format with or without column headings at the beginning.</p> <p>Syntax 2: Saves the data represented in the specified graph in the DataWindowChild to the specified filename in the specified format.</p> <hr/> <p>Platform note For information on the file formats which each platform supports, see the <i>PowerScript Reference</i>.</p>
ScrollNextPage	Long	Scrolls forward by the number of rows showing in the DataWindowChild.
ScrollNextRow	Long	Scrolls the DataWindowChild 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 DataWindowChild.
ScrollPriorRow	Long	Scrolls to the previous row. The ScrollPriorRow function changes the current row in the DataWindowChild 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 DataWindowChild but does not change the current column.
SelectRow	Integer	Selects or deselects the specified row of the DataWindowChild.

DataWindowChild function	Data type returned	Description
SetActionCode	Integer	<p>Defines the action a DataWindowChild takes following an event.</p> <hr/> <p>Obsolete function SetActionCode is an obsolete function and will be discontinued in the near future. You now set return codes in a return statement in the event script.</p> <hr/>
SetBorderStyle	Integer	Sets the border style of the specified column in the DataWindowChild.
SetColumn	Integer	Makes the specified column the current column in the DataWindowChild.
SetDetailHeight	Integer	Sets the height of each row in a specified range.
SetFilter	Integer	Defines the filter criteria for the DataWindowChild. The actual filtering is performed by the Filter function.
SetFormat	Integer	Sets the display format for the specified column of the DataWindowChild.
SetItem	Integer	Sets the value of the specified row and column of the specified DataWindowChild.
SetItemStatus	Integer	Sets the status of a row in a specified column of the DataWindowChild in the specified buffer.
SetPosition	Integer	Moves an object within the DataWindowChild to another band or changes the front-to-back order of objects within a band.

DataWindowChild function	Data type returned	Description
SetRedraw	Integer	Controls automatic redrawing of the DataWindowChild after each change in its properties or contents.
SetRow	Integer	Makes the specified row the current row in the DataWindowChild.
SetRowFocusIndicator	Integer	Sets the current row indicator for the DataWindowChild.
SetSort	Integer	Defines the sort criteria for the DataWindowChild. The actual sorting is performed by the Sort function.
SetSQLPreview	Integer	Sets the current SQL statement for the DataWindowChild.
SetSQLSelect	Integer	Changes the current SELECT statement for the DataWindowChild.
SetTabOrder	Integer	Changes the tab value of the specified column in the DataWindowChild.
SetText	Integer	Replaces the text in the edit control at the current row and column of the DataWindowChild with the specified text.
SetTrans	Integer	Sets values in the DataWindowChild's internal transaction object.
SetTransObject	Integer	Sets the transaction object for the DataWindowChild 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 DataWindowChild.
SetValue	Integer	Sets the value of the specified item in the value list or the code table of the specified column of the DataWindowChild.

DataWindowChild function	Data type returned	Description
ShareData	Integer	Shares data between a primary DataWindowChild and a secondary DataWindowChild.
ShareDataOff	Integer	Turns off sharing for the DataWindowChild. If that object is the primary DataWindowChild, all secondary DataWindowChild objects are disconnected and their DataWindow objects no longer contain data.
Sort	Integer	Sorts the rows of the DataWindowChild based on its current sort criteria.
TypeOf	Object	Returns the type of the control.
Update	Integer	Sends to the database all inserts, deletes, and updates of the DataWindowChild.

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 textbox. In other DropDownListBoxes, the user cannot modify the textbox 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 because ShowList=False, the user must click the down arrow at the end of the textbox to display it.

Properties

DropDownList Box property	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 textbox portion of the control. Values are: <ul style="list-style-type: none"> ◆ True — Can enter text in the textbox ◆ False — Cannot enter text in the textbox AllowEdit must be true when ShowList is true.
AutoHScroll	Boolean	Specifies whether the textbox portion of the control scrolls horizontally automatically when data is entered or deleted. Values are: <ul style="list-style-type: none"> ◆ True — Textbox scrolls horizontally automatically ◆ False — Textbox does not scroll horizontally automatically

DropDownListBox property	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>PowerScript Reference</i> .
Border	Boolean	Specifies whether the control has a border. However, setting this property to False has no effect on the DropDownListBox control.
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PowerBuilder will move the control to the top of the front-to-back order in the window. Values are: ◆ True — Control moved to top ◆ False — Control not moved to top
DragAuto	Boolean	Specifies whether PowerBuilder will put the control automatically into Drag mode. Values are: ◆ True — When the control is clicked, the control is automatically put in Drag mode. ◆ False — When the control is clicked, the control is not automatically put in Drag mode. You have to manually put the control into Drag mode by using the Drag function.

DropDownList Box property	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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control is enabled ◆ False — Control is not enabled
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. The application must be running on an appropriate version of PowerBuilder under an operating system that supports the selected character set. Values are:</p> <p>ANSI! ChineseBig5! DefaultCharSet! Hangeul! OEM! ShiftJIS! Symbol!</p>

DropDownList Box property	Data type	Description
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) used for the text in the control. Values are: AnyFont! Decorative! Modern! Roman! Script! Swiss!
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control. Values are: Default! Fixed! Variable!
Height	Integer	Specifies the height of the control, in PowerBuilder units.
HScrollBar	Boolean	Specifies whether a horizontal scrollbar is displayed in the control. Values are: ◆ True — Horizontal scrollbar is displayed ◆ False — Horizontal scrollbar is not displayed
Italic	Boolean	Specifies whether the text in the control is italic. Values are: ◆ True — Text is italic ◆ False — Text is not italic
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.

DropDownListBox property	Data type	Description
RightToLeft	Boolean	<p>Specifies that characters should be displayed in right-to-left order. The application must be running on a Hebrew or Arabic version of PowerBuilder under an operating system that supports right-to-left display. Values are:</p> <ul style="list-style-type: none"> ◆ True — Characters display in right-to-left order ◆ False — Characters display in left-to-right order
ShowList	Boolean	<p>Specifies whether the option list always displays in the ListBox portion of the DropDownListBox when the control displays. Values are:</p> <ul style="list-style-type: none"> ◆ True — Option list always displays ◆ False — Option list displays only when the user clicks the down arrow <p>This property is usually set to False. Note that AllowEdit must be true when ShowList is true.</p>
Sorted	Boolean	<p>Specifies whether the ListBox portion of the DropDownListBox is automatically sorted in ascending order. Values are:</p> <ul style="list-style-type: none"> ◆ True — ListBox automatically sorted ◆ False — ListBox not sorted
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>
Text	String	<p>Specifies the text in the control.</p>
TextColor	Long	<p>Specifies the numeric value of the color used for text: -2 to 16,777,215.</p> <p>For more information about color, see the <i>PowerScript Reference</i>.</p>

DropDownList Box property	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. Values are: <ul style="list-style-type: none"> ◆ True — Text is underlined ◆ False — Text is not underlined
Visible	Boolean	Specifies whether the control is visible. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
VScrollBar	Boolean	Specifies whether a vertical scrollbar is displayed in the control. Values are: <ul style="list-style-type: none"> ◆ True — Vertical scrollbar is displayed ◆ False — Vertical scrollbar is not displayed
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

DropDownListBox 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 ENTER or TAB 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

DropDownListBox function	Data type returned	Description
AddItem	Integer	<p>Adds a new item to the end of the ListBox portion of the control.</p> <p>The AddItem function does not update the Item[] property of this control.</p>

DropDownListBox function	Data type returned	Description
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.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
Hide	Integer	Makes the control invisible (hidden).
InsertItem	Integer	Adds a new item to the ListBox portion of the DropDownListBox before the item indicated by the index.

DropDownListBox function	Data type returned	Description
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 AllowEdit property must be True).
SelectItem	Integer	Finds and highlights an item in the control. Use Syntax 1 when you know the text of the item but not its position. Use Syntax 2 when you know the position of the item in the control's list or you want to clear the current selection.

DropDownListBox function	Data type returned	Description
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 properties.
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	Control	Returns the type of the control.

DropDownPictureListBox

A DropDownPictureListBox is similar to a DropDownListBox, but with the addition of pictures associated with the items in the list.

The pictures used in this control can be bitmaps (bmp file), icons (ICO file), or cursors (CUR file).

Platform note ICO files are only supported on the Windows and Macintosh platforms.

Making the list display In the development environment, if the list portion of the DropDownPictureListBox is not displayed because ShowList=False, the user must click the down arrow at the end of the textbox to display it.

Properties

DropDownPicture ListBox property	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 textbox portion of the control. Values are: <ul style="list-style-type: none"> ◆ True — Can enter text in the textbox ◆ False — Cannot enter text in the textbox AllowEdit must be true when ShowList is true.
AutoHScroll	Boolean	Specifies whether the textbox portion of the control scrolls horizontally automatically when data is entered or deleted. Values are: <ul style="list-style-type: none"> ◆ True — Textbox scrolls horizontally automatically ◆ False — Textbox does not scroll horizontally automatically

DropDownPicture ListBox property	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>PowerScript Reference</i> .
Border	Boolean	Specifies whether the control has a border. Note that setting this property to False has no effect on the control; it always has a border. Values are: ◆ True — Control has a border. ◆ False — Not applicable.
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PowerBuilder will move the control to the top of the front-to-back order in the window. Values are: ◆ True — Control moved to top ◆ False — Control not moved to top
DragAuto	Boolean	Specifies whether PowerBuilder will put the control automatically into Drag Mode. Values are: ◆ 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.

DropDownPictureListBox property	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). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control is enabled ◆ False — Control is not enabled
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. The application must be running on an appropriate version of PowerBuilder under an operating system that supports the selected character set. Values are:</p> <p>ANSI! ChineseBig5! DefaultCharSet! Hangeul! OEM! ShiftJIS! Symbol!</p>
FontFamily	FontFamily (enumerated)	<p>Specifies the font family (type style) used for the text in the control. Values are:</p> <p>AnyFont! Decorative! Modern! Roman! Script! Swiss!</p>

DropDownPicture ListBox property	Data type	Description
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control. Values are: Default! Fixed! Variable!
Height	Integer	Specifies the height of the control, in PowerBuilder units.
HScrollBar	Boolean	Specifies whether a horizontal scrollbar is displayed in the control. Values are: ◆ True — Horizontal scrollbar is displayed ◆ False — Horizontal scrollbar is not displayed
Italic	Boolean	Specifies whether the text in the control is italic. Values are: ◆ True — Text is italic ◆ False — Text is not italic
Item[]	String array	Specifies the initial item strings in the ListBox portion of the DropDownPictureListBox. This array is not updated after initialization.
ItemImageIndex[]	Integer	Initial picture index for each item in the Item property array. These values are not updated after initialization.
Limit	Integer	Specifies the maximum number of characters (0 to 32,767) the user can enter in the SingleLineEdit portion of the DropDownPictureListBox (0 means unlimited).
PictureHeight	Integer	Specifies the height of the picture, in pixels. This property may only be set when there are no images in the image list. If the value is 0 at the time the first image is added, the size of that image is used to set the size of the rest of the images added.

DropDownPicture ListBox property	Data type	Description
PictureWidth	Integer	<p>Specifies the width of the picture, in pixels.</p> <p>This property may only be set when there are no images in the image list. If the value is 0 at the time the first image is added, the size of that image is used to set the size of the rest of the images added.</p>
PictureMaskColor	Long	<p>Specifies the numeric value of the color to be used to mask user defined bitmaps added through the initial picture array or with the AddPicture function. System defined bitmaps know their mask color and this color will be ignored. This value is used when a picture is added, and therefore may be changed between AddPicture calls.</p> <p>Values can be: -2 to 16,777,215.</p> <p>For more information about color, see the <i>PowerScript Reference</i></p>
PictureName[]	String	<p>Specifies the names of the files containing the pictures added during initialization. The file extension BMP, ICO, or CUR is required.</p> <p>This property is not updated after initialization.</p>
Pointer	String	<p>Specifies the name of the stock pointer or the file containing the pointer that will be used for the control.</p>
RightToLeft	Boolean	<p>Specifies that characters should be displayed in right-to-left order. The application must be running on a Hebrew or Arabic version of PowerBuilder under an operating system that supports right-to-left display. Values are:</p> <ul style="list-style-type: none"> ◆ True — Characters display in right-to-left order ◆ False — Characters display in left-to-right order

DropDownPicture ListBox property	Data type	Description
ShowList	Boolean	<p>Specifies whether the option list always displays in the ListBox portion of the DropDownPictureListBox when the control displays. Values are:</p> <ul style="list-style-type: none"> ◆ True — Option list always displays ◆ False — Option list displays only when the user clicks the down arrow <p>This property is usually set to False. Note that AllowEdit must be true when ShowList is true.</p>
Sorted	Boolean	<p>Specifies whether the ListBox portion of the DropDownPictureListBox is automatically sorted in ascending order. Values are:</p> <ul style="list-style-type: none"> ◆ True — ListBox automatically sorted ◆ False — ListBox not sorted
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>
Text	String	<p>Specifies the text in the control.</p>
TextColor	Long	<p>Specifies the numeric value of the color used for text: -2 to 16,777,215.</p> <p>For more information about color, see the <i>PowerScript Reference</i>.</p>
TextSize	Integer	<p>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.</p>
Underline	Boolean	<p>Specifies whether the text in the control is underlined. Values are:</p> <ul style="list-style-type: none"> ◆ True — Text is underlined ◆ False — Text is not underlined

DropDownPicture ListBox property	Data type	Description
Visible	Boolean	Specifies whether the control is visible. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
VScrollBar	Boolean	Specifies whether a vertical scrollbar is displayed in the control. Values are: <ul style="list-style-type: none"> ◆ True — Vertical scrollbar is displayed ◆ False — Vertical scrollbar is not displayed
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

DropDownPicture ListBox 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.

DropDownPicture ListBox event	Occurs
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 enter or tab 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 DropDownPictureListBox.

Functions

DropDownPicture ListBox function	Data type returned	Description
AddItem	Integer	Adds a new item to the end of the ListBox portion of the control. The AddItem function does not update the Item[] or ItemPictureIndex[] properties of this control.
AddPicture	Integer	Adds the bitmap, icon, or cursor file to the main image list. This function does not update the PictureName[] property.
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).

DropDownPicture ListBox function	Data type returned	Description
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.
DeletePicture	Integer	Deletes the specified picture from the image list. This function does not update the PictureName[] property.
DeletePictures	Integer	Deletes all the pictures from the image list. This function does not update the PictureName[] property.
DirList	Boolean	Populates the ListBox portion of the DropDownPictureListBox 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.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
Hide	Integer	Makes the control invisible (hidden).

DropDownPicture ListBox function	Data type returned	Description
InsertItem	Integer	Adds a new item to the ListBox portion of the DropDownPictureListBox before the item indicated by the index. This function does not update the Item[] or ItemPictureIndex[] properties of this control.
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 AllowEdit property must be True).

DropDownPicture ListBox function	Data type returned	Description
SelectItem	Integer	Finds and highlights an item in the control. Use Syntax 1 when you know the text of the item but not its position. Use Syntax 2 when you know the position of the item in the control's list or you want to clear the current selection.
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 properties.
Show	Integer	Makes the control visible.
Text	String	Returns the text of the item in the ListBox portion of the DropDownPictureListBox identified by the index.
TotalItems	Integer	Returns the total number of items in the ListBox portion of the DropDownPictureListBox.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Control	Returns the type of the control.

DynamicDescriptionArea

DynamicDescriptionArea is a PowerBuilder data type. PowerBuilder uses a variable of this type to store information about the input and output parameters used in Format 4 of dynamic SQL.

PowerBuilder provides a global DynamicDescriptionArea named SQLDA that you can use when you need a DynamicDescriptionArea variable. If necessary, you can declare and create additional variables of this type using this system object as the data type.

FOR INFO For more information about using dynamic SQL, see the *PowerScript Reference*.

Properties

DynamicDescriptionArea property	Data type	Description
NumInputs	Integer	Specifies the number of input parameters found in the dynamic SQL PREPARE statement. PowerBuilder populates this property when the DESCRIBE statement is executed.
NumOutputs	Integer	Specifies the number of output parameters found in the PREPARE statement. If the database supports output parameter description, PowerBuilder populates this property when the DESCRIBE statement is executed. If the database does not support output parameter description, PowerBuilder populates this property when the FETCH statement is executed.

DynamicDescriptionArea property	Data type	Description
InParmType[]	ParmType (enumerated)	Array containing values specifying the data type of each input parameter. Values are: TypeBoolean! TypeDate! TypeDateTime! TypeDecimal! TypeDouble! TypeInteger! TypeLong! TypeReal! TypeString! TypeTime! TypeUInt! TypeULong! TypeUnknown
OutParmType[]	ParmType (enumerated)	Array containing values specifying the data type of each output parameter returned.

Events

DynamicDescriptionArea event	Occurs
Constructor	Immediately before the Open event occurs.
Destructor	Immediately after the Close event occurs.

Functions

DynamicDescriptionArea function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.

DynamicDescriptionArea function	Data type returned	Description
GetDynamicDate	Date	<p>Obtains data of type Date from the DynamicDescriptionArea after you have executed a dynamic SQL statement.</p> <p>Use GetDynamicDate when the value of OutParmType is TypeDate! for the value in the array that you want to retrieve.</p>
GetDynamicDateTime	DateTime	<p>Obtains data of type DateTime from the DynamicDescriptionArea after you have executed a dynamic SQL statement.</p> <p>Use GetDynamicDateTime when the value of OutParmType is TypeDateTime! for the value in the array that you want to retrieve.</p>
GetDynamicNumber	Double	<p>Obtains numeric data from the DynamicDescriptionArea after you have executed a dynamic SQL statement.</p> <p>Use GetDynamicNumber when the value of OutParmType is TypeInteger!, TypeDecimal!, TypeDouble!, TypeLong!, TypeReal!, or TypeBoolean! for the value in the array that you want to retrieve.</p>
GetDynamicString	String	<p>Obtains data of type String from the DynamicDescriptionArea after you have executed a dynamic SQL statement.</p> <p>Use GetDynamicString when the value of OutParmType is TypeString! for the value in the array that you want to retrieve.</p>

DynamicDescriptionArea function	Data type returned	Description
GetDynamicTime	Time	Obtains data of type Time from the DynamicDescriptionArea after you have executed a dynamic SQL statement. Use GetDynamicTime when the value of OutParmType is TypeTime! for the value in the array that you want to retrieve.
GetParent	PowerObject	Returns a reference to the name of the parent object.
PostEvent	Boolean	Adds the specified event to the end of the message queue for the object.
SetDynamicParm	Integer	Specifies a value for an input parameter in the DynamicDescriptionArea that will be used in an SQL OPEN or EXECUTE statement. Use SetDynamicParm to fill the parameters in the input parameter descriptor array in the DynamicDescriptionArea.
TriggerEvent	Integer	Triggers a specified event in the object and executes the script for the event.
TypeOf	Object	Returns the type of the object.

DynamicStagingArea

DynamicStagingArea is a PowerBuilder data type. PowerBuilder uses a variable of this type to store information for use in dynamic SQL statements.

The DynamicStagingArea is the only connection between the execution of a statement and a transaction object and is used internally by PowerBuilder. You cannot access information in the DynamicStagingArea, and there are no properties associated with the DynamicStagingArea.

PowerBuilder provides a global DynamicStagingArea variable named SQLSA that you can use when you need a DynamicStagingArea variable. If necessary, you can declare and create additional variables of this type using this system object as the data type.

FOR INFO For more information about using dynamic SQL, see the *PowerScript Reference*.

Events

DynamicStagingArea event	Occurs
Constructor	Immediately before the Open event occurs.
Destructor	Immediately after the Close event occurs.

Functions

DynamicStagingArea function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
PostEvent	Boolean	Adds the specified event to the end of the message queue for the object.

DynamicStagingArea function	Data type returned	Description
TriggerEvent	Integer	Triggers a specified event in the object and executes the script for the event.
TypeOf	Object	Returns the type of the object.

EditMask control

An EditMask is a box similar to a SingleLineEdit in which the user can enter and edit one line of text. The type and number of characters entered is restricted by the edit mask, and the appearance of the text is specified by the edit mask. For example, you might use an EditMask to format a telephone number or date automatically as the user enters it.

Properties

EditMask property	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 (enumerated)	Specifies the alignment of text in the control. Values are: Center! Justify! Left! Right!
AutoHScroll	Boolean	Specifies whether PowerBuilder automatically scrolls left or right when data is entered into or deleted from the control. Values are: ◆ True — Scrolls horizontally automatically ◆ False — Does not scroll automatically
AutoSkip	Boolean	Specifies whether to skip to the next control when the last character in the edit mask has been entered. Values are: ◆ True — Skip to the next control automatically ◆ False — Do not skip to the next control

EditMask property	Data type	Description
AutoVScroll	Boolean	<p>Specifies whether PowerBuilder automatically scrolls up or down when data is entered into or deleted from the control. Values are:</p> <ul style="list-style-type: none"> ◆ True — Scrolls vertically automatically ◆ False — Scrolling not automatic
BackColor	Long	<p>Specifies the numeric value of the background color: -2 to 16,777,215. For more information about color, see the <i>PowerScript Reference</i>.</p>
Border	Boolean	<p>Specifies whether the control has a border. Values are:</p> <ul style="list-style-type: none"> ◆ True — Control has a border ◆ False — Control does not have a border
BorderStyle	BorderStyle (enumerated)	<p>Specifies the style of the border of the control. Values are:</p> <p>StyleBox! StyleLowered! StyleRaised! StyleShadowBox!</p>
BringToTop	Boolean	<p>Specifies whether PowerBuilder moves the control to the top of the front-to-back order in the window:</p> <ul style="list-style-type: none"> ◆ True — Control moved to top ◆ False — Control not moved to top
DisplayData	String	<p>Specifies the data that initially displays in the control.</p>
DisplayOnly	Boolean	<p>Specifies whether the text in the control is display-only and cannot be changed by the user. Values are:</p> <ul style="list-style-type: none"> ◆ True — Text is display-only ◆ False — Text can be changed by user

EditMask property	Data type	Description
DragAuto	Boolean	<p>Specifies whether PowerBuilder puts the control automatically into Drag mode. Values are:</p> <ul style="list-style-type: none"> ◆ 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	<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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected):</p> <ul style="list-style-type: none"> ◆ True — Control is enabled ◆ False — Control is not enabled
FaceName	String	<p>Specifies the name of the typeface in which the text of the control displays (for example, ARIAL or COURIER).</p>

EditMask property	Data type	Description
FontCharSet	FontCharSet (enumerated)	Specifies the font character set used for the text in the control. Values are: ANSI! ChineseBig5! DefaultCharSet! Hangeul! OEM! ShiftJIS! Symbol!
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) used for the text in the control. Values are: AnyFont! Decorative! Modern! Roman! Script! Swiss!
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control. Values are: Default! Fixed! Variable!
Height	Integer	Specifies the height of the control, in PowerBuilder units.
HScrollBar	Boolean	Specifies whether a horizontal scrollbar displays in the control when all the data cannot be displayed at one time. Values are: ◆ True — Horizontal scrollbar displayed ◆ False — Horizontal scrollbar not displayed
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

EditMask property	Data type	Description
IgnoreDefault Button	Boolean	<p>Specifies whether the Clicked event for the window's Default command button is triggered when user presses ENTER. Values are:</p> <ul style="list-style-type: none"> ◆ True — Do not trigger Clicked event; add new line in control. ◆ False — Trigger Clicked event; do not add new line in control (default).
Increment	Double	Specifies the increment of the spin.
Italic	Boolean	<p>Specifies whether the text in the control is italic. Values are:</p> <ul style="list-style-type: none"> ◆ True — Text is italic ◆ False — Text is not italic
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)	<p>Specifies the data type of the control. Values are:</p> <p>DateMask! DateTimeMask! DecimalMask! NumericMask! StringMask! TimeMask!</p>
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.

EditMask property	Data type	Description
RightToLeft	Boolean	<p>Specifies that characters should be displayed in right-to-left order. The application must be running on a Hebrew or Arabic version of PowerBuilder under an operating system that supports right-to-left display. Values are:</p> <ul style="list-style-type: none"> ◆ True — Characters display in right-to-left order ◆ False — Characters display in left-to-right order
Spin	Boolean	<p>Specifies whether to scroll through the spin values. Values are:</p> <ul style="list-style-type: none"> ◆ True — Scroll through the spin values ◆ False — Do not scroll through the spin values
TabOrder	Integer	<p>Specifies the tab value of the control within the window (0 means the user cannot tab to the control).</p>
TabStop[]	Integer	<p>Specifies the positions of tab stops in the control.</p>
Tag	String	<p>Specifies the tag value assigned to the control.</p>
Text	String	<p>Specifies the text that displays in the control.</p>
TextColor	Long	<p>Specifies the color to be used for the text in the control.</p> <p>For more information about color, see the <i>PowerScript Reference</i>.</p>
TextCase	TextCase (enumerated)	<p>Specifies the case used to display text the user enters. Values are:</p> <p>AnyCase! Lower! Upper!</p>

EditMask property	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. For information about TextSize and EditMask behavior, see online Help.
UnderLine	Boolean	Specifies whether the text in the control is underlined. Values are: ◆ True — Text is underlined ◆ False — Text is not underlined
UseCodeTable	Boolean	Specifies whether PowerBuilder uses the code table for the column to validate data. Values are: ◆ True — Uses code table to validate data ◆ False — Does not use code table to validate data
Visible	Boolean	Specifies whether the control is visible. Values are: ◆ True — Control is visible ◆ False — Control is not visible
VScrollBar	Boolean	Specifies whether a vertical scrollbar displays in the control when not all the data can be displayed at one time. Values are: ◆ True — Vertical scrollbar is displayed ◆ False — Vertical scrollbar is not displayed
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.

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

Events

EditMask 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 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

EditMask 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.

EditMask function	Data type returned	Description
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.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetData	Integer	Obtains the unformatted data in the control.
GetParent	PowerObject	Returns a reference to the name of the parent object.
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.
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.

EditMask function	Data type returned	Description
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.
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its properties.
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.

EnumerationDefinition object

Information about the type of a variable when it is an enumerated data type. EnumerationDefinition is inherited from TypeDefinition. It has no events.

Properties

Enumeration Definition property	Data type	Description
Category	TypeCategory	Specifies if the type is simple, enumerated, or a class or structure. Values are: SimpleType! EnumeratedType! ClassOrStructureType!
ClassDefinition	PowerObject	An object of type ClassDefinition describing the TypeDefinition class.
DataTypeOf	String	The system class name or simple data type of the variable. DataTypeOf is a string representation of a value of the Object enumerated data type. Values are lowercase with no exclamation point. Sample values include: window string any dropdownlistbox For objects you have defined, the data type is the system class from which your object is inherited.
Enumeration[]	EnumerationItem Definition	An array of the name-value pairs for all the items in the enumeration.
IsStructure	Boolean	Indicates whether the type is a structure. Always False.
IsSystemType	Boolean	Indicates whether the class is a system class, that is one of the classes defined within PowerBuilder as opposed to a class defined in a PBL by a user.

Enumeration Definition property	Data type	Description
IsVariableLength	Boolean	Specifies whether the data type has a fixed size. Always True. Values are: <ul style="list-style-type: none"> ◆ True — The data type is variable length, meaning the data type is a string, any, blob, or unbounded array. ◆ False — The data type is a fixed length.
IsVisualType	Boolean	Indicates whether the type is a visual or nonvisual type. Always False.
LibraryName	String	The fully qualified name of the library the class was loaded from. Note that the library may no longer contain the class. If a program manipulates the contents of libraries, the class could have been moved or deleted after it was loaded.
Name	String	The name of the class. For a nested class, the name will be returned in the form of <i>libraryEntryName`className</i> .

Functions

Enumeration Definition function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object	Returns the type of the object.

EnumerationItemDefinition object

A class that provides information about the value names and the associated numeric values for an enumerated data type. It is used in the EnumerationDefinition class. It has no events.

Properties

EnumerationItemDefinition property	Data type	Description
ClassDefinition	PowerObject	An object of type ClassDefinition describing the EnumerationItemDefinition class.
Name	String	The name of an enumerated value. For example, Left! is a named value of the enumerated data type Alignment.
Value	Long	The numeric value associated with the name. For example, 0 is the value PowerBuilder associates with Left! The numeric value is only important if you are constructing source code for an object. Within PowerBuilder, you use the named value so that the data type is correct.

Functions

EnumerationItemDefinition function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object	Returns the type of the object.

Environment object

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

The Environment object has no events.

FOR INFO For more information about the GetEnvironment function, see the *PowerScript Reference*.

Properties

Environment property	Data type	Description
CPUType	CPUTypes (enumerated)	The type of CPU. For a complete list of CPUTypes values, see the Enumerated tab of the Browser.
MachineCode	Boolean	Specifies whether the application executable is machine code (compiled). Values are: <ul style="list-style-type: none"> ◆ True — Executable is machine code ◆ False — Executable is not machine code (pseudo-code)
OSFixesRevision	Integer	The maintenance version of the operating system.
OSMajorRevision	Integer	The major version of the operating system. For example, this value would be 3 for Windows 3.x, 4 for 32-bit Windows 95, 5 for SunOS 5.x (Solaris 2.x), 7 for Macintosh 7.x. <hr/> <p>16-bit applications under Windows 95 If you are running a 16-bit application under Windows 95, OSMajorRevision is 3, not 4, and OSMinorRevision is 95.</p> <hr/>
OSMinorRevision	Integer	The point release of the operating system. For example, this value would be 10 for Windows 3.10, 5 for SunOS 5.5 (Solaris 5.5), 5 for Macintosh 7.5.

Environment property	Data type	Description
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 (enumerated)	Operating system or environment. For a complete list of OSType values, see the Enumerated tab of the Browser.
PBType	PBTypes (enumerated)	Version of PowerBuilder product. For a complete list of PBType values, see the Enumerated tab of the Browser.
Win16	Boolean	Indicates the type of the operating system in which the application executable is running. Values are: <ul style="list-style-type: none"> ◆ True — Executable is running under a 16-bit operating system ◆ False — Executable is running under a 32-bit operating system

Functions

Environment function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GctParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object	Returns the type of the object.

Error object

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.

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

FOR INFO For more information about using the Error object in an application, see *Application Techniques*.

Properties

Error property	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

Error event	Occurs
Constructor	When the user object is created.
Destructor	When the user object is destroyed.

Functions

Error function	Data type returned	Description
ClassName	String	Returns the name assigned to the user object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent 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.

Graph object

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

Properties

Graph property	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>PowerScript Reference</i> .
Border	Boolean	Specifies whether the control has a border.
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order of the window or DataWindow control.
Category	grAxis	Specifies the properties 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.

Graph property	Data type	Description
DragAuto	Boolean	<p>Specifies whether PowerBuilder will put the graph automatically into Drag mode. Values are:</p> <ul style="list-style-type: none"> ◆ 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	<p>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.</p> <p>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.</p> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
Elevation	Integer	Specifies the angle of front-to-back elevation.
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control can be selected ◆ False — Control cannot be selected

Graph property	Data type	Description
FocusRectangle	Boolean	Specifies whether a dotted rectangle (the focus rectangle) will frame the control when it has focus. Values are: <ul style="list-style-type: none"> ◆ True — Control will be framed when it has focus ◆ False — Control will not be framed when it has focus
GraphType	grGraphType (enumerated)	Specifies the type of the graph. Values are: <ul style="list-style-type: none"> Area3D! AreaGraph! Bar3DGraph! Bar3DObjGraph! BarGraph! BarStack3DObjGraph! BarStackGraph! Col3DGraph! Col3DObjGraph! ColGraph! ColStack3DObjGraph! ColStackGraph! Line3D! LineGraph! Pie3D! PieGraph! ScatterGraph!
Height	Integer	Specifies the height of the control, in PowerBuilder units.
Legend	grLegendType (enumerated)	Specifies the alignment of the text in the graph legend. Values are: <ul style="list-style-type: none"> AtBottom! AtLeft! AtRight! AtTop! NoLegend!
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.

Graph property	Data type	Description
Perspective	Integer	Specifies the distance the graph is from the front of the window.
PieDispAttr	grDispAttr	Specifies properties 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.
Visible	Boolean	Specifies whether the control is visible. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible

Graph property	Data type	Description
Width	Integer	Specifies the stroke weight of 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

Graph 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.
DoubleClick	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

Graph 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.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetData	Double	Obtains the value of the specified data point in the specified series in the graph. See also the GetDataValue function.

Graph function	Data type returned	Description
GetDataPieExplode	Integer	Reports the percentage that a pie slice is exploded in a pie graph.
GetDataStyle	Integer	Finds out the appearance of a data point in a series. Each data point in a series can have individual appearance settings. There are different syntaxes, depending on what settings you want to check: colors, line style, fill pattern, or symbol.
GetDataValue	Integer	Returns the value of the specified data in the specified series in the specified graph.
GetSeriesStyle	Integer	Finds out the appearance of a series in a graph. The appearance settings for individual data points can override the series setting, so the values obtained from this function may not reflect the current state of the graph. There are several syntaxes, depending on what you want to get: colors; line style, fill pattern, or symbol; or whether the series is an overlay.
GetParent	PowerObject	Returns a reference to the name of the parent object.
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.

Graph function	Data type returned	Description
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.
ObjectAtPointer	GrObject Type	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	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. <hr/> Platform note For information on the file formats which each platform supports, see the <i>PowerScript Reference</i> . <hr/>
SeriesCount	Integer	Determines how many series there are in the graph.

Graph function	Data type returned	Description
SeriesName	String	Obtains the name of the specified series in the graph.
SetDataPieExplode	Integer	Explodes a pie slice in a pie graph.
SetDataStyle	Integer	Specifies the appearance of a data point in a graph. There are several syntaxes, depending on what you want to set: colors, line style and width, or fill pattern or symbol.
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 a change in its properties.
SetSeriesStyle	Integer	Specifies the appearance of a series in a graph. There are several syntaxes, depending on what you want to change: colors, line style and width, fill pattern or symbol, or whether the series is an overlay.
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.

grAxis object

The PowerBuilder class `grAxis` is used as part of a graph control. Each graph has three `grAxis` objects: Category, Series, and Values.

To refer to a property of an axis, use this syntax:

```
graphcontrolname.axisname.property
```

For example:

```
gr_1.Series.AutoScale
```

A `grAxis` object has no events.

Properties

grAxis property	Data type	Description
AutoScale	Boolean	Specifies whether PowerBuilder scales the axis automatically. Values are: <ul style="list-style-type: none"> ◆ True — Automatically scale ◆ False — Do not automatically scale
DataType	grAxisDataType (enumerated)	Specifies the data type of the axis. Values are: AdtDate! AdtDateTime! AdtDefault! AdtDouble! AdtText! AdtTime!
DispAttr	grDispAttr (object)	Settings that control the appearance of the text that labels the axis divisions.
DisplayEveryNLabels	Integer	Specifies which major axis divisions to label. For example, 2 means label every other tick mark. Use 0 to let the graph select the optimum number of labels to use. If the labels are too long, they are clipped.

grAxis property	Data type	Description
DropLines	LineStyle (enumerated)	Specifies the type of drop line for the axis. Values are: <ul style="list-style-type: none"> ◆ Transparent! — No line appears ◆ Continuous! — A solid line ◆ Dash! ◆ Dot! ◆ DashDot! ◆ DashDotDot!
Frame	LineStyle (enumerated)	Specifies the type of line used for the frame. See DropLines in this table for the list of values.
Label	String	Specifies the axis label.
LabelDispAttr	grDispAttr (object)	Settings that control the appearance of the axis label text.
MajorDivisions	Integer	Specifies the number of major divisions on the axis.
MajorGridLine	LineStyle (enumerated)	Specifies the type of line for the major grid. See DropLines in this table for the list of values.
MajorTic	grTicType (enumerated)	Specifies the type of the major tick marks. Values are: NoTic! Inside! Outside! Straddle!
MaximumValue	Double	Specifies the maximum value for the axis when its data type is numeric.
MaxValDateTime	DateTime	Specifies the maximum value for the axis when its data type is date or time.
MinimumValue	Double	Specifies the minimum value for the axis when its data type is numeric.

grAxis property	Data type	Description
MinorDivisions	Integer	Specifies the number of minor divisions on the axis.
MinorGridLine	LineStyle (enumerated)	Specifies the type of line for the minor grid. See DropLines in this table for the list of values.
MinorTic	grTicType (enumerated)	Specifies the type of the minor tick marks. Values are: NoTic! Inside! Outside! Straddle!
MinValDateTime	DateTime	Specifies the minimum value for the axis when its data type is date or time.
OriginLine	LineStyle (enumerated)	Specifies the type of origin line for the axis. See DropLines in this table for the list of values.
PrimaryLine	LineStyle (enumerated)	Specifies the type of primary line for the axis. See DropLines in this table for the list of values.
RoundTo	Double	Specifies the value to which you want to round the axis values.

grAxis property	Data type	Description
RoundToUnit	grRoundToType (enumerated)	<p>Specifies the units for the rounding value. The units must be appropriate for the axis data type. Values are listed below.</p> <p><i>For an axis of any data type:</i></p> <p>RndDefault!</p> <p><i>For an axis of any numeric data type:</i></p> <p>RndNumber!</p> <p><i>For an axis of type date or DateTime:</i></p> <p>RndYears! RndMonths! RndDays!</p> <p><i>For an axis of type time or DateTime:</i></p> <p>RndHours! RndMinutes! RndSeconds! RndMicroseconds!</p>
ScaleType	grScaleType (enumerated)	<p>Specifies the type of scale used for the axis. Values are:</p> <p>Linear! Log10! Loge!</p>
ScaleValue	grScaleValue (enumerated)	<p>Specifies the scale of values on the axis. Values are:</p> <p>Actual! Cumulative! Percentage! CumPercent!</p>
SecondaryLine	LineStyle (enumerated)	<p>Specifies the type of secondary line for the axis.</p> <p>See DropLines in this table for the list of values.</p>
ShadeBackEdge	Boolean	<p>Specifies whether the back edge of the axis is shaded. Values are:</p> <ul style="list-style-type: none"> ◆ True — Shaded ◆ False — Not shaded

Functions

grAxis function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object	Returns the type of the object.

grDispAttr object

The PowerBuilder class grDispAttr is used to specify the appearance of text objects on a graph. There are grDispAttr objects for graph Titles, Legends, Pie Graph text, and two (DispAttr and LabelDispAttr) for each of the three axes (Category, Series, and Value) in a graph.

To refer to a property of a grDispAttr property, use this syntax:

graphcontrolname.axisname.grdispattrname.property

For example:

```
gr_1.Series.DispAttr.Italic  
gr_1.Category.LabelDispAttr.DisplayExpression
```

A grDispAttr object has no events.

Properties

grDispAttr property	Data type	Description
Alignment	Alignment (enumerated)	Specifies the alignment of the text. Values are: Center! Justify! Left! Right!
AutoSize	Boolean	Specifies whether the text element should be autosized according to the amount of text being displayed. Values are: ◆ True — Autosize ◆ False — Do not autosize
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215. For more information about color, see the <i>PowerScript Reference</i> .

grDispAttr property	Data type	Description
DisplayExpression	String	An expression whose value is the label for the graph component. The default expression is the property containing the text for the graph component. The expression can include the text property and add other variable text.
Escapement	Integer	Specifies the rotation for the baseline of the text in tenths of a degree. 0 is horizontal. A value of 900 rotates the text 90 degrees; a value of 2700 rotate the text 270 degrees.
FaceName	String	Specifies a typeface name (for example, ARIAL) to use for the text.
FillPattern	FillPattern (enumerated)	Specifies the fill pattern for the text. Values are: BDiagonal! Diamond! FDiagonal! Horizontal! Solid! Square! Vertical! FDiagonal! is lines going from the lower left to the upper right. BDiagonal! is lines going from the upper left to the lower right.
FontCharSet	FontCharSet (enumerated)	Specifies the font character set to be used. Values are: ANSI! ChineseBig5! DefaultCharSet! HangEul! OEM! ShiftJIS! Symbol!

grDispAttr property	Data type	Description
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) for the text. Values are: AnyFont! Decorative! Modern! Roman! Script! Swiss!
FontPitch	FontPitch (enumerated)	Specifies the pitch (character spacing) for the text. Values are: Default! Fixed! Variable!
Format	String	Specifies the display format for the text.
Italic	Boolean	Specifies whether the text is italic. Values are: ◆ True — Text is italic ◆ False — Text is not italic
TextColor	Long	Specifies the color of the text. The color is a numeric value: -2 to 16,777,215. For more information about color, see the <i>PowerScript Reference</i> .
TextSize	Integer	Specifies the point size of the text. 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 is underlined. Values are: ◆ True — Text is underlined ◆ False — Text is not underlined
Weight	Integer	Specifies the stroke weight of the text. Sample values are 400 for normal or 700 for bold.

Functions

grDispAttr function	Data type returned	Description
ClassName	String	Returns the name assigned to the user object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object	Returns the type of the user object.

GroupBox control

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.

A GroupBox has no events.

Properties

GroupBox property	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>PowerScript Reference</i> .
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
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. Values are: <ul style="list-style-type: none"> ◆ 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.

GroupBox property	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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
Enabled	Boolean	<p>Specifies whether the text of the GroupBox is visible or grayed. Values are:</p> <ul style="list-style-type: none"> ◆ True — Text is visible ◆ False — Text is grayed
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. Values are:</p> <p>ANSI! ChineseBig5! DefaultCharSet! Hangeul! OEM! ShiftJIS! Symbol!</p>
FontFamily	FontFamily (enumerated)	<p>Specifies the font family (type style) used for the text in the control. Values are:</p> <p>AnyFont! Decorative! Modern! Roman! Script! Swiss!</p>

GroupBox property	Data type	Description
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control. Values are: Default! Fixed! Variable!
Height	Integer	Specifies the height of the control, in PowerBuilder units.
Italic	Boolean	Indicates whether the text in the control is italic. Values are: ◆ True — Text is italic ◆ False — Text is not italic
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
RightToLeft	Boolean	Specifies that characters should be displayed in right-to-left order. The application must be running on a Hebrew or Arabic version of PowerBuilder under an operating system that supports right-to-left display. Values are: ◆ True — Characters display in right-to-left order ◆ False — Characters display in left-to-right order
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 title.
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>PowerScript 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.

GroupBox property	Data type	Description
Underline	Boolean	Indicates whether the text in the control is underlined. Values are: <ul style="list-style-type: none"> ◆ True — Text is underlined ◆ False — Text is not underlined
Visible	Boolean	Specifies whether the control is visible. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
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.

Functions

GroupBox function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
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.

GroupBox function	Data type returned	Description
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 properties.
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 control

An HScrollBar is a horizontal bar with arrows at either end and a scroll box. Typically, you would use an HScrollBar control as a slider control for users to specify a value on a continuous scale, or as a way to graphically display information to the user.

Usage note

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

Properties

HScrollBar property	Data type	Description
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order. Values are: <ul style="list-style-type: none"> ◆ True — Control moved to top ◆ False — Control not moved to top
DragAuto	Boolean	Specifies whether PowerBuilder puts the control automatically into Drag mode. Values are: <ul style="list-style-type: none"> ◆ 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.

HScrollBar property	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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
Height	Integer	Specifies the height of the control, in PowerBuilder units.
MaxPosition	Integer	Specifies the value of the Position property when the scroll box is at the right edge of the control.
MinPosition	Integer	Specifies the value of the Position property 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	<p>Specifies whether PowerBuilder uses the standard height for the control. Values are:</p> <ul style="list-style-type: none"> ◆ True — Standard height used for control ◆ False — Standard height not enforced 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.

HScrollBar property	Data type	Description
Visible	Boolean	Specifies whether the control is visible. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
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

HScrollBar 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 property to determine the new location).
Other	When a Windows message occurs that is not a PowerBuilder event.

HScrollBar event	Occurs
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

HScrollBar function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
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 properties.
Show	Integer	Makes the control visible.

HScrollBar function	Data type returned	Description
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.

Inet

The Inet object provides the ability to display a Web page in the default browser, access the HTML for a specified page, and send data to a CGI, ISAPI, or NSAPI program.

Properties

Inet property	Data type	Description
ClassDefinition	PowerObject	An object of type ClassDefinition describing the Inet class.

Events

Inet event	Occurs
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.

Functions

Inet function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
GetURL	Integer	Returns HTML for the specified URL.
HyperLinkToURL	Integer	Opens the default Web browser, displaying the specified URL.

Inet function	Data type returned	Description
PostEvent	Boolean	Adds an event to the end of the message queue for the object
PostURL	Integer	Performs an HTTP Post, allowing a PowerBuilder application to send a request through CGI, NSAPI, or ISAPI.
TriggerEvent	Integer	Triggers a specific event in the object and executes the script for the event.
TypeOf	Object	Returns the type of the object.

InternetResult

The InternetResult object acts as a buffer, receiving and caching asynchronous data, as it is returned via the Internet in response to the GetURL and PostURL function calls. The InternetResult object also provides the ability to process this data.

To use an InternetResult object, create a standard class user object that defines an InternetData function to process the passed HTML.

Properties

InternetResult property	Data type	Description
ClassDefinition	PowerObject	An object of type ClassDefinition describing the InternetResult class.

Events

InternetResult event	Occurs
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.

Functions

InternetResult function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.

InternetResult function	Data type returned	Description
GetURL	Integer	Returns HTML for the specified URL.
InternetData	Integer	Processes the HTML data returned by a GetURL or PostURL function.
PostEvent	Boolean	Adds an event to the end of the message queue for the object.
TriggerEvent	Integer	Triggers a specific event in the object and executes the script for the event.
TypeOf	Object	Returns the type of the object.

Line control

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

A line cannot be selected and therefore has no events.

Properties

Line property	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>PowerScript Reference</i> .
LineStyle	LineStyle (enumerated)	Specifies the style of the line. Values are: Continuous! Dash! DashDot! DashDotDot! Dot! Transparent!
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!.
Tag	String	Specifies the tag value assigned to the control.

Line property	Data type	Description
Visible	Boolean	Specifies whether the control is visible. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible

Functions

Line function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
Hide	Integer	Makes the control invisible.
Resize	Integer	Changes the length of the control (changes the settings of the BeginX BeginY, EndX, and EndY properties).
Show	Integer	Makes the control visible.
TypeOf	Object	Returns the type of the control.

ListBox control

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) scrollbars.

Properties

ListBox property	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. For more information about color, see the <i>PowerScript Reference</i> .
Border	Boolean	Specifies whether the control has a border. Values are: <ul style="list-style-type: none"> ◆ True — Control has a border ◆ False — Control does not have a border
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order. Values are: <ul style="list-style-type: none"> ◆ True — Control moved to top ◆ False — Control not moved to top

ListBox property	Data type	Description
DisableNoScroll	Boolean	<p>Specifies behavior of a scrollbar. Values are:</p> <ul style="list-style-type: none"> ◆ True — The scrollbar will always be visible but will be disabled when all the items can be accessed without it. ◆ False — The scrollbar will be displayed only if it is necessary (based on the number of items and the height of the listbox).
DragAuto	Boolean	<p>Specifies whether PowerBuilder puts the control automatically into Drag mode. Values are:</p> <ul style="list-style-type: none"> ◆ 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	<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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control can be selected ◆ False — Control cannot be selected

ListBox property	Data type	Description
ExtendedSelect	Boolean	<p>Specifies whether users can select multiple items in the listbox at one time. Values are:</p> <ul style="list-style-type: none"> ◆ 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 CTRL+ CLICK on multiple items. ◆ False — Users cannot select multiple items. <hr/> <p>Used with MultiSelect The MultiSelect property 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	<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. Values are:</p> <p>ANSI! ChineseBig5! DefaultCharSet! Hangeul! OEM! ShiftJIS! Symbol!</p>
FontFamily	FontFamily (enumerated)	<p>Specifies the font family (type style) used for the text in the control. Values are:</p> <p>AnyFont! Decorative! Modern! Roman! Script! Swiss!</p>

ListBox property	Data type	Description
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control. Values are: Default! Fixed! Variable!
Height	Integer	Specifies the height of the control, in PowerBuilder units.
HScrollBar	Boolean	Specifies whether a horizontal scrollbar displays. Values are: ◆ True — Horizontal scrollbar displays ◆ False — Horizontal scrollbar does not display
Italic	Boolean	Specifies whether the text in the control is italic. Values are: ◆ True — Text is italic ◆ False — Text is not italic
Item[]	String	Specifies the items in the control.
MultiSelect	Boolean	Specifies whether users can select multiple items in the ListBox at one time. Values are: ◆ True — Users can select multiple items ◆ False — Users cannot select multiple items
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
RightToLeft	Boolean	Specifies that characters should be displayed in right-to-left order. The application must be running on a Hebrew or Arabic version of PowerBuilder under an operating system that supports right-to-left display. Values are: ◆ True — Characters display in right-to-left order ◆ False — Characters display in left-to-right order

ListBox property	Data type	Description
Sorted	Boolean	Specifies whether the items in the ListBox are sorted. Values are: <ul style="list-style-type: none"> ◆ True — Items are sorted ◆ False — Items are not sorted
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. For more information about color, see the <i>PowerScript 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. Values are: <ul style="list-style-type: none"> ◆ True — Text is underlined ◆ False — Text is not underlined
Visible	Boolean	Specifies whether the control is visible. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible

ListBox property	Data type	Description
VScrollBar	Boolean	Specifies whether a vertical scrollbar is displayed on the right of the ListBox. Values are: <ul style="list-style-type: none"> ◆ True — Vertical scrollbar is displayed ◆ False — Vertical scrollbar is not displayed
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

ListBox 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).

ListBox event	Occurs
Other	When a Controls 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

ListBox function	Data type returned	Description
AddItem	Integer	Adds a new item to the end of the ListBox. If the Sorted property of the control is True, the items are sorted again after the item is added. The AddItem function does not update the Item[] property of this control.
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.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
Hide	Integer	Makes the control invisible.

ListBox function	Data type returned	Description
InsertItem	Integer	Adds a new item to the ListBox before the item indicated by the index. If the Sorted property 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.
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	Finds and highlights an item in the control. Use Syntax 1 when you know the text of the item but not its position. Use Syntax 2 when you know the position of the item in the control's list or you want to clear the current selection. SelectItem has no effect on a ListBox whose MultiSelect property is True. Instead, use SetState to select items without affecting the selected state of other items in the list.
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 properties.

ListBox function	Data type returned	Description
SetState	Integer	Sets the state (highlighted or not highlighted) of the item indicated by the specified index. SetState works only for multiselect controls (those for which the MultiSelect property 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 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.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.

Listview control

A ListView displays list information to the user. Each item of the ListView consists of text and pictures, which can be manipulated during application runtime.

Platform information

On Windows 3.1 (16-bit) there is a limitation on the total number of rows within a ListView control. The rule is as follows:

$$(\text{no of rows}) \times [(\text{image height in pixels}) + (\text{margin height in pixels})] < 32,767$$

Properties

ListView property	Data type	Description
Accelerator	Integer	Specifies the ASCII value of the accelerator key assigned for the control.
AutoArrange	Boolean	Specifies whether PowerBuilder arranges icons automatically in large and small icon views.
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215.
Border	Boolean	Specifies whether the control has a border. Values are: <ul style="list-style-type: none"> ◆ True — Control has a border ◆ False — Control does not have a border
BorderStyle	BorderStyle (enumerated)	Specifies the border style of the control. Values are: <ul style="list-style-type: none"> StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order.

ListView property	Data type	Description
ButtonHeader	Boolean	Specifies whether the column titles in report view appear as buttons.
DeleteItems	Boolean	Specifies whether the user can delete a ListView item from a ListView control by pressing DELETE.
Dragauto	Boolean	Specifies whether PowerBuilder puts the control automatically into Drag mode. Values are: <ul style="list-style-type: none"> ◆ 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	<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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
EditLabels	Boolean	Specifies whether the user can edit the labels in a control by clicking on a selected item. Note that the item must be selected first, by clicking on it.

ListView property	Data type	Description
Enabled	Boolean	Specifies whether the control is enabled (can be clicked). Values are: <ul style="list-style-type: none"> ◆ True — Control can be clicked ◆ False — Control cannot be clicked
ExtendedSelect	Boolean	Specifies whether users can select multiple items in the listbox at one time. Values are: <ul style="list-style-type: none"> ◆ True — Users can select multiple items by selecting outside all items and dragging to create a rectangle enclosing the desired items; by using CLICK/ SHIFT+ CLICK to select a sequential group of items; or by using CTRL+ CLICK on multiple items. ◆ False — Users cannot select multiple items.
FaceName	String	Specifies the name of the typeface in which the text of the control displays (for example, Arial or Courier).
FixedLocations	Boolean	Specifies whether the user cannot drag items to new positions in a control.
FontCharSet	FontCharSet (enumerated)	Specifies the font character set used for the text in the control. The application must be running on an appropriate version of PowerBuilder under an operating system that supports the selected character set. Values are: <ul style="list-style-type: none"> ANSI! ChineseBig5! DefaultCharSet! Hangeul! OEM! ShiftJIS! Symbol!

ListView property	Data type	Description
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) used for the text in the control. Values are: AnyFont! Decorative! Modern! Roman! Script! Swiss!
FontPitch	FontPitch (enumerated)	Specifies the font pitch used for the text in the control. Values are: Default! Fixed! Variable!
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. Values are: ◆ True — Text does not stay highlighted ◆ False — Text stays highlighted
Italic	Boolean	Specifies whether the text in the control is italic. Values are: ◆ True — Text is italic ◆ False — Text is not italic
Item[]	String	Specifies the items in the control. Not updated after initialization.
ItemPictureIndex[]	Integer	Identifies the picture associated with the item. The picture index itself is associated with a specific icon, bitmap, or cursor. Not updated after initialization.

ListView property	Data type	Description
LabelWrap	Boolean	<p>Specifies whether long labels wrap under the ListView item in large icon and small icon views. Values are:</p> <ul style="list-style-type: none"> ◆ True — Labels will wrap ◆ False — Labels will not wrap <p>Label Wrap does not apply to report views and list views.</p>
LargePictureHeight	Integer	<p>Specifies the size, in pixels, for the height of the picture used in the large icon view.</p> <p>In a script, this value can only be set before a large picture has been added to the large picture index list.</p> <p>If the large picture height is 0, PowerBuilder uses the height of the first picture added to the large picture index.</p>
LargePictureMask Color	Long	<p>Specifies the color to be transparent when used in a large icon view. This color is used when the picture is added at initialization or with the function AddLargePicture.</p>
LargePictureName[]	String	<p>Specifies the name of the picture used in large icon view. The picture can be an icon, cursor, or bitmap supplied by the user or a stock picture from the PowerBuilder library. Not updated after initialization.</p>
LargePictureWidth	Integer	<p>Specifies the size, in pixels, for the width of the picture used in the large icon view.</p> <p>In a script, this value can only be set before a large picture has been added to the large picture index list.</p> <p>If the large picture width is 0, PowerBuilder uses the width of the first picture added to the large picture index.</p>

ListView property	Data type	Description
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
Scrolling	Boolean	Specifies whether the user can scroll vertically when not all of the items in a control are visible. Values are: <ul style="list-style-type: none"> ◆ True — Scrolling is enabled ◆ False — Scrolling is disabled
ShowHeader	Boolean	Specifies whether column titles appear in a report view. Values are: <ul style="list-style-type: none"> ◆ True — Titles appear in a report view ◆ False — Titles do not appear in a report view
SmallPictureHeight	Integer	Specifies the size, in pixels, for the height of the picture used in the small icon view. In a script, this value can only be set before a small picture has been added to the small picture index list. If the small picture height is 0, PowerBuilder uses the height of the first picture added to the small picture index.
SmallPictureMask Color	Long	Specifies the color to be transparent when used in a small icon view. Used when the picture is added at initialization or with the function AddSmallPicture.
SmallPictureName[]	String	Specifies the name of the picture used in small icon view. The picture can be an icon, cursor, or bitmap supplied by the user or a stock picture from the PowerBuilder library. Not updated after initialization.

ListView property	Data type	Description
SmallPictureWidth	Integer	<p>Specifies the size, in pixels, for the width of the picture used in the small icon view.</p> <p>In a script, this value can only be set before a small picture has been added to the small picture index list.</p> <p>If the small picture width is 0, PowerBuilder uses the width of the first picture added to the small picture index.</p>
SortType	grSortType	<p>Specifies whether items are sorted alphabetically based on the item label. Values are:</p> <p>Ascending! Descending! UserDefined! Unsorted!</p>
StatePictureHeight	Integer	<p>Specifies the size, in pixels, for the height of the state picture.</p> <p>In a script, this value can only be set before a state picture has been added to the state picture index list.</p> <p>If the state picture height is 0, PowerBuilder uses the height of the first picture added to the state picture index list.</p>
StatePictureMask Color	Long	<p>Specifies the color to be transparent when used in a state picture. Used when the picture is added at initialization or with the function AddStatePicture.</p>
StatePictureName[]	String	<p>Specifies the name of the picture used in as the state picture. The picture can be an icon, cursor, or bitmap supplied by the user or a stock picture from the PowerBuilder library. Not updated after initialization.</p>

ListView property	Data type	Description
StatePictureWidth	Integer	<p>Specifies the size, in pixels, for the width of the state picture.</p> <p>In a script, this value can only be set before a state picture has been added to the state picture index list.</p> <p>If the state picture width is 0, PowerBuilder uses the width of the first picture added to the state picture index list.</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>
TextColor	Long	<p>Specifies the numeric value of the color used for text: -2 to 16,777,215.</p>
TextSize	Integer	<p>Specifies the size of the text in the control, in points.</p> <p>For backward compatibility, the size is stored as a negative number; for example, 10-point text size is stored as -10.</p>
Underline	Boolean	<p>Specifies whether the text in the control is underlined. Values are:</p> <ul style="list-style-type: none"> ◆ True — Text is underlined ◆ False — Text is not underlined

ListView property	Data type	Description
View	ListViewView	<p>Specifies the layout of the ListBox. Valid values are:</p> <ul style="list-style-type: none"> ◆ ListViewLargeIcon! — Items are arranged from left to right. Uses large pictures. ◆ ListViewSmallIcon! — Items are arranged from left to right. Uses small pictures. ◆ ListViewList! — Items are arranged from top to bottom. Uses small pictures. ◆ ListViewReport! — Items are arranged from top to bottom. Uses small pictures. Additional columns of information can be associated with each item. Note that at least one column must be created to view data in this view.
Visible	Boolean	<p>Specifies whether the control is visible. Values are:</p> <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
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

ListView event	Occurs
BeginDrag	When the user begins a drag operation with the left mouse button. If the DragAuto property is set to True, the drag will begin automatically. If the DragAuto property is set to False, the drag operation must be done programmatically.
BeginLabelEdit	When the user starts to edit a ListView item label. Return 1 to prevent setting to the new text. Return 0 to accept the new text.
BeginRightDrag	When the user begins a drag operation with the right mouse button. If the DragAuto property is set to True, the drag will begin automatically. If the DragAuto property is set to False, the drag operation must be done programmatically.
Clicked	When the control is clicked.
ColumnClick	When the column is clicked
Constructor	When the object is created, immediately before the Open event occurs in the window.
DeleteAllItems	When all items in a ListView are deleted.
DeleteItem	When a ListView item is deleted.
Destructor	When the object is destroyed, immediately after the Close event occurs in the window.
DoubleClicked	When the control is double-clicked.
DragDrop	When a dragged control is dropped on the ListView control.
DragEnter	When a dragged control enters the control, including entering the narrow border around the display area.
DragLeave	When a dragged control leaves the control, including leaving by crossing into the tab page display area.
DragWithin	When a dragged control is within the control but not on a ListView item.
EndLabelEdit	When the user finishes editing a ListView item label. Return 1 to prevent setting to the new text. Return 0 to accept the new text.

ListView event	Occurs
GetFocus	Just before the control receives focus (before it is selected and becomes active).
InsertItem	When an item is being inserted.
ItemChanged	When an item has changed.
ItemChanging	When an item is changing. Return 1 to prevent the change, or 0 to accept the change.
Key	When the user presses a key.
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PowerBuilder event.
RightClicked	When the control is right-clicked.
RightDoubleClicked	When the control is right-double-clicked.
Sort	When two items are compared. Return codes: 1 — if item 1 > item 2 0 — if item 1 = item 2 -1 — if item 1 < item 2

Functions

ListView function	Data type returned	Description
AddColumn	Integer	Adds a column to a ListView control report view.
AddItem	Integer	Adds an item to a ListView control.
AddLargePicture	Integer	Adds an icon, cursor, or bitmap to the large image list.
AddSmallPicture	Integer	Adds an icon, cursor, or bitmap to the small image list.
AddstatePicture	Integer	Adds an icon, cursor, or bitmap to the state image list.

ListView function	Data type returned	Description
Arrange	Integer	Arranges the items in a ListView control large or small icon view.
ClassName	String	Returns the name of the control.
DeleteColumn	Integer	Deletes a column from a ListView control.
DeleteColumns	Integer	Deletes all columns from a ListView control.
DeleteItem	Integer	Deletes an item from a ListView control.
DeleteItems	Integer	Deletes all items from a ListView control.
DeleteLargePicture	Integer	Deletes a specified icon, cursor, or bitmap from the large image list.
DeleteLargePictures	Integer	Deletes all icons, cursors, and bitmaps from the large image list.
DeleteSmallPicture	Integer	Deletes a specified icon, cursor, or bitmap from the small image list.
DeleteSmallPictures	Integer	Deletes all icons, cursors, and bitmaps from the small image list.
DeleteStatePicture	Integer	Deletes a specified icon, cursor, or bitmap from the state image list.
DeleteStatePictures	Integer	Deletes all icons, cursors, and bitmaps from the large state list.
Drag	Integer	Starts or ends the dragging of a ListView item.
EditLabel	Integer	Starts editing a specific ListView item label.
FindItem	Integer	Searches for the next item that satisfies the specified search criteria.
GetColumn	Integer	Syntax 1: Does not apply to a ListView control. Syntax 2: Returns the properties of a specified column in a ListView control report view.

ListView function	Data type returned	Description
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetItem	Integer	Retrieves information for a specified item.
GetOrigin	Integer	Finds the X and Y coordinates of the upper-left corner of the ListView item.
GetParent	PowerObject	Returns a reference to the name of the parent object.
Hide	Integer	Hides the specified ListView item.
InsertColumn	Integer	Inserts a column into a ListView control report view.
InsertItem	Integer	Inserts an item into a ListView control.
Move	Integer	Moves a control or object to a specified location.
PointerX	Integer	Determines the distance from the left edge of an object to the pointer location.
PointerY	Integer	Determines the distance from the top edge of an object to the pointer location.
PostEvent	Boolean	Adds the event to the end of the event queue of an object.
Print	Integer	Includes this object in a print job. Only the part visible on the screen is printed.
Resize	Integer	Resizes a control to the specified dimensions.
SelectedIndex	Integer	Returns the number of the selected item in a ListView control.
SetColumn	Integer	Syntax 1: Does not apply to a ListView control. Syntax 2: Sets the properties of a particular column in a ListView control report view.

ListView function	Data type returned	Description
SetFocus	Integer	Sets focus for a specified object or control.
SetItem	Integer	Sets the values for a given ListView item.
SetOverlayPicture	Integer	Maps a picture index to an overlay picture index. Only four overlay picture indexes are available.
SetPosition	Integer	Sets the position of the ListView control in the front-to-back order within a window.
SetRedraw	Integer	Controls the automatic redraw of an object after its properties have changed.
Show	Integer	Makes an object or control visible if it is hidden. If the object is already visible, Show brings it to the top.
Sort	Integer	Sorts the items in a ListView control.
TotalColumns	Integer	Returns the number of columns in a ListView control report view.
TotalItems	Integer	Returns the number of items in a ListView control.
TotalSelected	Integer	Returns the number of selected items in a ListView control.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.

ListViewItem object

A ListViewItem object is a system structure that populates a ListView control. ListViewItems have no events.

Properties

ListViewItem property	Data type	Description
CutHighlighted	Boolean	Specifies whether the item is the target of a cut operation. Values are: <ul style="list-style-type: none"> ◆ True — The item is the target of a cut operation ◆ False — The item is not the target of a cut operation
Data	Any	Assigns any user-defined data to a ListViewItem.
DropHighlighted	Boolean	Specifies if the item is the target of a DragDrop operation. Values are: <ul style="list-style-type: none"> ◆ True — The item is the target of a DragDrop operation ◆ False — The item is not the target of a DragDrop operation
HasFocus	Boolean	Specifies if the item has focus. Values are: <ul style="list-style-type: none"> ◆ True — The item has focus ◆ False — The item does not have focus
ItemX	Integer	Identifies the X location of the item relative to the upper left corner of the control.
ItemY	Integer	Identifies the Y location of the item relative to the upper-left corner of the control.
Label	String	Identifies the string label associated with the item.
OverlayPictureIndex	Integer	Identifies the overlay picture associated with the item.

ListViewItem property	Data type	Description
PictureIndex	Integer	Identifies the large and small picture associated with the item.
Selected	Boolean	Specifies if the item is selected. Values are: <ul style="list-style-type: none">◆ True — The item is selected◆ False — The item is not selected
StatePictureIndex	Integer	Identifies the state picture associated with the item.

Functions

ListViewItem function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the parent of the object.
TypeOf	Object	Returns the type of the object.

mailFileDescription

The mailFileDescription object is a system structure containing information about an attachment file to a mail message. A mailFileDescription object has no events.

Platform note

The mail functions are only available on the Windows platform.

Properties

mailFileDescription property	Data type	Description
FileType	mailFileType (enumerated)	Specifies the type of the attachment file. Values are: <ul style="list-style-type: none"> ◆ mailAttach! — data file ◆ mailOLE! — embedded OLE object ◆ mailOLEStatic! — static OLE object
Filename	String	Specifies the name of the attachment file.
Pathname	String	Specifies the full path of the attachment file.
Position	Unsignedlong	Specifies the position of the attachment file within the message body. (Read-only.)

Functions

mailFileDescription function	Data type returned	Description
Classname	String	Returns the class of an object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.

mailFileDescription function	Data type returned	Description
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object	Returns the type of the object.

mailMessage

The mailMessage object is a system structure containing information about a specific mail message. A mailMessage object has no events.

Platform note

The mail functions are only available on the Windows platform.

Properties

mailMessage property	Data type	Description
AttachmentFile[]	mailFile Description	Specifies the file attachment for the current message. A mailFileDescription array contains information about an attachment file.
ConversationID	String	Specifies the conversation thread ID for the current message.
DateReceived	String	Indicates the date on which the current message was received.
MessageSent	Boolean	Indicates whether the current message has been sent to the mail server. (Read-only at runtime only.) <ul style="list-style-type: none"> ◆ True — Message has been sent to mail server ◆ False — Message has not yet been sent
MessageType	String	Indicates the type of the current message. A value other than null or an empty string indicates use by an application other than interpersonal mail. (Runtime only.)

mailMessage property	Data type	Description
NoteText	String	Specifies the content of the message body. (Runtime only.)
ReceiptRequested	Boolean	Indicates whether a return receipt is requested for the current message. (Runtime only.) <ul style="list-style-type: none"> ◆ True — Return receipt requested ◆ False — Return receipt not requested
Recipient[]	mailRecipient	Specifies the recipient of the current message.
Subject	String	Specifies the subject line, displayed in the message header, for the current message.
Unread	Boolean	Indicates whether or not the message has been read. (Read-only at runtime only.) <ul style="list-style-type: none"> ◆ True — Message has not been read ◆ False — Message has been read

Functions

mailMessage function	Data type returned	Description
Classname	String	Returns the class of the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object	Returns the type of the object.

mailRecipient

The mailRecipient object is a system structure containing information about the recipient of a mail message. You populate the mailRecipient structure using the mailAddress function. A mailRecipient object has no events.

Platform note

The mail functions are only available on the Windows platform.

FOR INFO For information about the mailAddress function, see the *PowerScript Reference*.

Properties

mailRecipient property	Data type	Description
Address	String	Specifies the electronic mail address of the current mail recipient. (Runtime only.)
EntryID	Blob	Binary entry identifier information used internally.
Name	String	Specifies the name of the current mail recipient. (Runtime only.)
RecipientType	mailRecipientType (enumerated)	Specifies the type of the current mail recipient. Values are: mailBCC! mailCC! mailOriginator! mailTo!

Functions

mailRecipient function	Data type returned	Description
Classname	String	Returns the class of the object.

mailRecipient function	Data type returned	Description
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object	Returns the type of the object.

mailSession

The mailSession nonvisual object signs on and establishes a MAPI (messaging application program interface) session.

Platform note

The mail functions are only available on the Windows platform.

Properties

mailSession property	Data type	Description
MessageID[]	String	Contains the IDs of the messages in a user's mail inbox.
SessionID	Long	Contains the handle of the current messaging session.

Events

mailSession event	Occurs
Constructor	When the object is created.
Destructor	When the object is destroyed.

Functions

mailSession function	Data type returned	Description
Classname	String	Returns the class of the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.

mailSession function	Data type returned	Description
GetParent	PowerObject	Returns a reference to the name of the parent object.
mailAddress	mailReturnCode	Updates the mailRecipient array for a mail message.
mailDeleteMessage	mailReturnCode	Deletes a mail message from the user's electronic mail inbox.
mailGetMessages	mailReturnCode	Populates the MessageID array of a mailSession object with the message IDs in the user's inbox.
mailHandle	UnsignedLong	Obtains the handle of a mailSession object.
mailLogoff	mailReturnCode	Ends the mail session, breaking the connection between the PowerBuilder application and mail.
mailLogon	mailReturnCode	Establishes a mail session for the PowerBuilder application.
mailReadMessage	mailReturnCode	Opens a mail message whose ID is stored in the mail session's message array.
mailRecipientDetails	mailReturnCode	Displays a dialog box with the specified recipient's address information.
mailResolveRecipient	mailReturnCode	Obtains a valid electronic mail address based on a partial or full user name and optionally updates information in the system's address list if the user has privileges to do so.
mailSaveMessage	mailReturnCode	Creates a new message in the user's inbox or replaces an existing message.
mailSend	mailReturnCode	Sends a mail message.
PostEvent	Integer	Adds an event to the end of the message queue for the object.

mailSession function	Data type returned	Description
TriggerEvent	Integer	Triggers a specified event in the object and executes the script for the event.
TypeOf	Object	Returns the type of the object.

MDIClientobject

An MDI window is a frame window in which you can open multiple document windows (sheets) and move among the sheets.

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

In a custom MDI 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.

An MDIClient object has no associated events.

Properties

MDIClient property	Data type	Description
BackColor	Long	Specifies the numerical value of the background color: -2 to 16,777,215. For more information about color, see the <i>PowerScript Reference</i> .
BringToTop	Boolean	Specifies whether PowerBuilder moves the MDIClient object to the top of the front-to-back order. Values are: <ul style="list-style-type: none"> ◆ True — Object moved to top ◆ False — Object not moved to top
Height	Integer	Specifies the height of the MDIClient object, in PowerBuilder units.
MicroHelpHeight	Integer	Holds the height of the MicroHelp in the MDIClient object. If the style of the MDI window is MDI Frame (no MicroHelp), MicroHelpHeight is 0; otherwise, it is the height of the MicroHelp. You cannot set the value of this property.

MDIClient property	Data type	Description
Tag	String	Specifies the tag value assigned to the MDIClient object.
Visible	Boolean	Specifies whether the MDIClient object is visible. Values are: <ul style="list-style-type: none"> ◆ True — Object is visible ◆ False — Object is not visible
Width	Integer	Specifies the width of the MDIClient object, 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

MDIClient function	Data type returned	Description
ClassName	String	Returns the name assigned to the MDIClient object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
Hide	Integer	Makes the MDIClient object invisible.
Move	Integer	Moves the MDIClient object to a specified location.
Resize	Integer	Changes the size of the MDIClient object.
SetRedraw	Integer	Controls automatic redrawing of the MDIClient object after each change in its properties. To reduce flicker when the user closes all the sheets, set SetRedraw to False.

MDIClient function	Data type returned	Description
Show	Integer	Makes the MDIClient object visible.
TypeOf	Object	Returns the type of the object.

Menu

Typically, menus are lists of items (usually commands or options) that a user can select in the currently active window. Menus can display in either a menu bar or a dropdown or cascading menu.

PowerBuilder provides a Menu system object that is used to develop menus. A Menu object can contain other Menu objects that appear, for example, as the items in a dropdown or cascading menu. When the user clicks a Menu object, a Clicked event is triggered. If there is a dropdown or cascading menu under the clicked object, the script for the Clicked event for the object is executed, and then the menu displays. If there is no menu under the object, the script for the Clicked event for the object is executed.

Properties

Menu property	Data type	Description
Checked	Boolean	Specifies whether the Menu object is selected. The state of the checked property will be reflected in the toolbar button. Values are: <ul style="list-style-type: none"> ◆ True — Object is selected ◆ False — Object is not selected
Enabled	Boolean	Specifies whether the Menu object is enabled (can be clicked). The state of the enabled property will be reflected in the toolbar button. Values are: <ul style="list-style-type: none"> ◆ True — Object is enabled ◆ False — Object is not enabled
Item[]	Menu	Specifies the Menu objects under a Menu object.
MenuItemType	MenuItemType (enumerated)	Allows you to identify special Menu objects that are used differently on different platforms. Values are: <ul style="list-style-type: none"> MenuItemTypeAbout! MenuItemTypeExit! MenuItemTypeNormal!

Menu property	Data type	Description
MergeOption	MenuMergeOption (enumerated)	<p>Specifies how the object is affected when an OLE 2.0 object is activated. Values are:</p> <p>EditMenu! Exclude! FileMenu! HelpMenu! Merge! WindowMenu!</p> <hr/> <p>Platform note This and other OLE functions are only available on the Windows platform.</p> <hr/> <p>For more information about MergeOption, see the chapter on using OLE in <i>Application Techniques</i>.</p>
MicroHelp	String	Specifies the text of the MicroHelp for the object.
ParentWindow	Window	Specifies the window that owns the Menu object.
ShiftToRight	Boolean	<p>Specifies whether the Menu object shifts down or to the right when other Menu objects are added in a descendent menu. Values are:</p> <ul style="list-style-type: none"> ◆ True — Object shifts to right ◆ False — Object shifts down
Shortcut	Integer	Specifies the shortcut key for the Menu object. This property should only be set in the Menu painter. It cannot be set at runtime.
Tag	String	Specifies the tag value assigned to the Menu object.
Text	String	Specifies the text in the Menu object.

Menu property	Data type	Description
ToolBarItemDown	Boolean	<p>Specifies how the toolbar button appears. Values are:</p> <ul style="list-style-type: none"> ◆ True — Toolbar button appears down ◆ False — Toolbar button appears up <p>This property is automatically reset when any button is pressed using the mouse.</p>
ToolBarItemDown Name	String	Specifies the name of the toolbar icon associated with the Menu object when it is down.
ToolBarItemBar Index	Integer	Specifies which toolbar the Menu object is on when multiple toolbars exist. If setting this index results in the object being the first item on a new toolbar, the toolbar is implicitly created. If setting this index results in emptying a toolbar, the toolbar is implicitly destroyed.
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	<p>Specifies whether the toolbar item displays. Values are:</p> <ul style="list-style-type: none"> ◆ True — The toolbar item is visible ◆ False — The toolbar item is not visible

Menu property	Data type	Description
Visible	Boolean	Specifies whether the Menu object is visible. Values are: <ul style="list-style-type: none"> ◆ True — Object is visible ◆ False — Object is not visible

Events

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

Functions

Menu function	Data type returned	Description
Check	Integer	Displays a checkmark next to the Menu object and sets the Checked property.
ClassName	String	Returns the class of the Menu object.
Disable	Integer	Disables (and grays) the Menu object so that it cannot be selected and unsets the Enabled property.
Enable	Integer	Enables the Menu object so that it can be selected and displays it normally (not grayed) and sets the Enabled property.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
Hide	Integer	Makes the Menu object invisible.
PopMenu	Integer	Displays the Menu object at the specified location.

Menu function	Data type returned	Description
PostEvent	Integer	Adds an event to the end of the message queue for the Menu object.
Show	Integer	Makes the Menu object visible.
TriggerEvent	Integer	Triggers a specified event in the Menu object and executes the script for the event.
TypeOf	Object	Returns the type of the control.
Uncheck	Integer	Removes the checkmark next to the Menu object and sets the Checked property to False.

MenuCascade

Menu objects contained within a MenuCascade object appear as a dropdown button palette.

Properties

MenuCascade property	Data type	Description
Checked	Boolean	Specifies whether the Menu object is selected. The state of the checked property will be reflected in the toolbar button. Values are: <ul style="list-style-type: none">◆ True — Object is selected◆ False — Object is not selected
Columns	Integer	Specifies the number of columns in the drop down button palette.
CurrentItem	Menu	Specifies the Menu object that is currently displayed on the main toolbar as the representative (active) button for the drop down button palette.
DropDown	Boolean	Specifies whether the Menu objects contained in the MenuCascade object display as a drop down button palette. Values are: <ul style="list-style-type: none">◆ True — Menu objects contained in the MenuCascade object are displayed as a dropdown button palette.◆ False — Menu objects contained in the MenuCascade object are displayed as normal toolbar items.

MenuCascade property	Data type	Description
Enabled	Boolean	<p>Specifies whether the Menu object is enabled (can be clicked). The state of the enabled property will be reflected in the toolbar button. Values are:</p> <ul style="list-style-type: none"> ◆ True — Object is enabled ◆ False — Object is not enabled
Item[]	Menu	Specifies the Menu objects under a Menu object.
MenuItemType	MenuItemType (enumerated)	<p>Allows you to identify special menu items that are used differently on different platforms. Values are:</p> <p>MenuItemTypeAbout! MenuItemTypeExit! MenuItemTypeNormal!</p>
MergeOption	MenuMergeOption (enumerated)	<p>Specifies how the object is affected when an OLE 2.0 object is activated. Values are:</p> <p>EditMenu! Exclude! FileMenu! HelpMenu! Merge! WindowMenu!</p> <hr/> <p>Platform note This and other OLE functions are only available on the Windows platform.</p> <hr/> <p>For more information about MergeOption, see the chapter on using OLE in <i>Application Techniques</i>.</p>
MicroHelp	String	Specifies the text of the MicroHelp for the object.
ParentWindow	Window	Specifies the window that owns the Menu object.

MenuCascade property	Data type	Description
ShiftToRight	Boolean	Specifies whether the Menu object shifts down or to the right when other Menu objects are added in a descendent menu. Values are: <ul style="list-style-type: none">◆ True — Object shifts to right◆ False — Object shifts down
Shortcut	Integer	Specifies the shortcut key for the object. This property should only be set in the Menu painter. It cannot be set at runtime.
Tag	String	Specifies the tag value assigned to the Menu object.
Text	String	Specifies the text in the Menu object.
ToolBarItemDown	Boolean	Specifies how the toolbar button appears. Values are: <ul style="list-style-type: none">◆ True — Toolbar button appears down◆ False — Toolbar button appears up This property is automatically reset when any button is pressed using the mouse.
ToolBarItemDown Name	String	Specifies the name of the toolbar icon associated with the Menu object when it is down.
ToolBarItemBarIndex	Integer	Specifies which toolbar the object is on when multiple toolbars exist. If setting this index results in the item being the first item on a new toolbar, the toolbar is implicitly created. If setting this index results in emptying a toolbar, the toolbar is implicitly destroyed.

MenuCascade property	Data type	Description
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. Values are: <ul style="list-style-type: none"> ◆ True — Toolbar item is visible ◆ False — Toolbar item is not visible
Visible	Boolean	Specifies whether the Menu object is visible. Values are: <ul style="list-style-type: none"> ◆ True — Object is visible ◆ False — Object is not visible

Events

MenuCascade event	Occurs
Clicked	When the Menu object is clicked (selected or unselected).
Selected	When the user moves to the Menu object using the arrow keys or the mouse.

Functions

MenuCascade function	Data type returned	Description
Check	Integer	Displays a checkmark next to the Menu object and sets the Checked property.
ClassName	String	Returns the class of the Menu object.
Disable	Integer	Disables (and grays) the Menu object so that it cannot be selected and unsets the Enabled property.
Enable	Integer	Enables the Menu object so that it can be selected, displays it normally (not grayed), and sets the Enabled property.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	Power Object	Returns a reference to the name of the parent object.
Hide	Integer	Makes the Menu object invisible.
PopupMenu	Integer	Displays the Menu object at the specified location.
PostEvent	Integer	Adds an event to the end of the message queue for the Menu object.
Show	Integer	Makes the Menu object visible.
TriggerEvent	Integer	Triggers a specified event in the Menu object and executes the script for the event.
TypeOf	Object	Returns the type of the control.
Uncheck	Integer	Removes the checkmark next to the Menu object and sets the Checked property to False.

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 to specify whether 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.

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

Properties

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

Message property	Data type	Description
Handle	Long	The handle of the window or control. Platform note Handle returns a handle you can use with external functions on the Windows and UNIX platforms.
Number	UnsignedInt	The number that identifies the event (this number comes from Windows).
WordParm	Long	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 a numeric variable.
StringParm	String	A string or a string variable.
PowerObjectParm	PowerObject	Any PowerBuilder object type including structures.

Message property	Data type	Description
Processed	Boolean	<p>A boolean value set in the script for the user-defined event or the Other event. Values are:</p> <ul style="list-style-type: none"> ◆ 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	<p>When Message.Processed is True, specifies the value you want returned to Windows. This property is ignored when Message.Processed is False.</p>

Events

Message event	Occurs
Constructor	When the user object is created.
Destructor	When the user object is destroyed.

Functions

Message function	Data type returned	Description
ClassName	String	Returns the name assigned to the user object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
PostEvent	Boolean	Adds an event to the end of the message queue of the user object.

Message function	Data type returned	Description
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.

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.

Properties

MultiLineEdit property	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. Values are: Center! Justify! Left! Right!
AutoHScroll	Boolean	Specifies whether the control automatically scrolls horizontally when data is entered or deleted. Values are: <ul style="list-style-type: none">◆ True — Control automatically scrolls horizontally◆ False — Control does not automatically scroll horizontally
AutoVScroll	Boolean	Specifies whether the control automatically scrolls vertically when data is entered or deleted. Values are: <ul style="list-style-type: none">◆ True — Control automatically scrolls vertically◆ False — Control wraps
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215. For more information about color, see the <i>PowerScript Reference</i> .

MultiLineEdit property	Data type	Description
Border	Boolean	Specifies whether the control has a border. Values are: <ul style="list-style-type: none">◆ True — Control has a border◆ False — Control does not have a border
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order in the window. Values are: <ul style="list-style-type: none">◆ True — Control moved to top◆ False — Control not moved to top
DisplayOnly	Boolean	Specifies whether the text is display-only and cannot be changed by the user. Values are: <ul style="list-style-type: none">◆ True — Text cannot be changed by user◆ False — Text can be changed by user
DragAuto	Boolean	Specifies whether PowerBuilder will put the control automatically into Drag mode. Values are: <ul style="list-style-type: none">◆ 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.

MultiLineEdit property	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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control can be selected ◆ False — Control cannot be selected
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. Values are:</p> <p>ANSI! ChineseBig5! DefaultCharSet! Hangeul! OEM! ShiftJIS! Symbol!</p>
FontFamily	FontFamily (enumerated)	<p>Specifies the font family (type style) used for the text in the control. Values are:</p> <p>AnyFont! Decorative! Modern! Roman! Script! Swiss!</p>

MultiLineEdit property	Data type	Description
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control. Values are: Default! Fixed! Variable!
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. Values are: ◆ True — Text does not stay highlighted ◆ False — Text stays highlighted
HScrollBar	Boolean	Specifies whether a horizontal scrollbar displays. Values are: ◆ True — Horizontal scrollbar displays ◆ False — Horizontal scrollbar does not display
IgnoreDefaultButton	Boolean	Specifies whether the Clicked event for the window's Default command button is triggered when user presses ENTER. Values are: ◆ True — Do not trigger Clicked event; add new line in MultiLineEdit control. ◆ False — (Default) Trigger Clicked event; do not add new line in MultiLineEdit control.
Italic	Boolean	Specifies whether the text in the control is italic. Values are: ◆ True — Text is italic ◆ False — Text is not italic
Limit	Integer	Specifies the maximum number of characters (0 to 32,767) that can be entered in the control (0 means unlimited).

MultiLineEdit property	Data type	Description
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
RightToLeft	Boolean	Specifies that characters should be displayed in right-to-left order. The application must be running on a Hebrew or Arabic version of PowerBuilder under an operating system that supports right-to-left display. Values are: <ul style="list-style-type: none"> ◆ True — Characters display in right-to-left order ◆ False — Characters display in left-to-right order
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. Values are: <p>AnyCase! Lower! Upper!</p>

MultiLineEdit property	Data type	Description
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>PowerScript 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. Values are: <ul style="list-style-type: none"> ◆ True — Text is underlined ◆ False — Text is not underlined
Visible	Boolean	Specifies whether the control is visible. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
VScrollBar	Boolean	Specifies whether a vertical scrollbar is displayed on the right of the control. Values are: <ul style="list-style-type: none"> ◆ True — Vertical scrollbar is displayed ◆ False — Vertical scrollbar is not displayed
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

MultiLineEdit 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).
LoseFocus	When the control loses focus (becomes inactive).
Modified	When a control has been changed and loses focus.
Other	When a Controls message occurs that is not a PowerBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

Functions

MultiLineEdit 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 to the clipboard.

MultiLineEdit function	Data type returned	Description
Drag	Integer	Starts or ends the dragging of the control.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
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.

MultiLineEdit function	Data type returned	Description
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.
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 properties.
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 control

An OLE control contains an object, such as a spreadsheet or word processing document, that was created by an OLE-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).

OLE controls are displayed in the OLE tab of the Browser as Insertable Objects.

Platform note

OLE Objects are only supported on the Windows platform.

FOR INFO For more information about using OLE in an application, see *Application Techniques*.

Properties

OLEControl property	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. For more information about color, see the <i>PowerScript Reference</i> .
Border	Boolean	Specifies whether the control has a border. Values are: <ul style="list-style-type: none"> ◆ True — Control has a border ◆ False — Control does not have a border
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: <ul style="list-style-type: none"> StyleBox! StyleLowered! StyleRaised! StyleShadowBox!

OLEControl property	Data type	Description
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order. Values are: <ul style="list-style-type: none"> ◆ True — Control moved to top ◆ False — Control not moved to top
ClassLongName	String	(Read-only) The long name for the server application associated with the OLE object in the control.
ClassShortName	String	(Read-only) The short name for the server application associated with the OLE object in the control.
ContentsAllowed	omContents Allowed	Specifies whether the OLE object in the control must be embedded or linked or whether either method is allowed when Insert is called at runtime.
DisplayName	String	User-readable name for your OLE control when the control is activated in place. This name is displayed in OLE dialog boxes and windows that show the object's name. If you don't specify a value, the name of the control (such as ole_1) is used for DisplayName.
DisplayType	omDisplayType	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.
DocFileName	String	(Read-only) The name of the file containing the object. If the object has never been saved to a disk file, then the value of this property is " ".

OLEControl property	Data type	Description
DragAuto	Boolean	<p>Specifies whether PowerBuilder puts the control automatically into Drag mode. Values are:</p> <ul style="list-style-type: none"> ◆ 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	<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). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control can be selected ◆ False — Control cannot be selected
FocusRectangle	Boolean	<p>Specifies whether a dotted rectangle (the focus rectangle) will frame the control when it has focus. Values are:</p> <ul style="list-style-type: none"> ◆ True — Control framed when it has focus ◆ False — Control not framed when it has focus
Height	Integer	<p>Specifies the height of the control, in PowerBuilder units.</p>

OLEControl property	Data type	Description
IsDragTarget	Boolean	Specifies whether an OLE object can be dropped on the control. Values are: <ul style="list-style-type: none"> ◆ True — OLE object can be dropped on control ◆ False — OLE object cannot be dropped on control
LinkItem	String	(Read-only) The entire link name of the item to which the object is linked. For example, if the object is linked to C:\FILENAME.XLS!A1:B2, then LinkItem would contain C:\FILENAME.XLS!A1:B2.
LinkUpdate Options	omLinkUpdate Options	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	Used in scripts to apply server commands to the linked or embedded OLE object within the control.
ObjectData	Blob	If the object is embedded, the object itself is stored as a blob in the ObjectData property. If the object is linked, this property contains the link information and the cached image (for display).
ParentStorage	omStorage	(Read-only) Specifies the parent storage.
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.

OLEControl property	Data type	Description
Resizable	Boolean	Specifies whether the control is resizable. Values are: <ul style="list-style-type: none"> ◆ True — Control is resizable ◆ False — Control is not resizable
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. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
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

OLEControl event	Occurs
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.

OLEControl event	Occurs
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.
Error	During OLE automation when an error occurs.
ExternalException	During OLE automation when the OLE server generates an exception during command execution (getting and setting properties, calling functions).
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.
PropertyChanged	When an OLE Server supporting notifications sends this message to the control.
PropertyRequestEdit	When an OLE Server supporting notifications sends this message to the control.
RButtonDown	When the right mouse button is pressed in the control.
Rename	When the server application notifies the control that the 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

OLEControl 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.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetData	Integer	Returns data in a caller-supplied format from an OLE server that supports Uniform Data Transfer.
GetNativePointer	Integer	Returns a pointer to the underlying OLE object.
GetParent	PowerObject	Returns a reference to the name of the parent object.
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.

OLEControl function	Data type returned	Description
Move	Integer	Moves the control to a specified location.
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.
ReleaseNativePointer	Integer	Releases the pointer to the underlying OLE object.
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).
SetData	Integer	Sends data in a caller-supplied format to an OLE server that supports Uniform Data Transfer.
SetFocus	Integer	Sets focus to the control.

OLEControl function	Data type returned	Description
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 properties.
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.
UpdateLinksDialog	Integer	Invokes the OLE dialog to update link information if the file has moved.

OLE custom control (OCX)

The PowerBuilder class `OLECustomControl` is a container for OLE custom controls, also known as OCXs. When you create a PowerBuilder OLE custom control container, the Insert Object dialog prompts you to select the custom control to insert in the container. Your choices are the controls that have been registered in the system registry. If a custom control is not registered by its install process, you can register it in the Insert Object dialog.

The Browser also displays the registered OLE custom controls. Select the OLE tab of the Browser and double-click OLE Custom Controls.

Platform note

OLE Objects are only supported on the Windows platform.

Properties

In OLE terminology, font information and the display name are called **ambient properties**. Ambient properties provide default information that the OCX can use, if it is programmed to recognize and use such information.

PowerBuilder does not display text for the OCX, so it does not use the font and display name properties directly. If the OCX is programmed to recognize ambient properties, it can use the values PowerBuilder provides when the OCX displays text or needs a name to display in a title bar.

OLECustom Control property	Data type	Description
Alignment	Alignment (enumerated)	Specifies the text alignment in the control. Values are: Center! Justify! Left! Right!
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215. For more information about color, see the <i>PowerScript Reference</i> .

OLECustom Control property	Data type	Description
Border	Boolean	Specifies whether the control has a border. Values are: <ul style="list-style-type: none"> ◆ True — Control has a border ◆ False — Control does not have a border
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order (set at runtime only). Values are: <ul style="list-style-type: none"> ◆ True — Control moved to top ◆ False — Control not moved to top
Cancel	Boolean	Specifies whether the control acts as the Cancel button in the window (the Cancel button receives a Clicked event if the user presses esc). Values are: <ul style="list-style-type: none"> ◆ True — Control is the Cancel button ◆ False — Control is not the Cancel button
ClassLongName	String	Specifies the long name for the server application associated with the OLE object in the control (read-only).
ClassShortName	String	Specifies the short name for the server application associated with the OLE object in the control (read-only).

OLECustom Control property	Data type	Description
DisplayName	String	Specifies 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 specify a value, the name of the control (such as ole_1) is used for DisplayName.
Default	Boolean	<p>Specifies whether the button-style OLE control is the default control in the window (the default control has a thick border and receives a Clicked event if the user presses ENTER without selecting a control).</p> <p>This property applies only to OCXs that act like command buttons.</p> <p>Values are:</p> <ul style="list-style-type: none"> ◆ True — Control is the default control ◆ False — Control is not the default control <hr/> <p>Editable controls Default behavior can be affected by editable controls on the window. For more information, see the <i>PowerBuilder User's Guide</i>.</p>
DragAuto	Boolean	<p>Specifies whether PowerBuilder puts the control automatically into Drag mode. Values are:</p> <ul style="list-style-type: none"> ◆ 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.

OLECustom Control property	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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control can be selected ◆ False — Control cannot be selected
FaceName	String	<p>Specifies a typeface name (for example, ARIAL) that you want the OCX to use for text (when the OCX is designed to use this ambient property).</p>
FocusRectangle	Boolean	<p>Specifies whether a dotted rectangle (the focus rectangle) will frame the control when it has focus. Values are:</p> <ul style="list-style-type: none"> ◆ True — Control framed when it has focus ◆ False — Control not framed when it has focus

OLECustom Control property	Data type	Description
FontCharSet	FontCharSet (enumerated)	<p>Specifies the font character set that you want the OCX to use for text (when the OCX is designed to use this ambient property). Values are:</p> <ul style="list-style-type: none"> ANSI! ChineseBig5! DefaultCharSet! HangEul! OEM! ShiftJIS! Symbol!
FontFamily	FontFamily (enumerated)	<p>Specifies the font family (type style) that you want the OCX to use for text (when the OCX is designed to use this ambient property). Values are:</p> <ul style="list-style-type: none"> AnyFont! Decorative! Modern! Roman! Script! Swiss!
FontPitch	FontPitch (enumerated)	<p>Specifies the pitch (character spacing) that you want the OCX to use for text (when the OCX is designed to use this ambient property). Values are:</p> <ul style="list-style-type: none"> Default! Fixed! Variable!
Height	Integer	<p>Specifies the height of the control, in PowerBuilder units.</p>
IsDragTarget	Boolean	<p>Specifies whether data can be dropped on the control. Values are:</p> <ul style="list-style-type: none"> ◆ True — Data can be dropped on control ◆ False — Data cannot be dropped on control

OLECustom Control property	Data type	Description
Italic	Boolean	Specifies that you want the OCX to display text in italic (when the OCX is designed to use this ambient property). Values are: <ul style="list-style-type: none"> ◆ True — Text is italic ◆ False — Text is not italic
Object	omObject	Specifies the link information that connects the control to the server's data.
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.
TextColor	Long	Specifies the color that you want the OCX to use for text (when the OCX is designed to use this ambient property). The color is a numeric value: -2 to 16,777,215. For more information about color, see the <i>PowerScript Reference</i> .
TextSize	Integer	Specifies the point size that you want the OCX to use for displaying text (when the OCX is designed to use this ambient property). For backward compatibility, the size is stored as a negative number; for example, 10-point text size is stored as -10.
Underline	Boolean	Specifies that you want the OCX to underline text (when the OCX is designed to use this ambient property). Values are: <ul style="list-style-type: none"> ◆ True — Text is underlined ◆ False — Text is not underlined

OLECustom Control property	Data type	Description
Visible	Boolean	Specifies whether the control is visible. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
Weight	Integer	Specifies the stroke weight that you want the OCX to use for text (when the OCX is designed to use this ambient property). Sample values are 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

The PowerBuilder Script painter displays PowerBuilder events and events defined by the OCX inserted in the control.

FOR INFO To get information about an event that doesn't appear in this list, check the documentation for the OCX.

OLECustomControl event	Occurs
Clicked	When the control is clicked (selected or unselected).
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).

OLECustomControl event	Occurs
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.
Error	During OLE automation when an error occurs.
ExternalException	During OLE automation when the OLE server generates an exception during command execution (getting and setting properties, calling functions).
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.
PropertyChanged	When an OLE Server supporting notifications sends a message that a property value has been changed.
PropertyRequestEdit	When an OLE Server supporting notifications sends a message that a property value is about to be changed.
RButtonDown	When the right mouse button is pressed in the control.

Functions

OLECustomControl function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Puts the object into drag mode.
GetData	Integer	Returns data in a format you specify from an OLE server that supports Uniform Data Transfer.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.

OLECustomControl function	Data type returned	Description
GetNativePointer	Integer	Returns a pointer to the underlying OLE object.
GetParent	Power Object	Returns a reference to the name of the parent object.
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.
ReleaseNativePointer	Integer	Releases pointer to underlying OLE object.
Resize	Integer	Changes the size of the control.
SetAutomationLocale	Integer	Sets the language for automation programming.
SetData	Integer	Sends data in a caller-supplied format to an OLE server that supports Uniform Data Transfer.
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 properties
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.

OLEObject

The OLEObject object acts as a proxy for a remote OLE object. The OLEObject object has no properties.

You can customize your own version of the OLEObject object by defining a standard class user object inherited from the built-in OLEObject. You can then access the OLEObject events by writing scripts that contain code for the events.

OLEObject objects are displayed in the OLE tab of the Browser as Programmable Objects.

Platform note

OLE Objects are only supported on the Windows platform.

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

FOR INFO For more information about using the OLEObject object in an application, see *Application Techniques*.

OLEObject is a dynamic object

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

Events

OLEObject event	Occurs
Constructor	When the user object is created.
Destructor	When the user object is destroyed.
Error	When an error is found in a data or property expression for an external object.
ExternalException	When the evaluation of an expression involving properties of an external object causes an error. This type of event occurs before the Error event.

Functions

OLEObject function	Data type returned	Description
ClassName	String	Returns the name assigned to the user object.
ConnectToNewObject	Integer	Creates a new instance of the class and connects to it.
ConnectToObject	Integer	Opens a specified file and connects to the corresponding server application.
DisconnectObject	Integer	Releases all objects previously connected.
GetAutomationNativePointer	Integer	Returns a pointer to the underlying OLE object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
PostEvent	Boolean	Adds an event to the end of the message queue of the user object.
ReleaseAutomationNativePointer	Integer	Releases the pointer to the underlying OLE object.
SetAutomationLocale	Integer	Sets the language locale to be used for automation programming on the 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 customize your own version of the OLEStorage object by defining a class user object inherited from the built-in OLEStorage object.

Platform note

OLE Objects are only supported on the Windows platform.

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

FOR INFO For more information about using the OLEStorage object in an application, see *Application Techniques*.

Properties

OLEStorage property	Data type	Description
DocumentName	String	Specifies the name of the storage currently open. For a root level storage, this is the name of the file containing the storage. For substorages, this is the member name of the substorage.

Events

OLEStorage event	Occurs
Constructor	When the user object is created.
Destructor	When the user object is destroyed.

Functions

OLEStorage 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.)
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
MemberDelete	Integer	Deletes the specified member.
MemberExists	Integer	Specifies whether the specified member exists.
MemberRename	Integer	Renames the 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 customize your own version of the OLEStream object by defining a class user object inherited from the built-in OLEStream object.

Platform note

OLE Objects are only supported on the Windows platform.

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

FOR INFO For more information about using the OLEStream object in an application, see *Application Techniques*.

Properties

OLEStream property	Data type	Description
Name	String	Specifies the member name of the stream within its parent storage.
Storage	OMStorage	(Read-only) Specifies the storage containing the stream. Streams are always opened from within an OLEStorage object.

Events

OLEStream event	Occurs
Constructor	When the user object is created.
Destructor	When the user object is destroyed.

Functions

OLEStream function	Data type returned	Description
ClassName	String	Returns the name assigned to the user object.
Close	Integer	Releases any stream previously opened.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
Length	Integer	Obtains the length of the open stream.
Open	Integer	Opens the 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.

Oval control

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.

An oval cannot be selected and therefore has no events.

Properties

Oval property	Data type	Description
FillColor	Long	Specifies the numeric value of the color used to fill the control: -2 to 16,777,215. For information on color, see the <i>PowerScript Reference</i> .
FillPattern	FillPattern (enumerated)	Specifies the hatch pattern used to fill the control. Values are: BDiagonal! Diamond! FDiagonal! Horizontal! Solid! Square! Vertical! 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 information on color, see the <i>PowerScript Reference</i> .

Oval property	Data type	Description
LineStyle	LineStyle (enumerated)	Specifies the style of the line used to draw the control. Values are: Continuous! Dash! DashDot! DashDotDot! Dot! Transparent!
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. Values are: ◆ True — Control is visible ◆ False — Control is not visible
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

Oval function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.

Oval function	Data type returned	Description
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 control

A picture is a bitmap image (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 BMP, RLE, or WMF format.

Platform note

WMF files are only supported on the Windows platform.

Properties

Picture property	Data type	Description
Border	Boolean	Specifies whether the control has a border. Values are: <ul style="list-style-type: none">◆ True — Control has a border◆ False — Control does not have a border
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order of the window. Values are: <ul style="list-style-type: none">◆ True — Control moved to top◆ False — Control not moved to top

Picture property	Data type	Description
DragAuto	Boolean	<p>Specifies whether PowerBuilder puts the control automatically into Drag mode. Values are:</p> <ul style="list-style-type: none"> ◆ 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	<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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control can be selected ◆ False — Control cannot be selected
FocusRectangle	Boolean	<p>Specifies whether a dotted rectangle (focus rectangle) frames the picture when it has focus. Values are:</p> <ul style="list-style-type: none"> ◆ True — Control will be framed when it has focus ◆ False — Control will not be framed when it has focus
Height	Integer	<p>Specifies the height of the control, in PowerBuilder units.</p>

Picture property	Data type	Description
Invert	Boolean	Specifies whether the control displays with its colors inverted. Values are: <ul style="list-style-type: none"> ◆ True — Colors are inverted ◆ False — Colors are not inverted
OriginalSize	Boolean	Specifies whether the width and height properties of a bitmap image (picture) will be set to their original values. Values are: <ul style="list-style-type: none"> ◆ True — Width and height set to original values ◆ False — Existing width and height not changed <p>In the Window painter, setting OriginalSize to True overrides the existing width and height. You cannot change this property in a script.</p>
PictureName	String	Specifies the name of the file that contains the picture. The file extension BMP, RLE, or WMF 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 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. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
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.

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

Events

Picture 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.
DoubleClick	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

Picture function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.

Picture function	Data type returned	Description
Draw	Integer	Draws a picture in the parent window at a specified location.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
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.
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 properties.
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 control

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.

Properties

PictureButton property	Data type	Description
<code>HTextAlign</code>	Alignment (enumerated)	Specifies how the text in the control is aligned. Values are: Center! Justify! Left! Right!
<code>BringToTop</code>	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order of the window.
<code>Cancel</code>	Boolean	Specifies whether the control acts as the Cancel button (the Cancel button receives a <code>Clicked</code> event if the user presses <code>ESC</code>). Values are: <ul style="list-style-type: none">◆ True — Control acts as Cancel button◆ False — Control does not act as Cancel button

PictureButton property	Data type	Description
Default	Boolean	<p>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). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control is default PictureButton ◆ False — Control is not default PictureButton <hr/> <p>Editable controls Default behavior can be affected by editable controls on the window. For more information, see the <i>PowerBuilder User's Guide</i>.</p>
DisabledName	String	<p>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.</p> <hr/> <p>Platform note WMF files are only supported on the Windows platform.</p>
DragAuto	Boolean	<p>Specifies whether PowerBuilder puts the control automatically into Drag mode. Values are:</p> <ul style="list-style-type: none"> ◆ 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.

PictureBox property	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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control is enabled ◆ False — Control is not enabled
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. Values are:</p> <p>ANSI! ChineseBig5! DefaultCharSet! Hangeul! OEM! ShiftJIS! Symbol!</p>

PictureButton property	Data type	Description
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) used for the text in the control. Values are: AnyFont! Decorative! Modern! Roman! Script! Swiss!
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control. Values are: Default! Fixed! Variable!
Height	Integer	Specifies the height of the control, in PowerBuilder units.
Italic	Boolean	Specifies whether the text in the control is italic. Values are: ◆ True — Text is italic ◆ False — Text is not italic
OriginalSize	Boolean	Specifies whether the width and height properties of a bitmap image (picture) are set to their original values. Values are: ◆ True — Width and height are set to original values ◆ False — Existing width and height are not changed to original values In the Window painter, setting OriginalSize to True overrides the existing width and height. You cannot change this property 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.

PictureButton property	Data type	Description
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. Values are: <ul style="list-style-type: none"> ◆ True — Text is underlined ◆ False — Text is not underlined
Visible	Boolean	Specifies whether the control is visible. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
VTextAlign	VTextAlign (enumerated)	Specifies how the text in the control is aligned. Values are: <p>Bottom! MultiLine! Top! VCenter!</p> <p>All these values except MultiLine! assume there is only one line of text.</p>
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.

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

Events

PictureButton event	Occurs
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

PictureButton function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.

PictureButton function	Data type returned	Description
GetParent	PowerObject	Returns a reference to the name of the parent object.
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 properties.
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.

PictureListBox control

A PictureListBox displays available options or values, which can include pictures. If more options or values exist than can display in the PictureListBox at one time or the text exceeds the width of the PictureListBox, the PictureListBox will have one or two (vertical or horizontal) scrollbars.

The pictures used in a PictureListBox control can be bitmaps (BMP files), icons (ICO files), or cursors (CUR files).

Platform note

ICO files are only supported on the Windows and Macintosh platforms.

Properties

PictureListBox property	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. For more information about color, see the <i>PowerScript Reference</i> .
Border	Boolean	Specifies whether the control has a border. Values are: <ul style="list-style-type: none"> ◆ True — Control has a border ◆ False — Control does not have a border
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!

PictureListBox property	Data type	Description
BringToTop	Boolean	<p>Specifies whether PowerBuilder moves the control to the top of the front-to-back order. Values are:</p> <ul style="list-style-type: none">◆ True — Control moved to top◆ False — Control not moved to top
DisableNoScroll	Boolean	<p>Specifies behavior of scrollbar. Values are:</p> <ul style="list-style-type: none">◆ True — The scrollbar will always be visible, but will be disabled when all the items can be accessed without it◆ False — The scrollbar will be displayed only if it is necessary (based on the number of items and the height of the listbox)
DragAuto	Boolean	<p>Specifies whether PowerBuilder puts the control automatically into Drag mode. Values are:</p> <ul style="list-style-type: none">◆ 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.

PictureListBox property	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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control can be selected ◆ False — Control cannot be selected
ExtendedSelect	Boolean	<p>Specifies whether users can select multiple items in the listbox at one time. Values are:</p> <ul style="list-style-type: none"> ◆ 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 CTRL+ CLICK on multiple items. ◆ False — Users cannot select multiple items. <hr/> <p>Used with MultiSelect The MultiSelect property 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>

PictureListBox property	Data type	Description
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. Values are: ANSI! ChineseBig5! DefaultCharSet! Hangeul! OEM! ShiftJIS! Symbol!
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) used for the text in the control. Values are: AnyFont! Decorative! Modern! Roman! Script! Swiss!
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control. Values are: Default! Fixed! Variable!
Height	Integer	Specifies the height of the control, in PowerBuilder units.
HScrollBar	Boolean	Specifies whether a horizontal scrollbar displays. Values are: ◆ True — Horizontal scrollbar displays ◆ False — Horizontal scrollbar does not display
Italic	Boolean	Specifies whether the text in the control is italic. Values are: ◆ True — Text is italic ◆ False — Text is not italic

PictureListBox property	Data type	Description
Item[]	String	Specifies the items in the control. Not updated after initialization.
ItemPictureIndex[]	Integer	Specifies the picture index for each item in the Item property array. Not updated after initialization.
MultiSelect	Boolean	Specifies whether users can select multiple items in the PictureListBox at one time. Values are: <ul style="list-style-type: none"> ◆ True — Users can select multiple items ◆ False — Users cannot select multiple items
PictureHeight	Integer	Specifies height of the picture, in pixels. In a script, this property may only be set when there are no images in the image list. If the value is 0 at the time the first image is added, the size of that image is used to set the size of the rest of the images added.
PictureWidth	Integer	Specifies width of the picture, in pixels. In a script, this property may only be set when there are no images in the image list. If the value is 0 at the time the first image is added, the size of that image is used to set the size of the rest of the images added.
PictureMaskColor	Long	Specifies the numeric value of the color to be used to mask user defined bitmaps added through the initial picture array or with the AddPicture function. System-defined bitmaps know their mask color and this color will be ignored. This value is used when a picture is added, and therefore may be changed between AddPicture calls. Values can be: -2 to 16,777,215. For more information about color, see the <i>PowerScript Reference</i> .

PictureListBox property	Data type	Description
PictureName[]	String	<p>Specifies the names of the files containing the pictures added during initialization. The file extension BMP, ICO, or CUR is required.</p> <p>Not updated after initialization.</p> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
RightToLeft	Boolean	<p>Specifies that characters should be displayed in right-to-left order. The application must be running on a Hebrew or Arabic version of PowerBuilder under an operating system that supports right-to-left display. Values are:</p> <ul style="list-style-type: none"> ◆ True — Characters display in right-to-left order ◆ False — Characters display in left-to-right order
Sorted	Boolean	<p>Specifies whether the items in the PictureListBox are sorted. Values are:</p> <ul style="list-style-type: none"> ◆ True — Items are sorted ◆ False — Items are not sorted
TabOrder	Integer	Specifies the tab value of the control (0 means the user cannot tab to the control).

PictureListBox property	Data type	Description
TabStop[]	Integer array	Specifies the positions of the tab stops in the PictureListBox. 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. For more information about color, see the <i>PowerScript 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. Values are: <ul style="list-style-type: none"> ◆ True — Text is underlined ◆ False — Text is not underlined
Visible	Boolean	Specifies whether the control is visible. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible

PictureListBox property	Data type	Description
VScrollBar	Boolean	Specifies whether a vertical scrollbar is displayed on the right of the PictureListBox. Values are: <ul style="list-style-type: none"> ◆ True — Vertical scrollbar is displayed ◆ False — Vertical scrollbar is not displayed
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

PictureListBox 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).

PictureListBox event	Occurs
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Controls 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

PictureListBox function	Data type returned	Description
AddItem	Integer	Adds a new item to the end of the PictureListBox. If the Sorted property of the control is True, the items are sorted again after the item is added. The AddItem function does not update the Item[] or ItemPicture[] properties of this control.
AddPicture	Integer	Adds the bitmap, icon, or cursor file to the main image list. Does not update PictureName[].
ClassName	String	Returns the name assigned to the control.
DeleteItem	Integer	Deletes the item indicated by the index from the PictureListBox.
DeletePicture	Integer	Deletes the specified picture from the image list. Does not update PictureName [].
DeletePictures	Integer	Deletes all the pictures from the image list. Does not update PictureName[].
DirList	Boolean	Populates the PictureListBox 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.

PictureBox function	Data type returned	Description
FindItem	Integer	Finds the first item in the PictureBox (after the specified index) that begins with the specified string.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
Hide	Integer	Makes the control invisible.
InsertItem	Integer	Adds a new item to the PictureBox before the item indicated by the index. If the Sorted property 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.
SelectedIndex	Integer	Returns the index of the item in the PictureBox 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.

PictureListBox function	Data type returned	Description
SelectItem	Integer	<p>Finds and highlights an item in the control. Use Syntax 1 when you know the text of the item but not its position. Use Syntax 2 when you know the position of the item in the control's list or you want to clear the current selection.</p> <p>SelectItem has no effect on a PictureListBox whose MultiSelect property is True. Instead, use SetState to select items without affecting the selected state of other items in the list.</p>
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 properties.
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 property 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.

PictureBox function	Data type returned	Description
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.

Pipeline object

A Pipeline system object is used to manage a data pipeline during execution. You use a Pipeline object by defining a standard class user object inherited from the built-in Pipeline object in the User Object painter. You can then access the Pipeline events by writing scripts that contain code for the events.

FOR INFO For more information about piping data, see *Application Techniques*.

Properties

Pipeline property	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

Pipeline 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

Pipeline function	Data type returned	Description
Cancel	Integer	Stops execution of a pipeline.
ClassName	String	Returns the name assigned to the user object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	Power Object	Returns a reference to the name of the parent 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.

ProfileCall object

The ProfileCall object provides information about the calls in the performance analysis model, including information about the called routine and the calling routine, the number of times the call was made and the elapsed time (in seconds). You use the ProfileCall object in conjunction with the ProfileRoutine and Profiling objects.

The ProfileCall object has no events.

Properties

ProfileCall property	Data type	Description
AbsoluteSelfTime	Decimal	The time (in seconds) spent in the called routine.
AbsoluteTotalTime	Decimal	The time (in seconds) spent in the called routine and in subsequent called routines.
CalledRoutine	ProfileRoutine	An object of data type ProfileRoutine containing the destination of the call.
CallingLine	ProfileLine	An object of data type ProfileLine containing the initiating line of the call. If the call object represents an aggregation of multiple calls from a routine, an invalid object is returned.
CallingRoutine	ProfileRoutine	An object of data type ProfileRoutine containing the routine that initiated the call.
HitCount	Long	The number of times the calling routine called the called routine.
PercentCalleeSelfTime	Double	AbsoluteSelfTime as a percentage of the total time (in seconds) the calling routine was active.
PercentCalleeTotalTime	Double	AbsoluteTotalTime as a percentage of the total time (in seconds) the calling routine was active.

ProfileCall property	Data type	Description
PercentCallerTotalTime	Double	The total time (in seconds) spent in the calling routing as a percentage of the total time the calling routine was active.

Functions

ProfileCall function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	Power Object	Returns a reference to the name of the parent object.
TypeOf	Object (enumerated)	Returns the type of the object.

ProfileClass object

The ProfileClass object provides information about the classes in the performance analysis model, including the routines that exist within a class. You use the ProfileClass object in conjunction with the Profiling object.

The ProfileClass object has no events.

Properties

ProfileClass property	Data type	Description
LibraryName	String	The name of the library that contains the class. The value is " " for system classes and ESQL statements.
Name	String	The name of the class or Embedded SQL to represent all embedded SQL activities. Nested classes (like controls on a window) have a name of the form <i>class name`embedded class name</i> .

Functions

ProfileClass function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	Power Object	Returns a reference to the name of the parent object.
RoutineList	ErrorReturn (enumerated)	Provides a list of the routines (defined as ProfileRoutine objects) that exist in the model within a class.
TypeOf	Object (enumerated)	Returns the type of the object.

ProfileLine object

The ProfileLine object provides information about the lines in each routine in the performance analysis model, including the number of times the line was hit, any calls made from the line and the time (in seconds) spent on the line and in any called functions. You use the ProfileLine object in conjunction with the ProfileRoutine and Profiling objects.

The ProfileLine object has no events.

Properties

ProfileLine property	Data type	Description
AbsoluteSelfTime	Decimal	The time (in seconds) spent on this line itself. If the line executed more than once, this is the total time spent on the line.
AbsoluteTotalTime	Decimal	The time (in seconds) spent on this line and on lines called from this line. If the line executed more than once, this is the total time spent on the line and on called lines.
HitCount	Long	The number of times the line was called.
LineNumber	Long	The line number. Line 0 is a special line that represents the time (in seconds) taken to initialize the local variables (including calling constructors for autoinstantiated objects).
MaxSelfTime	Decimal	The longest time(in seconds) spent just on this line. If the line executed only once, this is the same as AbsoluteSelfTime.
MaxTotalTime	Decimal	The longest time (in seconds) spent on this line and on called lines. If the line executed only once, this is the same as AbsoluteTotalTime.

ProfileLine property	Data type	Description
MinSelfTime	Decimal	The shortest time (in seconds) spent just on this line. If the line executed only once, this is the same as AbsoluteSelfTime.
MinTotalTime	Decimal	The shortest time (in seconds) spent on this line and on called lines. If the line executed only once, this is the same as AbsoluteTotalTime.
PercentSelfTime	Double	AbsoluteSelfTime as a percentage of the total time (in seconds) tracing was active.
PercentTotalTime	Double	AbsoluteTotalTime as a percentage of the total time (in seconds) tracing was active.
Routine	ProfileRoutine	The routine that the line is in.

Functions

ProfileLine function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	Power Object	Returns a reference to the name of the parent object.
OutGoingCall-List	ErrorReturn (enumerated)	Provides a list of the calls (defined as ProfileCall objects) to other routines from a line.
TypeOf	Object (enumerated)	Returns the type of the object.

ProfileRoutine object

The ProfileRoutine object provides information about the routines in the performance analysis model. It includes the time (in seconds) spent in the routine, any called routines, the number of times each routine was called, and the class to which the routine belongs. You use the ProfileRoutine object in conjunction with the Profiling and ProfileCall or ProfileLine objects.

The ProfileRoutine object has no events.

Properties

ProfileRoutine property	Data type	Description
AbsoluteSelfTime	Decimal	The time (in seconds) spent in this routine. If the routine executed more than once, this is the total time spent in the routine.
AbsoluteTotalTime	Decimal	The time (in seconds) spent in this routine and in routines called from this routine. If the routine executed more than once, this is the total time spent in the routine and in called routines.
Class	ProfileClass	The class the routine is in. For ESQL activities, the value is Embedded SQL. For global and system functions, the value is an invalid object.
HitCount	Long	The number of times this routine was called or the number of objects created/destroyed.

ProfileRoutine property	Data type	Description
Kind	ProfileRoutine Kind (enumerated)	The kind of routine node. Values are: <ul style="list-style-type: none"> ◆ RoutineESQL!—Represents an embedded SQL statement ◆ RoutineEvent!—Represents an event ◆ RoutineFunction!—Represents a function ◆ RoutineGarbageCollection!—Represents a garbage collection phase ◆ RoutineObjectCreation!—Represents object creation ◆ RoutineObjectDestruction!—Represents object destruction ◆ RoutineRoot!—Represents the windowing system
MaxSelfTime	Decimal	The longest time (in seconds) spent in the routine itself. If the routine executed only once, this is the same as AbsoluteSelfTime.
MaxTotalTime	Decimal	The longest time (in seconds) spent in the routine and in called routines. If the routine executed only once, this is the same as AbsoluteTotalTime.
MinSelfTime	Decimal	The shortest time (in seconds) spent in the routine itself. If the routine executed only once, this is the same as AbsoluteSelfTime.
MinTotalTime	Decimal	The shortest time (in seconds) spent in the routine and in called routines. If the routine executed only once, this is the same as AbsoluteTotalTime.

ProfileRoutine property	Data type	Description
Name	String	The name of the routine including the argument data types and return value. For ESQL activities, the value is the name of the statement (for example, SELECT). For object creation/destruction, the value is Object Create or Object Destroy.
PercentSelfTime	Double	AbsoluteSelfTime as a percentage of the total time (in seconds) tracing was active.
PercentTotalTime	Double	AbsoluteTotalTime as a percentage of the total time (in seconds) tracing was active.

Functions

ProfileRoutine function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	Power Object	Returns a reference to the name of the parent object.
IncomingCallList	ErrorReturn (enumerated)	Provides a list of the callers (defined as ProfileCall objects) of this routine.
LineList	ErrorReturn (enumerated)	Provides a list, in line order, of the lines (defined as ProfileLine objects) in the routine.
OutgoingCallList	ErrorReturn (enumerated)	Provides a list of the calls (defined as ProfileCall objects) to other routines from within this routine.
TypeOf	Object (enumerated)	Returns the type of the object.

Profiling object

The Profiling object is used to analyze the performance of a PowerBuilder application. It provides a performance analysis model listing all the routines (both functions and events) logged in a given trace file. It includes the functions you call to name the trace file to be analyzed, build the model, and list the classes and routines included in the model. You use the Profiling object in conjunction with the ProfileCall, ProfileClass, ProfileLine, and ProfileRoutine objects.

The Profiling object has no events.

Properties

Profiling property	Data type	Description
ApplicationName	String	The name of the application used to generate the trace file.
CollectionTime	Decimal	The amount of time (in seconds) taken by the collection of trace data. This time has already been accounted for in the time stamps from the trace file (ie, the proper amount of time has been subtracted from the time stamps before they are put in the trace file). If no model has been created, NULL is returned.
NumberOfActivities	Long	The total number of activities that exist in the trace file. The value is 0 if this property is called before the trace file name is set.
TraceFileName	String	The name of the trace file to use to build the model. The value is an empty string if the name has not been successfully set.

Functions

Profiling function	Data type returned	Description
BuildModel	ErrorReturn (enumerated)	Builds a performance analysis model based on the previously specified trace file.
ClassList	ErrorReturn (enumerated)	Provides a list of the classes (defined as ProfileClass objects) included in the model.
ClassName	String	Returns the name assigned to the object.
DestroyModel	ErrorReturn (enumerated)	Destroys the current performance analysis model.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	Power Object	Returns a reference to the name of the parent object.
RoutineList	ErrorReturn (enumerated)	Provides a list of the routines (defined as ProfileRoutine objects) included in the model.
SetTraceFileName	ErrorReturn	Indicates the name of the trace file to use for analysis and validates the header format.
SystemRoutine	ProfileRoutine	Provides the routine node (defined as a ProfileRoutine object) representing the system root.
TypeOf	Object (enumerated)	Returns the type of the object.

RadioButton control

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.

In a GroupBox

RadioButtons are often grouped in a GroupBox. In this case, the user can select only one button in the group, and the group usually has a default button.

Properties

RadioButton property	Data type	Description
Automatic	Boolean	Specifies whether the control becomes dark when it is clicked. Values are: <ul style="list-style-type: none"> ◆ True — Control becomes dark when clicked ◆ False — Control does not become dark when clicked
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215. For more information about color, see the <i>PowerScript Reference</i> .
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order in the window. Values are: <ul style="list-style-type: none"> ◆ True — Control moved to top ◆ False — Control not moved to top

RadioButton property	Data type	Description
Checked	Boolean	<p>Specifies whether the item is selected (the center is dark). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control is selected ◆ False — Control is not selected
DragAuto	Boolean	<p>Specifies whether PowerBuilder puts the control automatically into Drag mode. Values are:</p> <ul style="list-style-type: none"> ◆ 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	<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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control is enabled ◆ False — Control is not enabled
FaceName	String	<p>Specifies the name of the typeface in which the text of the control displays (for example, HELV or COURIER).</p>

RadioButton property	Data type	Description
FontCharSet	FontCharSet (enumerated)	Specifies the font character set used for the text in the control. Values are: ANSI ChineseBig5! DefaultCharSet! Hangeul! OEM! ShiftJIS! Symbol!
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) used for the text in the control. Values are: AnyFont! Decorative! Modern! Roman! Script! Swiss!
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control. Values are: Default! Fixed! Variable!
Height	Integer	Specifies the height of the control, in PowerBuilder units.
Italic	Boolean	Specifies whether the text in the control is italic. Values are: ◆ True — Text is italic ◆ False — Text is not italic
LeftText	Boolean	Specifies whether the text displays to the left of the control. Values are: ◆ True — Text displays to the left ◆ False — Text does not display to the left
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.

RadioButton property	Data type	Description
RightToLeft	Boolean	<p>Specifies that characters should be displayed in right-to-left order. The application must be running on a Hebrew or Arabic version of PowerBuilder under an operating system that supports right-to-left display. Values are:</p> <ul style="list-style-type: none"> ◆ True — Characters display in right-to-left order ◆ False — Characters display in left-to-right order
TabOrder	Integer	<p>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.</p> <p>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).</p>
Tag	String	Specifies the tag value assigned to the control.
Text	String	Specifies the text that displays next to the control.
TextColor	Long	<p>Specifies the numeric value of the color used for text: -2 to 16,777,215.</p> <p>For more information about color, see the <i>PowerScript Reference</i>.</p>
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	<p>Specifies whether the text in the control is underlined. Values are:</p> <ul style="list-style-type: none"> ◆ True — Text is underlined ◆ False — Text is not underlined

RadioButton property	Data type	Description
Visible	Boolean	Specifies whether the control is visible. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
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

RadioButton 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.

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

Functions

RadioButton function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
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 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 properties.
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 control

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.

A rectangle cannot be selected and therefore has no events.

Properties

Rectangle property	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>PowerScript Reference</i> .
FillPattern	FillPattern (enumerated)	Specifies the hatch pattern used to fill the control. Values are: BDiagonal! Diamond! FDiagonal! Horizontal! Solid! Square! Vertical! 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 <code>PowerBuilder</code> units.
LineColor	Long	Specifies the numeric value of the line color: -2 to 16,777,215. For more information about color, see the <i>PowerScript Reference</i> .

Rectangle property	Data type	Description
LineStyle	LineStyle (enumerated)	Specifies the pattern of the line used to draw the control. Values are: Continuous! Dash! DashDot! DashDotDot! Dot! Transparent!
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. Values are: ◆ True — Control is visible ◆ False — Control is not visible
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

Rectangle function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.

Rectangle function	Data type returned	Description
GetParent	PowerObject	Returns a reference to the name of the parent object.
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.

RichTextEdit control

A RichTextEdit control contains a document, which it displays as formatted text. It can include input fields that are linked to a DataWindow control. When a DataWindow's data is shared with the RichTextEdit control, there is one instance of the document in the control that may be displayed multiple times with different occurrences of row data. Input fields whose names match columns in the DataWindow are filled with data from the current row.

Properties

RichTextEdit property	Data type	Description
Accelerator	Integer	Specifies the ASCII value of the accelerator key you want to assign as the accelerator for the control.
BackColor	Long	Specifies the numeric value of the background color of the text editing area of the RichTextEdit. Values are -2 to 16,777,215. For more information about color, see RGB.
Border	Boolean	Specifies whether the control has a border. Values are: <ul style="list-style-type: none"> ◆ True — Has a border ◆ False — Does not have a border
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BottomMargin	Long	Specifies the width of the bottom margin on the printed page.

RichTextEdit property	Data type	Description
BringToTop	Boolean	Specifies whether PowerBuilder will move the control to the top of the front-to-back order in the window. Values are: <ul style="list-style-type: none">◆ True — Will move to the top◆ False — Will not move to the top
DisplayOnly	Boolean	Specifies whether users can make changes to the contents. Values are: <ul style="list-style-type: none">◆ True — The content, including text and input files, is protected (the user cannot edit it)◆ False — The user can edit the content
DocumentName	String	Specifies the name that will display in the print queue when the user prints the contents of the control.
DragAuto	Boolean	Specifies whether PowerBuilder will put the control automatically into Drag mode. Values are: <ul style="list-style-type: none">◆ 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.

RichTextEdit property	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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected). Values are:</p> <ul style="list-style-type: none"> ◆ True — Can be selected ◆ False — Cannot be selected
HeaderFooter	Boolean	<p>Specifies whether the RichTextEdit control has a header/footer section. This property must be set in the painter and cannot be changed during execution. Values are:</p> <ul style="list-style-type: none"> ◆ True — The control has a header/footer section ◆ False — The control does not have a header/footer section <p>If a document has a header or footer and is opened in a control that does not support a header/footer section, then header/footer information in the document is ignored. If the document is then saved in the same file, the header/footer information is lost.</p>
Height	Integer	Specifies the height of the control, in PowerBuilder units.

RichTextEdit property	Data type	Description
HScrollBar	Boolean	Specifies whether the RichTextEdit control has a horizontal scrollbar. Values are: <ul style="list-style-type: none"> ◆ True — A scrollbar displays ◆ False — A scrollbar does not display
InputFieldBackColor	Long	Specifies default background color for all input fields: -2 to 16,777,215.
InputFieldNames Visible	Boolean	Specifies whether input field names are displayed in input fields, rather than the input field values. Values are: <ul style="list-style-type: none"> ◆ True — Input fields display their names ◆ False — Input fields display their data
InputFieldsVisible	Boolean	This property is not currently available. Specifies whether input fields are visible. Values are: <ul style="list-style-type: none"> ◆ True — Input fields are visible ◆ False — Input fields are hidden
LeftMargin	Long	Specifies the width of the left margin on the printed page.
Modified	Boolean	Specifies whether the document has been modified since it was opened or last saved. Modified is the control's "dirty" flag, indicating that the document is in an unsaved state. Values are: <ul style="list-style-type: none"> ◆ True — The document has been modified ◆ False — The document has not been modified <p>When the first change is made to a newly opened or saved document, PowerBuilder sets the Modified attribute and triggers the Modified event.</p>

RichTextEdit property	Data type	Description
PicturesAsFrame	Boolean	When the value is True, graphics (bitmaps) are displayed as frames.
Pointer	String	Specifies the name of the stock pointer of the file containing the pointer that is used for the control.
PopupMenu	Boolean	Specifies whether the user has access to popup menu by clicking the right mouse button on the control. The menu allows the user to cut and paste, insert a file, and select formatting options. Values are: <ul style="list-style-type: none"> ◆ True — Popup menu is enabled ◆ False — Popup menu is disabled
Resizable	Boolean	Specifies whether the user can resize the control. Values are: <ul style="list-style-type: none"> ◆ True — Control is resizable ◆ False — Control is not resizable
ReturnsVisible	Boolean	Specifies whether carriage returns are visible. Values are: <ul style="list-style-type: none"> ◆ True — Carriage returns are visible ◆ False — Carriage returns are hidden
RightMargin	Long	Specifies the width of the right margin on the printed page.
RulerBar	Boolean	Specifies whether a ruler bar is visible above the editing area. If visible, the user can use it to see measurements while setting tabs and margins on the tab bar (see the TabBar property). Values are: <ul style="list-style-type: none"> ◆ True — Ruler bar is visible ◆ False — Ruler bar is hidden <p>If the RichTextEdit popup menu is enabled, the user can use it to turn ruler bar display on and off (see the PopMenu property).</p>

RichTextEdit property	Data type	Description
SpacesVisible	Boolean	Specifies whether spaces are visible. Values are: <ul style="list-style-type: none"> ◆ True — Spaces are visible ◆ False — Spaces are hidden
TabBar	Boolean	Specifies whether a bar for setting tabs is visible above the editing area. Values are: <ul style="list-style-type: none"> ◆ True — Tab bar is visible ◆ False — Tab bar is hidden <p>If the popup menu is enabled, the user can use it to turn tab bar display on and off (see the PopMenu property).</p>
TabOrder	Integer	Specifies the tab value of the control within the window (0 means the user cannot tab to the control). Note that because a RichTextEdit allows tabs within its content, the user cannot tab away from the control.
TabsVisible	Boolean	Specifies whether tabs are visible. Values are: <ul style="list-style-type: none"> ◆ True — Spaces are visible ◆ False — Spaces are hidden
Tag	String	Specifies the tag value assigned to the control.
ToolBar	Boolean	Specifies whether a tool bar for formatting text is visible above the editing area. Values are: <ul style="list-style-type: none"> ◆ True — Tool bar is visible ◆ False — Tool bar is not visible <p>If the popup menu is enabled, the user can use it to turn tool bar display on and off (see the PopMenu property).</p>
TopMargin	Long	Specifies the width of the top margin on the printed page.

RichTextEdit property	Data type	Description
UndoDepth	Integer	Sets the maximum number of activities that Undo will undo.
Visible	Boolean	Specifies whether the control is visible: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
VScrollBar	Boolean	Specifies whether the RichTextEdit control has a vertical scrollbar. Values are: <ul style="list-style-type: none"> ◆ True — A scrollbar displays ◆ False — A scrollbar does not display
Width	Integer	Specifies the width of the control, in PowerBuilder units.
WordWrap	Boolean	Specifies whether text wraps automatically to the next line when the line reaches the margin. Values are: <ul style="list-style-type: none"> ◆ True — Automatic wordwrap is enabled ◆ False — Automatic wordwrap is disabled
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

RichTextEdit event	Occurs
Constructor	Immediately before the Open event occurs in the window.

RichTextEdit event	Occurs
Destructor	Immediately after the Close event occurs in the window.
DoubleClicked	When the user double-clicks anywhere in the RichTextEdit control.
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.
FileExists	When the document in the RichTextEdit is saved to a file, but a file of the specified name already exists.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
InputFieldSelected	When the user selects an input field by double-clicking it or pressing ENTER after clicking on it.
Key	When the user presses a key.
LoseFocus	When the control loses focus (becomes inactive).
Modified	When the first change is made to the contents of the RichTextEdit control, and it has not been saved. In the Modified event script, you can set a flag indicating that the document needs to be saved.
MouseDown	When the left or middle mouse button is pressed on the control.
MouseMove	When the mouse has moved within the control.
MouseUp	When the left or middle mouse button is released on the control.
Other	When a Windows message occurs that is not a PowerBuilder event.
PictureSelected	When the user selects bitmap by double-clicking it or pressing ENTER after clicking on it.
PrintFooter	When the contents of the control is being printed and a footer is about to be printed for a page. For information about page numbers, see the note for PrintHeader.

RichTextEdit event	Occurs
PrintHeader	<p>When the contents of the control is being printed and a header is about to be printed for a new page.</p> <hr/> <p>Page numbers You can control the printed page number by including an input field in the header or footer whose value is set in the PrintHeader or PrintFooter script. You can decide whether the print job is continuously numbered or whether each document instance restarts the page count.</p>
RButtonDown	When the right mouse button is pressed on the control. If the popup menu is enabled, this event does not occur.
RButtonUp	When the right mouse button is released on the control.

Functions

RichTextEdit function	Data type returned	Description
CanUndo	Boolean	Returns True if there is an editing function that can be undone.
ClassName	String	Returns the name assigned to the control.
Clear	Long	Clears the selected text (if any) from the control (but does not place it in the clipboard).
Copy	Long	Copies (but does not delete) the selected contents of the RichTextEdit control (if any) to the clipboard.
CopyRTF	String	Copies the selected contents of the RichTextEdit control to a string in rich text format.
Cut	Long	Cuts (deletes) the selected contents from the RichTextEdit control (if any) to the clipboard.

RichTextEdit function	Data type returned	Description
DataSource	Integer	Associates a DataWindow with the RichTextEdit control, matching columns with input fields of the same name.
Drag	Integer	Starts or ends dragging of the control.
Find	Integer	Finds text in the control.
FindNext	Integer	Finds the next occurrence of text specified with the Find function.
GetAlignment	Alignment	Obtains the alignment of the selected text.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParagraphSetting	Long	Gets the size of the indentation, left margin, or right margin of the paragraph containing the insertion point.
GetParent	PowerObject	Returns a reference to the name of the parent object.
GetSpacing	Spacing	Obtains the line spacing of the selected text.
GetTextColor	Long	Returns the color of the selected text.
GetTextStyle	Boolean	Obtains font settings for the selected text.
Hide	Integer	Makes the control invisible.
InputFieldChangeData	Integer	Modifies the data value for all input fields of the given input field name.
InputFieldCurrentName	String	Gets the name of the selected input field.

RichTextEdit function	Data type returned	Description
InputFieldDeleteCurrent	Integer	Deletes the current occurrence of the selected input field. (That is, this function will not delete all input fields of the same name, only the current occurrence.)
InputFieldGetData	String	Obtains the data in the specified input field.
InputFieldInsert	Integer	Inserts the named input field at the insertion point.
InputFieldLocate	String	Locates an input field. You can find any occurrence of an input field or an input field with a specific name.
InsertDocument	Integer	Inserts the named file in the RichTextEdit control. The file can replace the current contents or be added at the insertion point. The file can be in rich text format (RTF) or ASCII text.
InsertPicture	Integer	Inserts the specified bitmap file at the insertion point.
IsPreview	Boolean	Checks whether the RichTextEdit control is in preview mode.
LineCount	Integer	Returns the total number of lines in the document.
LineLength	Integer	Returns the length of the current line.
Move	Integer	Moves the control to a specified location.
PageCount	Integer	Returns the number of pages in the document.
Paste	Integer	Inserts the contents of the clipboard (if any) at the insertion point in the control.

RichTextEdit function	Data type returned	Description
PasteRTF	Long	Inserts a string at the insertion point when the string contains text in rich text format. You can specify whether to insert the string in the header or footer band, as well as the main body (detail band).
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 or selected text in the control.
PostEvent	Integer	Adds the specified event to the end of the event queue for the specified object
Preview	Integer	Flips between Preview of the document (where header/footer, page size, and so on, are shown) and normal RTE control display.
Print	Integer	Prints the contents of the control. You can specify a range of pages and other settings.
ReplaceText	Integer	Replaces selected text with given string. If no text has been selected, insert given string at cursor location.
Resize	Integer	Changes the size of the control.
SaveDocument	Integer	Saves the current document in a file as either rich text format (RTF) or ASCII text.
Scroll	Integer	Scrolls the RichTextEdit the specified direction the specified number of lines.

RichTextEdit function	Data type returned	Description
ScrollNextPage	Integer	Scrolls forward to the next page in the RichTextEdit document. If the current page is the last page of a document instance, it scrolls to the next instance.
ScrollNextRow	Long	Scrolls the RichTextEdit to the next row and document instance.
ScrollPriorPage	Long	Scrolls back to the prior page in the RichTextEdit document. If the current page is the first page of a document instance, it scrolls to the prior instance.
ScrollPriorRow	Long	Scrolls the RichTextEdit to the prior row and document instance.
ScrollToRow	Long	Causes the control to scroll to the specified row and document instance.
SelectedColumn	Integer	Returns the number of the character column just after the insertion point.
SelectedLength	Long	Returns the length of the selected text.
SelectedLine	Long	Returns the number of the line in which the insertion point is currently located within the document instance.
SelectedPage	Long	Returns the number of the page in which the insertion point is currently located within the document instance.
SelectedStart	Integer	Returns the starting position of the selected text (if any) from the beginning of the line.
SelectedText	String	Returns a string containing the selected text from the control.
SelectText	Long	Selects text in the RichTextEdit control.

RichTextEdit function	Data type returned	Description
SelectTextAll	Integer	Selects all text in the control.
SelectTextLine	Integer	Selects all text in the line in which the insertion point is currently located.
SelectTextWord	Integer	Selects the word in which the insertion point is currently located.
SetAlignment	Integer`	Sets the alignment for the selected paragraphs.
SetFocus	Integer	Sets focus to the RichTextEdit control.
SetParagraphSetting	Integer	Specifies the size of the paragraph's indentation, left margin, or right margin.
SetPosition	Integer	Specifies the control's position in the front-to-back order within the window.
SetRedraw	Integer	Controls automatic redrawing of the control.
SetSpacing	Integer	Sets the line spacing for the selected paragraphs.
SetTextColor	Integer	Sets the color of the selected text.
SetTextStyle	Integer	Sets the font properties of the selected text.
Show	Integer	Makes the RichTextEdit control visible.
ShowHeadFoot	Integer	Allows editing of the header and footer in the RichTextEdit document.
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.

RichTextEdit function	Data type returned	Description
Undo	Integer	Cancels the previous editing function performed in the control, if the editing function can be undone. (Some editing functions cannot be undone.)

RoundRectangle control

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.

A RoundRectangle cannot be selected and therefore has no events.

Properties

Round Rectangle property	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>PowerScript Reference</i> .
FillPattern	FillPattern (enumerated)	Specifies the hatch pattern used to fill the control. Values are: BDiagonal! Diamond! FDiagonal! Horizontal! Solid! Square! Vertical! 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.

Round Rectangle property	Data type	Description
LineColor	Long	Specifies the numeric value of the line color: -2 to 16,777,215. For more information about color, see the <i>PowerScript Reference</i> .
LineStyle	LineStyle (enumerated)	Specifies the style of the line used to draw the control. Values are: Continuous! Dash! DashDot! DashDotDot! Dot! Transparent!
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. Values are: ◆ True — Control is visible ◆ False — Control is not visible
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

RoundRectangle function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
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.

ScriptDefinition object

Information about a script associated with a class definition. ScriptDefinition is used in the ClassDefinition object.

You can't instantiate a ScriptDefinition object for a particular script independently of a ClassDefinition object. Instead you access the ScriptDefinition instances that are elements of the ScriptList array of a ClassDefinition instance.

The ScriptDefinition object has information about:

- ◆ The script's name and whether it is a function or an event
- ◆ The return type, arguments, and local variables
- ◆ The source code
- ◆ Whether the script is defined locally or in an ancestor
- ◆ External function declarations

A ScriptDefinition object has no events.

Properties

ScriptDefinition property	Data type	Description
Access	VarAccess	The access level of the script (what objects can call the script). Values are: Private! Public! Protected! System!
AliasName	String	The alias value for an external function. The value is an empty string ("") for scripts that are not aliased external functions. Corresponds to the ALIAS FOR keyword in the external function declaration.

ScriptDefinition property	Data type	Description
ArgumentList	VariableDefinition	An unbounded array whose elements are VariableDefinition objects, one object per argument. The array is empty if there are no arguments.
ClassDefinition	PowerObject	An object of type ClassDefinition describing the ScriptDefinition class.
EventId	Long	The numeric event id for an event. For events that do not have an id, the value is -1.
EventIdName	String	The event id name for an event. For events that do not have an id, the value is an empty string ("").
ExternalUser Function	String	The filename of the DLL containing the external user function. The value is an empty string ("") for scripts that are not external user functions. Corresponds to the LIBRARY keyword in the external function declaration.
IsExternalEvent	Boolean	Indicates If this is an external event. External events are automatically generated events that get dispatched elsewhere.
IsLocallyDefined	Boolean	Indicates whether the event is defined at this level in the inheritance hierarchy. Values are: <ul style="list-style-type: none"> ◆ True — The event is defined at this level of the object's inheritance hierarchy ◆ False — The event is defined at an ancestor level IsLocallyDefined is not applicable to functions.

ScriptDefinition property	Data type	Description
IsLocallyScripted	Boolean	<p>Indicates whether the script is implemented at this level in the inheritance hierarchy. Values are:</p> <ul style="list-style-type: none"> ◆ True — There is code for the event or function at this level of the object's inheritance hierarchy ◆ False — There is no code for the event or function at this level
IsRPCFunction	Boolean	<p>Whether this is an RPC function. Values are:</p> <ul style="list-style-type: none"> ◆ True — Is an RPC function ◆ False — Is not an RPC function <p>Corresponds to the RPCFUNC keyword in a declaration for a stored procedure.</p>
IsScripted	Boolean	<p>Whether the event has a definition but no code at any level of the collapsed inheritance hierarchy. Values are:</p> <ul style="list-style-type: none"> ◆ True — The event has a script at some level of the object's inheritance hierarchy ◆ False — The event does not have a script <p>Only events can be defined but not scripted. For functions, IsScripted is always True.</p>
Kind	ScriptKind	<p>Whether the script is a function or event. Values are:</p> <p>ScriptEvent! ScriptFunction!</p>
LocalVariableList	VariableDefinition	<p>An unbounded array whose elements are VariableDefinition objects, one object per local variable. The array is empty if there are no local variables.</p>

ScriptDefinition property	Data type	Description
Name	String	The name of the script.
ReturnType	TypeDefinition	The type information of the return value. For scripts which do not return anything, ReturnType is an invalid object. Use the IsValid function to test the value.
Source	String	The source code for the script. Source is an empty string ("") if the source is not available (for example, when running an executable).
SystemFunction	String	For built-in PowerBuilder functions, the filename of the DLL containing the function. The value is an empty string ("") for scripts that are not built-in PowerBuilder functions.

Functions

ScriptDefinition function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object	Returns the type of the object.

SimpleTypeDefinition object

Information about the type of a scalar variable. SimpleTypeDefinition is inherited from TypeDefinition and has no additional properties or functions.

FOR INFO For the list of properties and functions, see the TypeDefinition object.

SingleLineEdit control

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.

Properties

SingleLineEdit property	Data type	Description
Accelerator	Integer	Specifies 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. Values are: <ul style="list-style-type: none"> ◆ True — Control automatically scrolls horizontally ◆ False — Control does not automatically scroll horizontally
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215. For more information about color, see the <i>PowerScript Reference</i> .
Border	Boolean	Specifies whether the control has a border. Values are: <ul style="list-style-type: none"> ◆ True — Control has a border ◆ False — Control does not have a border
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order in the window. Values are: <ul style="list-style-type: none"> ◆ True — Control moved to top ◆ False — Control not moved to top

SingleLineEdit property	Data type	Description
DisplayOnly	Boolean	<p>Specifies whether the text in the control is display-only and cannot be changed by the user. Values are:</p> <ul style="list-style-type: none"> ◆ True — Text cannot be changed by user ◆ False — Text can be changed by user
DragAuto	Boolean	<p>Specifies whether PowerBuilder puts the control automatically into Drag mode. Values are:</p> <ul style="list-style-type: none"> ◆ 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	<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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control can be selected ◆ False — Control cannot be selected
FaceName	String	<p>Specifies the name of the typeface in which the text of the control displays (for example, ARIAL or COURIER).</p>

SingleLineEdit property	Data type	Description
FontCharSet	FontCharSet (enumerated)	Specifies the font character set used for the text in the control. Values are: ANSI! ChineseBig5! DefaultCharSet! Hangeul! OEM! ShiftJIS! Symbol!
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) used for the text in the control. Values are: AnyFont! Decorative! Modern! Roman! Script! Swiss!
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font used for the text in the control. Values are: Default! Fixed! Variable!
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. Values are: ◆ True — Text does not stay highlighted ◆ False — Text stays highlighted
Italic	Boolean	Specifies whether the text in the control is italic. Values are: ◆ True — Text is italic ◆ False — Text is not italic
Limit	Integer	Specifies the maximum number of characters (0 to 32,767) that can be entered in the control (0 means unlimited).

SingleLineEdit property	Data type	Description
Password	Boolean	Specifies whether the control is a password field (whether asterisks appear when the user types characters). Values are: <ul style="list-style-type: none"> ◆ True — Control is a password field ◆ False — Control is not a password field
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
RightToLeft	Boolean	Specifies that characters should be displayed in right-to-left order. The application must be running on a Hebrew or Arabic version of PowerBuilder under an operating system that supports right-to-left display. Values are: <ul style="list-style-type: none"> ◆ True — Characters display in right-to-left order ◆ False — Characters display in left-to-right order
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. Values are: <p>AnyCase! Lower! Upper!</p>
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>PowerScript 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.

SingleLineEdit property	Data type	Description
Underline	Boolean	Specifies whether the text in the control is underlined. Values are: <ul style="list-style-type: none"> ◆ True — Text is underlined ◆ False — Text is not underlined
Visible	Boolean	Specifies whether the control is visible. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
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

SingleLineEdit 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).

SingleLineEdit event	Occurs
LoseFocus	When the control loses focus (becomes inactive).
Modified	When the control text has been changed and the user presses ENTER or TAB or changes focus to another control.
Other	When a Windows message occurs that is not a PowerBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

Functions

SingleLineEdit 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.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
Hide	Integer	Makes the control invisible.
Move	Integer	Moves the control to a specified location.

SingleLineEdit function	Data type returned	Description
Paste	Integer	Inserts the contents of the clipboard (if any) at the insertion point in the control and 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.
PostEvent	Boolean	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 properties.
Show	Integer	Makes the control visible.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.

SingleLineEdit function	Data type returned	Description
TypeOf	Object	Returns the type of the control.
Undo	Integer	Cancels the previous editing function performed in the control.

StaticText control

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

Properties

StaticText property	Data type	Description
Alignment	Alignment (enumerated)	Specifies the text alignment in the control. Values are: Left! Center! Right!
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215. For more information about color, see the <i>PowerScript Reference</i> .
Border	Boolean	Specifies whether the control has a border. Values are: ◆ True — Control has a border ◆ False — Control does not have a border
BorderColor	Long	Specifies the numerical value of the border color: -2 to 16,777,215. For more information about colors, see the <i>PowerScript Reference</i> .
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order of the window. Values are: ◆ True — Control moved to top ◆ False — Control not moved to top

StaticText property	Data type	Description
DragAuto	Boolean	<p>Specifies whether PowerBuilder puts the control automatically into Drag mode.</p> <p>Values are:</p> <ul style="list-style-type: none"> ◆ 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	<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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control can be selected ◆ False — Control cannot be selected
FaceName	String	<p>Specifies the name of the typeface in which the text of the control will display (for example, ARIAL or COURIER).</p>

StaticText property	Data type	Description
FillPattern	FillPattern (enumerated)	<p>Specifies the hatch pattern used to fill the control. Values are:</p> <p>BDiagonal! Diamond! FDiagonal! Horizontal! Solid! Square! Vertical!</p> <p>FDiagonal! is lines going from the lower left to the upper right. BDiagonal! is lines going from the upper left to the lower right.</p>
FocusRectangle	Boolean	<p>Specifies whether a dotted rectangle (focus rectangle) frames the control when it has focus. Values are:</p> <ul style="list-style-type: none"> ◆ True — Control framed when it has focus ◆ False — Control not framed when it has focus
FontCharSet	FontCharSet (enumerated)	<p>Specifies the font character set used for the text in the control. Values are:</p> <p>ANSI! ChineseBig5! DefaultCharSet! Hangeul! OEM! ShiftJIS! Symbol!</p>
FontFamily	FontFamily (enumerated)	<p>Specifies the font family (type style) used for the text in the control. Values are:</p> <p>AnyFont! Decorative! Modern! Roman! Script! Swiss!</p>
FontPitch	FontPitch (enumerated)	<p>Specifies the pitch (spacing) of the font used for the text in the control. Values are:</p> <p>Default! Fixed! Variable!</p>

StaticText property	Data type	Description
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. Values are: <ul style="list-style-type: none"> ◆ True — Text is italic ◆ False — Text is not italic
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
RightToLeft	Boolean	Specifies that characters should be displayed in right-to-left order. The application must be running on a Hebrew or Arabic version of PowerBuilder under an operating system that supports right-to-left display. Values are: <ul style="list-style-type: none"> ◆ True — Characters display in right-to-left order ◆ False — Characters display in left-to-right order
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>PowerScript 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.

StaticText property	Data type	Description
Underline	Boolean	Specifies whether the text in the control is underlined. Values are: <ul style="list-style-type: none"> ◆ True — Text is underlined ◆ False — Text is not underlined
Visible	Boolean	Specifies whether the control is visible. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
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

StaticText 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.

StaticText event	Occurs
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

StaticText function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
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 focus to the control.
SetPosition	Integer	Specifies the position of the control in the front-to-back order of the window.

StaticText function	Data type returned	Description
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its properties.
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.

Tab control

A Tab control contains tab pages, which are user objects that contain controls. Tab pages can be defined within the Tab control or they can be defined in the User Object painter and inserted into the Tab control.

Each tab page can have its own label, picture, and background color.

All tab pages share the same font settings.

Tab Position controls where the tabs on the tab pages are displayed. Tabs can be displayed on any one of the four edges of the Tab control. They can also be displayed on opposite edges with the tabs before or after the selected tab jumping to the other edge.

Properties

Tab property	Data type	Description
Alignment	Alignment (enumerated)	Specifies the alignment of the text labeling all the tabs. Values are: Left! Center! Right!
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215. For more information about color, see the RGB function.
BoldSelectedText	Boolean	Specifies whether the text for the selected tab is bold. Values are: ◆ True — The text on the selected tab is bold ◆ False — The text on the selected tab has the same setting as the other tabs
BringToTop	Boolean	Specifies whether PowerBuilder will move the control to the top of the front-to-back order in the window. Values are: ◆ True — Will move to the top ◆ False — Will not move to the top

Tab property	Data type	Description
CreateOnDemand	Boolean	<p>Specifies whether PowerBuilder creates graphical representations of controls on all tab pages when the Tab control is created. Values are:</p> <ul style="list-style-type: none"> ◆ True — Graphical representations of tab pages are not created until the tab page is selected. ◆ False — (Default) Graphical representations of tab pages are created when the Tab control is created. <p>When CreateOnDemand is True, all controls on all tab pages of a Tab control are instantiated when the Tab control is created, but the Constructor event of controls on hidden tab pages is not triggered until the user views the tab page. Until the Constructor event for a control has run, a graphical representation of the control is not created, and its handle is not available. Constructor events for controls on the selected tab page are always triggered when the Tab control is created.</p> <p>A window will open more quickly if the creation of a graphical representation is delayed for tab pages with many controls. However, scripts cannot refer to a control on a tab page until its Constructor event has run and a graphical representation has been created.</p>
Control[]	UserObject	Specifies the array of tab pages within the Tab control.

Tab property	Data type	Description
DragAuto	Boolean	<p>Specifies whether PowerBuilder will put the control automatically into Drag mode. Values are:</p> <ul style="list-style-type: none">◆ 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	<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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
Enabled	Boolean	<p>Specifies whether the control is enabled (can be selected). Values are:</p> <ul style="list-style-type: none">◆ True — Can be selected◆ False — Cannot be selected
FaceName	String	<p>Specifies the name of the typeface used for the text labels on tabs (for example, ARIAL or COURIER).</p>

Tab property	Data type	Description
FixedWidth	Boolean	<p>Specifies whether tabs have a fixed width, meaning they do not shrink to the length of their text labels. Values are:</p> <ul style="list-style-type: none"> ◆ True — Tab width is fixed; the width is determined by the longest text label ◆ False — Tab width adjusts to the length of the text labels
FocusOnButton Down	Boolean	<p>Specifies whether each tab gets focus when the user clicks on it. Values are:</p> <ul style="list-style-type: none"> ◆ True — The tab the user clicks on gets focus; a dotted rectangle marks the tab ◆ False — The tab does not get focus <p>In either case, the selected tab page comes to the front.</p> <p>The dotted focus rectangle will appear on the tab when the user clicks on it a second and subsequent times even if this property is set to False.</p>
FontCharSet	FontCharSet (enumerated)	<p>Specifies the character set for the text labels on the tabs. Values are:</p> <p>ANSI! ChineseBig5! DefaultCharset! HangEul! OEM! Shiftjis! Symbol!</p>
FontFamily	FontFamily (enumerated)	<p>Specifies the font family (type style) for the text labels on the tabs. Values are:</p> <p>AnyFont! Decorative! Modern! Roman! Script! Swiss!</p>

Tab property	Data type	Description
FontPitch	FontPitch (enumerated)	Specifies the pitch (spacing) of the font for text labels on the tabs. Values are: Default! Fixed! Variable!
Height	Integer	Specifies the height of the control, in PowerBuilder units.
Italic	Boolean	Specifies whether the text on the tabs is italic. Values are: ◆ True — Text labels are italic ◆ False — Text labels are not italic
Multiline	Boolean	Specifies whether the tabs can appear in more than one row. Values are: ◆ True — If there is not room for all the tabs in a single row, the tabs will be arranged in multiple rows ◆ False — If there is not room for all the tabs in a single row, a dual arrow control will display to allow the user to scroll to tabs that don't fit
PerpendicularText	Boolean	Specifies whether the tab labels are drawn perpendicular to the tab page. Values are: ◆ True — Text is perpendicular to the edge of the tab page, resulting in narrower tabs ◆ False — Text runs along the edge of the tab page, resulting in wider tabs
PictureOnRight	Boolean	Specifies whether a picture that is part of the tab label is to the right or left of the text. Values are: ◆ True — The picture is on the right ◆ False — The picture is on the left
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.

Tab property	Data type	Description
PowerTips	Boolean	<p>Specifies whether PowerTipText for a tab page is displayed as a PowerTip (a popup label for the tab) when the mouse pointer pauses over the tab. PowerTips are useful if the tabs are labeled with pictures. Values are:</p> <ul style="list-style-type: none"> ◆ True — PowerTipText, if any, displays as a popup label for each tab ◆ False — No PowerTips are displayed
RaggedRight	Boolean	<p>Specifies whether tabs are stretched so that they fill space along the edge of the control. Values are:</p> <ul style="list-style-type: none"> ◆ True — Tabs remain the size determined by their label text and the FixedWidth property ◆ False — Tabs are stretched to fill the edge
SelectedTab	Integer	<p>Specifies the index number of the selected tab. The default value is 1, and the integer must be in the range 1 to N, where N is the number of tab pages</p>
ShowPicture	Boolean	<p>Specifies whether the picture selected for each tab is displayed. Values are:</p> <ul style="list-style-type: none"> ◆ True — The picture for each tab, if any, is displayed ◆ False — No pictures are displayed
ShowText	Boolean	<p>Specifies whether the text specified for each tab label is displayed. Values are:</p> <ul style="list-style-type: none"> ◆ True — The text for each tab is displayed ◆ False — The text for each tab is not displayed
TabOrder	Integer	<p>Specifies the tab value of the control within the window (0 means the user cannot tab to the control).</p>

Tab property	Data type	Description
TabPosition	TabPosition (enumerated)	<p>Specifies where the tabs appear around the Tab control. Values are:</p> <ul style="list-style-type: none"> ◆ TabsOnBottom! — Tabs are at the bottom ◆ TabsOnBottomAndTop! — Tabs before the selected tab are on top; the selected tab itself and tabs after it are on the bottom ◆ TabsOnLeft! — Tabs are on the left ◆ TabsOnLeftAndRight! — Tabs before the selected tab and the selected tab itself are on the left; tabs after the selected tab are on the right ◆ TabsOnRight! — Tabs are on the right ◆ TabsOnRightAndLeft! — Tabs before the selected tab are on the left; the selected tab and tabs after it are on the right ◆ TabsOnTop! — Tabs are on top ◆ TabsOnTopAndBottom! — Tabs before the selected tab and the selected tab itself are on top; tabs after the selected tab are on the bottom
Tag	String	Specifies the tag value assigned to 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	<p>Specifies whether the text on the tabs is underlined. Values are:</p> <ul style="list-style-type: none"> ◆ True — Text labels are underlined ◆ False — Text labels are not underlined

Tab property	Data type	Description
Visible	Boolean	Specifies whether the control is visible. Values are: <ul style="list-style-type: none"> ◆ True — Is visible ◆ False — Is not visible
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

Tab Event	Occurs
Clicked	When the user clicks in the Tab control, except in the display area of the tab page. For the tab page, the Clicked event (pbm_bnclicked) for the user object is triggered instead.
Constructor	When the object is created, immediately before the Open event occurs in the window.
Destructor	When the object is destroyed, immediately after the Close event occurs in the window.
DoubleClicked	When the user double-clicks in the Tab control, except in the display area of the tab page. For the tab page, the DoubleClicked event (pbm_bndoubleclicked) for the user object is triggered instead.
DragDrop	When a dragged control is dropped in the tab area of the control.
DragEnter	When a dragged control enters the control, including entering the narrow border around the display area.

Tab Event	Occurs
DragLeave	When a dragged control leaves the control, including leaving by crossing into the tab page display area.
DragWithin	When a dragged control is within the control, but not within the tab page display area.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
Key	When the user presses a key.
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PowerBuilder event.
RightClicked	When the user clicks with the right mouse button in the Tab control, except in the display area of the tab page. If the user right clicks on the tab page, the controls or user objects on the tab page get an RButtonDown event.
RightDoubleClicked	When the user double-clicks with the right mouse button in the Tab control, except in the display area of the tab page. For the tab page, the RightDoubleClicked event (pbm_rbuttondblclk) for the user object is triggered instead.
SelectionChanged	Just after the selection changes to another tab. SelectionChanged is triggered when the tab is created and the initial selection is established.
SelectionChanging	Just before the selection changes to another tab. To prevent the selection from changing, return 1 in the event script. SelectionChanging is triggered when the tab is created and the initial selection is established.

Functions

Tab function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
CloseTab	Integer	Closes a tab page that was opened with the OpenTab function.
Drag	Integer	Starts or ends dragging of the control.

Tab function	Data type returned	Description
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the parent of the Tab control
Hide	Integer	Makes the control invisible.
Move	Integer	Moves the control to a specified location.
MoveTab	Integer	Moves a tab to a new position in the order of tabs.
OpenTab	Integer	Opens the specified user object as a tab page, making its properties available to scripts.
OpenTabWithParm	Integer	Opens the user object as a tab page, making its properties available to scripts, and stores a parameter in the Message object.
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	Integer	Adds the specified event to the end of the event queue for the specified object.
Print	Integer	Prints the contents of the control. You can specify a range of pages and other settings.
Resize	Integer	Changes the size of the tab page to the size specified in the width and height arguments.
SelecltTab	Integer	Selects a tab page.
SetFocus	Integer	Sets focus to the Tab control.
SetPosition	Integer	Specifies the control's position in the front-to-back order within the window.

Tab function	Data type returned	Description
SetRedraw	Integer	Controls automatic redrawing of the control.
Show	Integer	Makes the control visible.
TabPostEvent	Integer	Adds an event to the end of the message queues for each of the tab pages.
TabTriggerEvent	Integer	Sends an event to every tab page and, for each page, executes the script associated with the event.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.

Timing

Timing is a nonvisual system object that can be used when a Timer event cannot be associated with a window. To use a timing object, create a standard class user object that inherits from the Timing system class, and then create an instance of the inherited timing object.

Properties

Timing property	Data type	Description
ClassDefinition	PowerObject	An object of type ClassDefinition describing the Timing class.
Interval	Double	(Read-only) The minimum number of seconds the timing object waits between calls to the Timer event. This property is initially set to NULL and is modified whenever the Start function is called.
Running	Boolean	(Read-only) True if the timing object has been started and is currently running. False if the timing object is not running.

Events

Timing event	Occurs
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
Timer	When a specified number of seconds elapses after the Start function has been called.

Functions

Timing function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
PostEvent	Boolean	Adds an event to the end of the message queue for the object.
Start	Integer	Activates the timing object using a specified interval.
Stop	Integer	Deactivates the timing object.
TriggerEvent	Integer	Triggers a specific event in the object and executes the script for the event.
TypeOf	Object	Returns the type of the object.

TraceActivityNode object

The TraceActivityNode object provides information about the nodes in a trace file, including the type of activity represented by a node. You use the TraceActivityNode object in conjunction with the TraceFile object.

The TraceActivityNode object has no events.

Properties

TraceActivityNode property	Data type	Description
ActivityType	TraceActivity (enumerated)	<p>A value of the enumerated data type TraceActivity that identifies the activity represented by the node. Values are:</p> <ul style="list-style-type: none">◆ ActBegin!—Start and finish of logging◆ ActError!—Occurrences of system errors and warnings◆ ActESQL!—Embedded SQL statement entry and exit◆ ActGarbageCollect!—Start and finish of garbage collection◆ ActLine!—Routine line hits◆ ActObjectCreate!—Object creation◆ ActObjectDestroy!—Object destruction◆ ActRoutine!—Routine entry and exit◆ ActUser!—Occurrences of an activity you selected

TraceActivityNode property	Data type	Description
Category	TraceCategory (enumerated)	The category of the activity represented by the node. Values are: <ul style="list-style-type: none"> ◆ TraceAtomic!—The node is an activity that occurred in a single statement. ◆ TraceIn!—The node is the beginning of an activity that spans several statements. ◆ TraceOut!—The node is the end of an activity that spanned several statements.
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceActivityNode function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object (enumerated)	Returns the type of the object.

TraceBeginEnd object

The TraceBeginEnd object provides information about a node in a trace file identified as an occurrence of a logging start or finish. To access the extra properties of the TraceBeginEnd object, you assign a TraceActivityNode object whose activity type is ActBegin! to the TraceBeginEnd object.

The TraceBeginEnd object has no events.

Properties

TraceBeginEnd property	Data type	Description
ActivityType	TraceActivity (enumerated)	The value ActBegin! which identifies the activity represented by the node as an occurrence of a logging start or finish.
Category	TraceCategory (enumerated)	The category of the activity represented by the node. Values are: <ul style="list-style-type: none">◆ TraceIn!—The node is the beginning of an activity that spans several statements.◆ TraceOut!—The node is the end of an activity that spanned several statements.
Message	String	The message passed in for the TraceBegin function. For the TraceEnd function, the value is " ".
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceBegin End function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.

TraceBegin End function	Data type returned	Description
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object (enumerated)	Returns the type of the object.

TraceError object

The TraceError object provides information about a node in a trace file identified as an occurrence of a system error or warning, including the error message and severity level. To access the extra properties of the TraceError object, you assign a TraceActivityNode object whose activity type is ActError! to the TraceError object.

The TraceError object has no events.

Properties

TraceError property	Data type	Description
ActivityType	TraceActivity (enumerated)	The value ActError! which identifies the activity represented by the node as an occurrence of a system error or warning.
Category	TraceCategory (enumerated)	The value TraceAtomic! which indicates that the node is an activity that occurred in a single statement.
Message	String	The system error message or the message passed to the TraceError function.
Severity	Long	The system error severity or the TraceError argument passed to the function.
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceError function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.

TraceError function	Data type returned	Description
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object (enumerated)	Returns the type of the object.

TraceESQL object

The TraceESQL object provides information about a node in a trace file identified as an occurrence of an Embedded SQL (ESQL) statement. To access the extra properties of the TraceESQL object, you assign a TraceActivityNode object whose activity type is ActESQL! to the TraceESQL object.

The TraceESQL object has no events.

Properties

TraceESQL property	Data type	Description
ActivityNode	TraceActivity	The value ActESQL! which identifies the activity represented by the node as an occurrence of an ESQL statement entry or exit.
Category	TraceCategory	The category of the activity represented by the node. Values are: <ul style="list-style-type: none">◆ TraceIn!—The node is the beginning of an activity that spans several statements.◆ TraceOut!—The node is the end of an activity that spanned several statements.
Name	String	The name of the ESQL statement.
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceESQL function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.

TraceESQL function	Data type returned	Description
TypeOf	Object (enumerated)	Returns the type of the object.

TraceFile object

The TraceFile object is used to access the contents of a trace file created from a PowerBuilder application. Unlike the Profiling and TraceTree objects, the TraceFile object does not provide properties and functions to create an analysis model. You use the TraceFile object in conjunction with the TraceActivityNode, TraceBeginEnd, TraceError, TraceESQL, TraceGarbageCollect, TraceLine, TraceObject, TraceRoutine, and TraceUser objects.

The TraceFile object has no events.

Properties

TraceFile property	Data type	Description
ApplicationName	String	The name of the application used to generate the trace file.
CollectionTime	Decimal	The amount of time (in seconds) taken by the collection of trace data. This time has already been accounted for in the time stamps from the trace file, that is, the proper amount of time has been subtracted from the time stamps before they are put in the trace file. The value is NULL if the file is not open.
LastError	ErrorReturn (enumerated)	The error code for the last error that occurred
NumberOfActivities	Long	The number of activities that exist in the trace file.
FileName	String	The name of the opened trace file. The value is an empty string if the file is not open.

Functions

TraceFile function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
Close	ErrorReturn (enumerated)	Closes the open trace file.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
NextActivity	TraceActivityNode	Provides the next activity in the trace stream. If there are no more activities, or if the file is not open, an invalid object is returned. Use the LastError property to determine what kind of error occurred.
Open	ErrorReturn (enumerated)	Opens for reading the trace file with the passed name.
Reset	ErrorReturn (enumerated)	Resets the stream to the beginning of the trace file.
TypeOf	Object (enumerated)	Returns the type of the object.

TraceGarbageCollect object

The TraceGarbageCollect object provides information about a node in a trace file identified as an occurrence of garbage collection. To access the extra properties of the TraceGarbageCollect object, you assign a TraceActivityNode object whose activity type is ActGarbageCollect! to the TraceGarbageCollect object.

The TraceGarbageCollect object has no events.

Properties

TraceGarbageCollect property	Data type	Description
ActivityType	TraceActivity (enumerated)	The value ActGarbageCollect! which identifies the activity represented by the node as an occurrence of garbage collection start or finish
Category	TraceCategory (enumerated)	The category of the activity represented by the node. Values are: <ul style="list-style-type: none">◆ TraceIn!—The node is the beginning of an activity that spans several statements.◆ TraceOut!—The node is the end of an activity that spanned several statements.
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceGarbageCollect function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.

TraceGarbageCollect function	Data type returned	Description
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object (enumerated)	Returns the type of the object.

TraceLine object

The TraceLine object provides information about a node in a trace file identified as an occurrence of a routine line hit. To access the extra properties of the TraceLine object, you assign a TraceActivityNode object whose activity type is ActLine! to the TraceLine object.

The TraceLine objects has no events.

Properties

TraceLine property	Data type	Description
ActivityType	TraceActivity (enumerated)	The value ActLine! which identifies the activity represented by the node as an occurrence of a routine line hit.
Category	TraceCategory (enumerated)	The value TraceAtomic! which indicates that the node is an activity that occurred in a single statement.
LineNumber	UnsignedLong	The line number.
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceLine function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object (enumerated)	Returns the type of the object.

TraceObject object

The TraceObject object provides information about a node in a trace file identified as the creation or destruction of an object. To access the extra properties of the TraceObject object, you assign a TraceActivityNode object whose activity type is ActObjectCreate! or ActObjectDestroy! to the TraceObject object.

The TraceObject object has no events.

Properties

TraceObject property	Data type	Description
ActivityType	TraceActivity (enumerated)	The value ActObjectCreate! or ActObjectDestroy! which identifies the activity represented by the node as the creation or destruction of an object.
Category	TraceCategory (enumerated)	The category of the activity represented by the node. Values are: <ul style="list-style-type: none"> ◆ TraceIn!—The node is the beginning of an activity that spans several statements. ◆ TraceOut!—The node is the end of an activity that spanned several statements.
ClassName	String	The name of the class that is the type of the object. Nested classes (like controls in windows) have a name of the form <i>class name`embedded class name</i> .
IsCreate	Boolean	True if the node represents the creation of an object and False if the node represents the destruction of an object.
LibraryName	String	The name of the library that contains the class of the object. The value is " " for system classes.
ObjectID	UnsignedLong	A unique identifier for the object.

TraceObject property	Data type	Description
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceObject function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object (enumerated)	Returns the type of the object.

TraceRoutine object

The TraceRoutine object provides information about a node in a trace file identified as an occurrence of a routine. To access the extra properties of the TraceRoutine object, you assign a TraceActivityNode object whose activity type is ActRoutine! to the TraceRoutine object.

The TraceRoutine object has no events.

Properties

TraceRoutine property	Data type	Description
ActivityType	TraceActivity (enumerated)	The value ActRoutine! which identifies the activity represented by the node as an occurrence of a routine entry or exit.
Category	TraceCategory (enumerated)	The category of the activity represented by the node. Values are: <ul style="list-style-type: none"> ◆ TraceIn!—The node is the beginning of an activity that spans several statements. ◆ TraceOut!—The node is the end of an activity that spanned several statements.
ClassName	String	The name of the class that contains the routine. The value is " " for system functions. Nested classes (like controls in windows) have a name of the form <i>class name`embedded class name</i> .
IsEvent	Boolean	True if the routine is an event and False if the routine is a function.
LibraryName	String	The name of the library that contains the class that includes the routine. The value is " " for system classes.
Name	String	The name of the routine including the parameter data types and return value.

TraceRoutine property	Data type	Description
ObjectID	UnsignedLong	The object ID for the object on which the routine is executing. The ID is 0 if the routine executing is a global or system routine.
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceRoutine function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object (enumerated)	Returns the type of the object.

TraceTree object

The TraceTree object is used to analyze the performance of a PowerBuilder application. It provides a tree model listing all the nodes logged in a given trace file. It includes the functions you call to name the trace file to be analyzed, build the tree model, and list the top-level entries in the tree model. You use the TraceTree object in conjunction with the TraceTreeNode, TraceTreeError, TraceTreeESQL, TraceTreeGarbageCollect, TraceTreeLine, TraceTreeObject, TraceTreeRoutine, and TraceTreeUser objects.

The TraceTree object has no events.

Properties

TraceTree property	Data type	Description
ApplicationName	String	The name of the application used to generate the trace file.
CollectionTime	Decimal	The amount of time (in seconds) taken by the collection of trace data. This time has already been accounted for in the timestamps from the trace file (the proper amount of time has been subtracted from the timestamps before they are put in the trace file). The value is NULL if no model was built.
NumberOfActivities	Long	The total number of activities that exist in the trace file. The value is 0 if this is called before the trace file name is set.
TraceFileName	String	The name of the trace file to use to build the model. The value is an empty string if the name has not been successfully set.

Functions

TraceTree function	Data type returned	Description
BuildModel	ErrorReturn (enumerated)	Builds a tree model based on the previously specified trace file.
ClassName	String	Returns the name assigned to the object.
DestroyModel	ErrorReturn (enumerated)	Destroys the current tree model.
EntryList	ErrorReturn (enumerated)	Provides a list of top-level entries (defined as TraceTreeNode objects) included in the model.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
SetTraceFileName	ErrorReturn	Indicates the name of the trace file to use for analysis and creates the format of the file header.
TypeOf	Object (enumerated)	Returns the type of the object.

TraceTreeError object

The TraceTreeError object provides information about a tree model node identified as an occurrence of a system error or warning, including the error message and severity level. To access the extra properties of the TraceTreeError object, you assign a TraceTreeNode object whose activity type is ActError! to the TraceTreeError object.

The TraceTreeError object has no events.

Properties

TraceTreeError property	Data type	Description
ActivityType	TraceActivity (enumerated)	The value ActError! which identifies the activity represented by the node as an occurrence of a system error or warning.
Message	String	The system error message or the message passed to the TraceError function.
ParentNode	TraceTreeNode	The parent of this node.
Severity	Long	The system error severity or the TraceError argument passed to the function.
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceTree-Error function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.

TraceTreeError object

TraceTree-Error function	Data type returned	Description
TypeOf	Object (enumerated)	Returns the type of the object.

TraceTreeESQL object

The TraceTreeESQL object provides information about a tree model node identified as an occurrence of an Embedded SQL (ESQL) statement. To access the extra properties of the TraceTreeESQL object, you assign a TraceTreeNode object whose activity type is ActESQL! to the TraceTreeESQL object.

The TraceTreeESQL object has no events.

Properties

TraceTreeESQL property	Data type	Description
ActivityNode	TraceActivity (enumerated)	The value ActESQL! which identifies the activity represented by the node as an occurrence of an ESQL statement entry and exit.
EnterTimerValue	Decimal	The timer value (in seconds) of the entry for this statement.
ExitTimerValue	Decimal	The timer value (in seconds) of the exit for this statement.
Name	String	The name of the ESQL statement.
ParentNode	TraceTreeNode	The parent of this node.

Functions

TraceTreeESQL function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object (enumerated)	Returns the type of the object.

TraceTreeGarbageCollect object

The TraceTreeGarbageCollect object provides information about a tree model node identified as an occurrence of garbage collection, including the children or classes and routines called by that node. To access the extra properties of the TraceTreeGarbageCollect object, you assign a TraceTreeNode object whose activity type is ActGarbageCollect! to the TraceTreeGarbageCollect object.

The TraceTreeGarbageCollect object has no events.

Properties

TraceTreeGarbageCollect property	Data type	Description
ActivityType	TraceActivity (enumerated)	The value ActGarbageCollect! which identifies the activity represented by the node as garbage collection start and finish.
EnterTimerValue	Decimal	The timer value (in seconds) of the entry for the garbage collector.
ExitTimerValue	Decimal	The timer value (in seconds) of the exit for the garbage collector.
ParentNode	TraceTreeNode	The parent of this node.

Functions

TraceTreeGarbageCollect function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetChildrenList	ErrorReturn (enumerated)	Provides a list of the children (defined as TraceTreeNode objects) of this routine.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.

TraceTreeGarbage Collect function	Data type returned	Description
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object (enumerated)	Returns the type of the object.

TraceTreeLine object

The TraceTreeLine object provides information about a tree model node identified as an occurrence of a routine line hit. To access the extra properties of the TraceTreeLine object, you assign a TraceTreeNode object whose activity type is ActLine! to the TraceTreeLine object.

The TraceTreeLine object has no events.

Properties

TraceTreeLine property	Data type	Description
ActivityType	TraceActivity (enumerated)	The value ActLine! which identifies the activity represented by the node as an occurrence of a routine line hit.
LineNumber	UnsignedLong	The line number.
ParentNode	TraceTreeNode	The parent of this node.
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceTreeLine function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object (enumerated)	Returns the type of the object.

TraceTreeNode object

The TraceTreeNode object provides information about the nodes in the tree model, including the type of activity represented by the node. You use the TraceTreeNode object in conjunction with the TraceTree object.

The TraceTreeNode object has no events.

Properties

TraceTreeNode property	Data type	Description
ActivityType	TraceActivity (enumerated)	<p>A value of the enumerated data type TraceActivity that identifies the activity represented by the node. Values are:</p> <ul style="list-style-type: none"> ◆ ActBegin!—Start and finish of logging ◆ ActError!—Occurrences of system errors and warnings ◆ ActESQL!—Embedded SQL statement entry and exit ◆ ActGarbageCollect!—Start and finish of garbage collection ◆ ActLine!—Routine line hits ◆ ActObjectCreate!—Object creation ◆ ActObjectDestroy!—Object destruction ◆ ActRoutine!—Routine entry and exit ◆ ActUser!—Occurrences of an activity you selected
ParentNode	TraceTreeNode	The parent of this node. If the parent is a top-level node, that is, a node returned by the EntryList function, the value is an invalid object.

Functions

TraceTreeNode function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object (enumerated)	Returns the type of the object.

TraceTreeObject object

The TraceTreeObject object provides information about a tree model node identified as an occurrence of an object. To access the extra properties of the TraceTreeObject object, you assign a TraceTreeNode object whose activity type is ActObjectCreate! or ActObjectDestroy! to the TraceTreeObject object.

The TraceTreeObject object has no events.

Properties

TraceTreeObject property	Data type	Description
ActivityType	TraceActivity (enumerated)	The value ActObjectCreate! or ActObjectDestroy! which identifies the activity represented by the node as object creation or destruction.
ClassName	String	The name of the class that is the object type.
Create	Boolean	True if the node represents the creation of an object and False if it represents the destruction of an object.
EnterTimerValue	Decimal	The timer value (in seconds) when the activity began.
ExitTimerValue	Decimal	The timer value (in seconds) when the activity ended.
LibraryName	String	The name of the library that contains the class of the object. The value is " " for system classes.
ObjectID	UnsignedLong	The internal identifier for the object.
ParentNode	TraceTreeNode	The parent of this node.

Functions

TraceTreeObject function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetChildrenList	ErrorReturn (enumerated)	Provides a list of the children (defined as TraceTreeNode objects) of this object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object (enumerated)	Returns the type of the object.

TraceTreeRoutine object

The TraceTreeRoutine object provides information about a tree model node identified as an occurrence of a routine. To access the extra properties of the TraceTreeRoutine object, you assign a TraceTreeNode object whose activity type is ActRoutine! to the TraceTreeRoutine object.

The TraceTreeRoutine object has no events.

Properties

TraceTreeRoutine property	Data type	Description
ActivityType	TraceActivity (enumerated)	The value ActRoutine! which identifies the activity represented by the node as an occurrence of a routine entry and exit.
ClassName	String	The name of the class that contains the routine. The value is " " for system functions. Nested classes (like controls on a window) will have a name of the form <i>class name`embedded class name</i> .
EnterTimerValue	Decimal	The timer value (in seconds) of the entry for this call.
ExitTimerValue	Decimal	The timer value (in seconds) of the exit for this call.
IsEvent	Boolean	True if the routine is an event and False if the routine is a function.
LibraryName	String	The name of the library that contains the class that contains the routine. The value is " " for system classes.
Name	String	The name of the routine including the argument data types and return value.
ObjectID	UnsignedLong	The internal ID for the object on which the routine is executing. The value is 0 for global and system functions.

TraceTreeRoutine property	Data type	Description
ParentNode	TraceTreeNode	The parent of this node.

Functions

TraceTreeRoutine function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetChildrenList	ErrorReturn (enumerated)	Provides a list of the children (defined as TraceTreeNode objects) of this routine.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object (enumerated)	Returns the type of the object.

TraceTreeUser object

The TraceTreeUser object provides information about a tree model node identified as an occurrence of an activity you selected for logging, including the activity argument and message. To access the extra properties of the TraceTreeUser object, you assign a TraceTreeNode object whose activity type is ActUser! to the TraceTreeUser object.

The TraceTreeUser object has no events.

Properties

TraceTreeUser property	Data type	Description
ActivityType	TraceActivity (enumerated)	The value ActUser! which identifies the activity represented by the node as an occurrence of an activity you selected.
Argument	Long	The argument passed to the TraceUser function.
Message	String	The message passed to the TraceUser function.
ParentNode	TraceTreeNode	The parent of this node.
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceTreeUser function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.

TraceTreeUser function	Data type returned	Description
TypeOf	Object (enumerated)	Returns the type of the object.

TraceUser object

The TraceUser object provides information about a node in a trace file identified as an occurrence of an activity you selected for logging, including the activity argument and message. To access the extra properties of the TraceUser object, you assign a TraceActivityNode object whose activity type is ActUser! to the TraceUser object.

The TraceUser object has no events.

Properties

TraceUser property	Data type	Description
ActivityType	TraceActivity (enumerated)	The value ActUser! which identifies the activity represented by the node as an occurrence of an activity you selected.
Argument	Long	The argument passed to the TraceUser function.
Category	TraceCategory (enumerated)	The value TraceAtomic! which indicates that the node is an activity that occurred in a single statement.
Message	String	The message passed to the TraceUser function.
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceUser function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.

TraceUser object

TraceUser function	Data type returned	Description
TypeOf	Object (enumerated)	Returns the type of the object.

Transaction object

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

You can customize your own version of the Transaction object by defining a class user object inherited from the built-in Transaction object.

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

FOR INFO For more information about using the Transaction object in an application, see *Application Techniques*.

Properties

Transaction property	Data type	Description
AutoCommit	Boolean	The automatic commit indicator (SQL Server only). Values are: <ul style="list-style-type: none"> ◆ True — Commit automatically after every database activity ◆ False — Do not commit automatically after every database activity
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.

Transaction property	Data type	Description
SQLCode	Long	The success or failure code of the most recent operation. Return codes: 0 — Success 100 — Not found -1 — Error (use SQLDBCode or SQLErrText to obtain the details)
SQLDBCode	Long	The database vendor's error code.
SQLErrText	String	The database vendor's error message.
SQLNRows	Long	The number of rows affected (the database vendor supplies this number, so the meaning may not be the same in every DBMS).
SQLReturnData	String	DBMS-specific information.
UserID	String	The name or ID of the user who will connect to the database.

Events

Transaction event	Occurs
Constructor	When the user object is created.
Destructor	When the user object is destroyed.

Functions

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

Transaction function	Data type returned	Description
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
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.

Transport

The Transport object contains the parameters that PowerBuilder uses to process client requests to connect to a server application. The Transport object controls all communications between the client and the server.

The Transport object has no events.

Local driver

The Local driver is *not* supported with the Transport object.

Properties

Transport property	Data type	Description
Application	String	<p>Identifies the server application.</p> <p>For the WinSock driver Specifies either of the following:</p> <ul style="list-style-type: none"> ◆ The port number for the application (for example, 10099). Each server application requires a unique port number on the server machine. If you specify a port number, select a number that is greater than 4096 ◆ The service name for the application. The service name is an indirect reference to the application's port number. The mapping of the service name to the port number is specified in the TCP/IP services file <p>For the NamedPipes driver Specifies the application portion of the pipe name.</p>
Driver	String	<p>The communications driver that will be used to listen for client connections. Values are:</p> <ul style="list-style-type: none"> ◆ WinSock ◆ NamedPipes

Transport property	Data type	Description
ErrCode	Long	Code indicating the success or failure of the most recent operation. Values are: <ul style="list-style-type: none"> ◆ 0 — Success ◆ 50 — Distributed service error ◆ 52 — Distributed communications error ◆ 78 — Listen only supported in Enterprise version of PowerBuilder
ErrText	String	Text indicating the success or failure of the most recent operation
Location	String	For the NamedPipes driver Optionally specifies a dot (.) to indicate that the server is the local machine. You do not need to specify a location for the server. If no location is specified, a local pipe name is constructed using a dot (.) in the machine name portion For the WinSock driver This property is ignored
Options	String	Specifies one or more communications options. If you specify more than one option, you need to separate the options with commas For a description of each of the options, see "Options" next.
Timeout	Long	Specifies the number of seconds a connection can remain idle before being disconnected.
Trace	String	Specifies one or more trace options. The trace options are used to troubleshoot client/server connections. If you specify more than one trace option, you need to separate the options with commas For a description of each of the trace options, see "Trace options" on page 394.

Options

Option	Description
BufSize= <i>n</i>	Sets the buffer size for the transport to the value specified.

Option	Description
MaxListeningThreads= <i>n</i>	Determines the maximum number of listening threads available on the server application On UNIX, you need to set MaxListeningThreads to 1.
NoDelay=1	Specifies that each packet be sent without delay. Corresponds to the TCP_NODELAY option. Setting this option may degrade performance significantly. Do not use this option unless you thoroughly understand its implications Applies to the WinSock driver only
RawData=1	Specifies that raw data be passed over the network. By default, the WinSock driver obscures the data that is passed over the network. Setting this option to 1 overrides the default behavior. Both the client and server applications must have the same setting. If there is a discrepancy between the client and server settings, the communication fails Setting this option to 1 may improve performance slightly Applies to the WinSock driver only

Trace options

Trace option	Description
All=1	Logs all activity against the server. This option logs internal information that can be helpful in debugging, including memory usage, an internal call trace, and the types and values for passed parameters
Console=1	Logs activity to a console window (not supported on Windows 3.1)
DumpArgs=1	Logs parameter types and values
Level=1	Enables Console=1, ObjectCalls=1, ObjectLife=1, and WebPB=1
Log= <i>filename</i>	Logs activity to the specified file. If the server application uses multiple transport objects, you need to specify a unique log file name for each transport object

Trace option	Description
ObjectCalls=1	Logs each object method call and indicates whether the call was successful
ObjectLife=1	Logs each attempt to create or destroy a remote object and indicates whether the operation was successful
WebPB=1	Logs all Web.PB activity against the server application

Functions

Transport function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
Listen	Long	Instructs a server application to begin listening for client connections
StopListening	Long	Instructs a server application to stop listening for client connections
TypeOf	Object	Returns the type of the object.

TreeView control

A TreeView control is a hierarchical display of information. Each item in a TreeView control consists of text and pictures, which can be manipulated during program runtime.

Properties

TreeView property	Data type	Description
Accelerator	Integer	Specifies the ASCII value of the accelerator key you want to for the control.
BackColor	Long	Specifies the numeric value of the background color: - 2 to 16,777,215
Border	Boolean	Specifies whether the control has a border. Values are: <ul style="list-style-type: none"> ◆ True — Control has a border ◆ False — Control does not have a border
BorderStyle	BorderStyle (enumerated)	Specifies the border style of the control. Values are: <ul style="list-style-type: none"> StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order. Values are: <ul style="list-style-type: none"> ◆ True — Control is on top of other controls ◆ False — Control is not on top of other controls

TreeView property	Data type	Description
DeleteItems	Boolean	<p>Specifies whether the user can delete a TreeView item from a TreeView control using the DELETE key. Values are:</p> <ul style="list-style-type: none">◆ True — The user can delete items from the control◆ False — The user cannot delete items from the control
DisableDragDrop	Boolean	<p>Disable Drag Drop determines whether events for dragging, such as BeginDrag, are triggered when the user clicks on an item and drags. Values are:</p> <ul style="list-style-type: none">◆ True — Drag events are not triggered◆ False — Drag events are triggered
DragAuto	Boolean	<p>Specifies whether PowerBuilder puts the control automatically into Drag mode. Values are:</p> <ul style="list-style-type: none">◆ 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>In either case, DisableDragDrop must be set to FALSE for dragging to occur.</p>

TreeView property	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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p>
EditLabels	Boolean	<p>Specifies whether the user can edit the item labels in a control by clicking on a selected item. Values are:</p> <ul style="list-style-type: none"> ◆ True — The user can edit item labels ◆ False — The user can not edit item labels
Enabled	Boolean	<p>Specifies whether the control is enabled (can be clicked). Values are:</p> <ul style="list-style-type: none"> ◆ True — Control can be clicked ◆ False — Control cannot be clicked
FaceName	String	<p>Specifies the name of the typeface in which the text of the control displays (for example, ARIAL or COURIER).</p>

TreeView property	Data type	Description
FontCharSet	FontCharSet (enumerated)	Specifies the font character set used for the text in the control. The application must be running on an appropriate version of PowerBuilder under an operating system that supports the selected character set. Values are: ANSI! ChineseBig5! DefaultCharSet! Hangeul! OEM! ShiftJIS! Symbol!
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) used for the text in the control. Values are: AnyFont! Decorative! Modern! Roman! Script! Swiss!
FontPitch	FontPitch (enumerated)	Specifies the font pitch for the text in the control. Values are: Default! Fixed! Variable!
HasButtons	Boolean	Specifies whether TreeView parent items will have + and - buttons associated with them to indicate if they are expanded (-) or collapsed (+). Values are: ◆ True — Parent items will have buttons ◆ False — Parent items will not have buttons

TreeView property	Data type	Description
HasLines	Boolean	<p>Specifies whether TreeView items will be connected by lines. Values are:</p> <ul style="list-style-type: none"> ◆ True — Items will be connected by lines ◆ False — Items will not be connected by lines
Height	Integer	<p>Specifies the height of the control, in PowerBuilder units.</p>
HideSelection	Boolean	<p>Specifies whether selected text stays selected (highlighted) even when the control does not have focus. Values are.</p> <ul style="list-style-type: none"> ◆ True — Text does not stay highlighted ◆ False — Text stays highlighted
Indent	Integer	<p>Specifies the size, in PowerBuilder units, that TreeView items will be indented. Negative values are accepted but the items are outdented beyond the left edge of the control.</p> <p>Room is always reserved for the regular picture, whether or not it is displayed. An indent of less than 90 has no effect for the standard picture width. Set PictureWidth to 0 to remove extra space.</p>
Italic	Boolean	<p>Specifies whether the text in the control is italic. Values are:</p> <ul style="list-style-type: none"> ◆ True — Text is italic ◆ False — Text is not italic
LinesAtRoot	Boolean	<p>Specifies whether PowerBuilder will connect TreeView root items with lines. Values are:</p> <ul style="list-style-type: none"> ◆ True — TreeView control connects root items with lines when HasLines is also true ◆ False — Control does not connect root items with lines

TreeView property	Data type	Description
PictureHeight	Integer	<p>Specifies the size, in pixels, for the height of the TreeView item picture.</p> <p>In a script, this value can only be set before a picture has been added to the picture index list.</p> <p>If the picture height is 0, PowerBuilder uses the height of the first picture added to the picture index list.</p>
PictureMaskColor	Long	<p>Specifies the color to be transparent when used in a TreeView item picture. Used when the picture is added at initialization or with the function AddPicture, and may be changed between adds.</p>
PictureName	String	<p>Specifies the name of the picture used in the TreeView. The picture can be an icon, cursor, or bitmap supplied by the user or a stock picture from the PowerBuilder library. Not updated after initialization.</p>
PictureWidth	Integer	<p>Specifies the size, in pixels, for the width of the TreeView item picture.</p> <p>In a script, this value can only be set before a picture has been added to the picture index list.</p> <p>If the picture width is 0, PowerBuilder uses the width of the first picture added to the picture index list.</p>
Pointer	String	<p>Specifies the name of the stock pointer or the file containing the pointer used for the control.</p>

TreeView property	Data type	Description
SortType	grSortType	<p>Selects the sort method. Values are:</p> <ul style="list-style-type: none"> ◆ Ascending! — Alphabetic by label ◆ Descending! — Reverse-alphabetic by label ◆ UserDefinedSort! — According to the script in the Sort event ◆ Unsorted! — Not sorted <p>When SortType specifies sorting, sorting happens automatically. For Unsorted!, you can call functions for alphabetic sorting.</p>
StatePictureHeight	Integer	<p>Specifies the size, in pixels, for the height of the state picture.</p> <p>In a script, this value can only be set before a state picture has been added to the state picture list.</p> <p>If the state picture height is 0, PowerBuilder uses the height of the first picture added to the state picture index list.</p>
StatePictureMask Color	Long	<p>Specifies the color to be transparent when used in a state picture. Used when the picture is added at initialization or with the function AddStatePicture, and may be changed between adds.</p>
StatePictureName	String	<p>Specifies the name of the picture used as the state picture. The state picture is displayed to the left of the regular picture. The item is shifted right to make room for it.</p> <p>The picture can be an icon, cursor, or bitmap supplied by the user or a stock picture from the PowerBuilder library. Not updated after initialization.</p>

TreeView property	Data type	Description
StatePictureWidth	Integer	<p>Specifies the size in pixels for the width of the state picture.</p> <p>In a script, this value can only be set before a state picture has been added to the state picture list.</p> <p>If the state picture width is 0, PowerBuilder uses the width of the first picture added to the state picture index list.</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>
TextColor	Long	<p>Specifies the numeric value of the color used for text: -2 to 16,777,215.</p>
TextSize	Integer	<p>Specifies the size of the text in the control, in points.</p> <p>For backward compatibility, the size is stored as a negative number; for example, 10-point text size is stored as -10.</p>
Underline	Boolean	<p>Specifies whether the text in the control is underlined. Values are:</p> <ul style="list-style-type: none"> ◆ True — Text is underlined ◆ False — Text is not underlined
Visible	Boolean	<p>Specifies whether the control is visible. Values are:</p> <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
Weight	Integer	<p>Specifies the stroke weight of the text in the control; for example, 400 for normal or 700 for bold.</p>
Width	Integer	<p>Specifies the width of the control, in PowerBuilder units.</p>
X	Integer	<p>Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.</p>

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

Events

TreeView event	Occurs
BeginDrag	When the user begins a drag operation with the left mouse button. If the DragAuto property is set to True, the drag will begin automatically. If the DragAuto property is set to False, the drag operation must be done programmatically.
BeginLabelEdit	When the user starts to edit a TreeView item label. Return 1 to prevent setting to the new text. Return 0 to accept the new text.
BeginRightDrag	When the user begins a drag operation with the right mouse button. If the DragAuto property is set to True, the drag will begin automatically. If the DragAuto property is set to False, the drag operation must be done programmatically.
Clicked	When the control is clicked.
Constructor	When the object is created, immediately before the Open event occurs in the window.
DeleteItem	When a TreeView item is deleted.
Destructor	When the object is destroyed, immediately after the Close event occurs in the window.
DoubleClicked	When the control is double-clicked.
DragDrop	When a dragged control is dropped on the TreeView control.
DragEnter	When a dragged control enters the control, including entering the narrow border around the display area.
DragLeave	When a dragged control leaves the control, including leaving by crossing into the tab page display area.
DragWithin	When a dragged control is within the control, but not on a TreeView item.

TreeView event	Occurs
EndLabelEdit	When the user finishes editing a TreeView item label. Return 1 to prevent setting to the new text. Return 0 to accept the new text.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
ItemCollapsed	When an item has collapsed.
ItemCollapsing	When an item is collapsing. Return 1 to prevent collapsing or 0 to allow it.
ItemExpanded	When an item has expanded.
ItemExpanding	When an item is expanding. Return 1 to prevent expansion or 0 to allow it. If you wish to populate the children each time an item expands, it should be done in the ItemExpanding event. If no children are created during the ItemPopulate or Item Expanding events, the item will not expand.
ItemPopulate	When an item is expanding for the first time. Return 1 to prevent expansion or 0 to allow it. If no children are created during the ItemPopulate or Item Expanding events, the item will not expand.
Key	When the user presses a key.
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PowerBuilder event.
RightClicked	When the control is right-clicked.
RightDoubleClicked	When the control is right double-clicked.
SelectionChanged	When the selection has changed.
SelectionChanging	When the selection is changing. Return 1 to prevent the selection from changing or 0 to allow it.
Sort	When sorting occurs and the SortType property is set to UserDefinedSort! The event occurs for each pair of items being sorted.

Functions

TreeView function	Data type returned	Description
AddPicture	Integer	Adds an icon, cursor, or bitmap to the image list. Does not update PictureName.
AddStatePicture	Integer	Adds an icon, cursor, or bitmap to the state image list. Does not update StatePictureName.
ClassName	String	Returns the name of the control.
CollapseItem	Integer	Collapses the specified TreeView item.
DeleteItem	Integer	Deletes the specified TreeView item and all its children, if any.
DeletePicture	Integer	Deletes the specified icon, cursor, or bitmap from the image list.
DeletePictures	Integer	Deletes all icons, cursors, or bitmaps from the image list.
DeleteStatePicture	Integer	Deletes the specified icon, cursor, or bitmap from the state image list.
DeleteStatePictures	Integer	Deletes all icons, cursors, or bitmaps from the state image list.
Drag	Integer	Starts or ends the dragging of a TreeView item.
EditLabel	Integer	Starts the editing of a specific TreeView item label.
ExpandAll	Integer	Expands the children and subsequent levels for the specified TreeView item.
ExpandItem	Integer	Expands the specified TreeView item.
FindItem	Long	Returns the handle for the specified TreeView item.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetItem	Integer	Retrieves information for a specified item.

TreeView function	Data type returned	Description
GetParent	PowerObject	Returns a reference to the name of the parent object.
Hide	Integer	Hides the specified TreeView item.
InsertItem	Long	Inserts a specified item at a specified position and level in a TreeView control.
InsertItemFirst	Long	Inserts a specified item as the first item at a specified level.
InsertItemLast	Long	Inserts a specified item as the last item at a specified level.
InsertItemSort	Long	Inserts a specified item at a specified level in the correct alphabetic position, if possible.
Move	Integer	Moves a control or object to a specified location.
PointerX	Integer	Determines the distance from the left edge of an object to the pointer location.
PointerY	Integer	Determines the distance from the top edge of an object to the pointer location.
PostEvent	Boolean	Adds the event to the end of the event queue of an object.
Print	Integer	Includes an object or lines of text in a print job.
Resize	Integer	Resizes a control to the specified dimensions.
SelectItem	Integer	Highlights an item in the control, making it the current item.
SetDropHighlight	Integer	Designates the specified TreeView item as the target of a DragDrop operation.

TreeView function	Data type returned	Description
SetFirstVisible	Integer	Sets the specified TreeView item as the first item visible in a TreeView control. If there are enough items to allow it, the specified item scrolls to the top of the control. If not, it is selected.
SetFocus	Integer	Sets the focus for a specified object or control.
SetItem	Integer	Sets the information for the specified TreeView item.
SetLevelPictures	Integer	Specifies a picture index for all TreeView items at a specific level in a TreeView control.
SetOverlayPicture	Integer	Maps a picture index to an overlay picture index. Only 4 overlay picture indexes are available.
SetPosition	Integer	Sets the position of the TreeView control in the front-to-back order within a window.
SetRedraw	Integer	Controls the automatic redraw of an object after its properties have changed.
Show	Integer	Makes an object or control visible, if it is hidden. If the object is already visible, Show brings it to the top.
Sort	Integer	Sorts the children of a specified TreeView item according to the method of the SortType property or alphabetically if SortType is Unsorted!.
SortAll	Integer	Sorts the children of an item and all subsequent levels according to the method of the SortType property or alphabetically if SortType is Unsorted!.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.

TreeViewItem object

A TreeViewItem is a system structure that populates the properties for individual items in a TreeView control. A TreeViewItem has no events.

Properties

TreeViewItem property	Data type	Description
Bold	Boolean	Specifies whether the item is bold. Values are: <ul style="list-style-type: none"> ◆ True — The item is bold ◆ False — The item is not bold
Children	Boolean	Specifies whether the item has children. Values are: <ul style="list-style-type: none"> ◆ True — The item has children ◆ False — The item does not have children <p>You can use this property to make the TreeView think it has children, even when it does not. You would want to do this in order to get ItemPopulate and ItemExpanding events when the item does not yet have any children.</p>
CutHighLighted	Boolean	Specifies whether the item is the target of a cut operation. Values are: <ul style="list-style-type: none"> ◆ True — The item is the target of a cut operation ◆ False — The item is not the target of a cut operation
Data	Any	Assigns any user-defined data to a TreeView item.

TreeViewItem property	Data type	Description
DropHighLighted	Boolean	Specifies whether the item is the target of a DragDrop operation. Values are: <ul style="list-style-type: none"> ◆ True — The item is the target of a DragDrop operation ◆ False — The item is not the target of a DragDrop operation
Expanded	Boolean	Specifies whether the item is expanded. Values are: <ul style="list-style-type: none"> ◆ True — The item is expanded ◆ False — The item is not expanded
ExpandedOnce	Boolean	Specifies whether the item has been expanded at least once, also meaning the item has been populated with children. Values are: <ul style="list-style-type: none"> ◆ True — The item has been expanded once ◆ False — The item has not been expanded once
HasFocus	Boolean	Specifies if the item has focus. Values are: <ul style="list-style-type: none"> ◆ True — The item has focus ◆ False — The item does not have focus
ItemHandle	Long	Identifies the handle associated with the item.
Label	Label	Identifies the string label associated with the item.
Level	Integer	Indicates the level of the item in the TreeView control.
OverlayPictureIndex	Integer	Identifies the overlay picture associated with the item. The overlay picture is display on top of the item's picture. If 0, no overlay is displayed.

TreeViewItem property	Data type	Description
PictureIndex	Integer	Identifies the picture displayed to the left of the item label. If 0, no picture appears and the space specified by the TreeView's PictureWidth property is blank.
SelectedPictureIndex	Integer	Identifies the picture associated with the item when it is selected. If 0, no picture is displayed when selected.
Selected	Boolean	Specifies whether the item is selected. Values are: <ul style="list-style-type: none"> ◆ True — The item is selected ◆ False — The item is not selected
StatePictureIndex	Integer	Identifies the state picture associated with the item. The state picture appears to the left of the regular picture. If 0, no state picture appears and no space is reserved for the picture.

Functions

TreeViewItem function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object	Returns the type of the object.

TypeDefinition object

Information about the type of a variable. The variable may be a single value, an object, or an array. TypeDefinition is an abstract class that is the ancestor of ClassDefinition, SimpleTypeDefinition, and EnumerationDefinition. It has no events.

TypeDefinition is used in the VariableDefinition class.

Properties

TypeDefinition property	Data type	Description
Category	TypeCategory	Specifies if the type is simple, enumerated, or a class or structure. Values are: SimpleType! EnumeratedType! ClassOrStructureType!
ClassDefinition	PowerObject	An object of type ClassDefinition describing the TypeDefinition class.
DataTypeOf	String	The system class name or simple data type of the variable. DataTypeOf is a string representation of a value of the Object enumerated data type. Values are lowercase with no exclamation point. Sample values include: window string any dropdownlistbox For objects you have defined, the data type is the system class from which your object is inherited.
IsStructure	Boolean	Indicates whether the type is a structure.
IsSystemType	Boolean	Indicates whether the type is defined by PowerBuilder as opposed to a type defined in a PBL by a user.

TypeDefinition property	Data type	Description
IsVariableLength	Boolean	Specifies whether the data type has a fixed size. Values are: <ul style="list-style-type: none"> ◆ True — The data type is variable length, meaning the data type is a string, any, blob, or unbounded array. ◆ False — The data type is a fixed length.
IsVisualType	Boolean	Indicates whether the type is a visual (displayable) or nonvisual type. Values are: <ul style="list-style-type: none"> ◆ True — The type is visual, for example, a window or a control ◆ False — The type is non-visual, for example, a class user object or a simple data type.
LibraryName	String	The fully qualified name of the library the type was loaded from. Note that the library may no longer contain the type. If a program manipulates the contents of libraries, its class could have been moved or deleted after it was loaded.
Name	String	The name of the type. For a nested type, the name will be returned in the form of <i>libraryEntryName`typeName</i>

Functions

TypeDefinition function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object	Returns the type of the object.

UserObject

UserObjects are custom visual objects that you can build to supplement the standard PowerBuilder objects. UserObjects can display information, request information from a user, and respond to mouse or keyboard actions. You can also create a TabPage UserObject. Use the User Object painter to build UserObjects.

When you place a visible UserObject in a window, you are actually placing a UserObject **control** in the window. The control holds an instance of the UserObject you select for the window.

Platform note

Functions for VBX user objects are only supported by Windows 3.1 and Windows 95.

Properties

UserObject property	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>PowerScript Reference</i> .
Border	Boolean	Specifies whether the object has a border. Values are: <ul style="list-style-type: none">◆ True — Object has a border◆ False — Object does not have a border
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the object. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!

UserObject property	Data type	Description
BringToTop	Boolean	Specifies whether PowerBuilder moves the object to the top of the front-to-back order of the window. Values are: <ul style="list-style-type: none"> ◆ True — Object moved to top ◆ False — Object not moved to top
ClassName	String	(External user objects only) Returns the name assigned to the object.
ColumnsPerPage	Integer	Specifies the number of columns on a scroll page. The default is 0 (10 columns per page). For information, see Scrolling in windows and user objects. PowerBuilder multiplies UnitsPerColumn by ColumnsPerPage to determine the number of PowerBuilder units to scroll the object horizontally when the user clicks in the scrollbar.
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 object automatically into Drag mode. Values are: <ul style="list-style-type: none"> ◆ True — When the object is clicked, it is automatically in Drag mode. ◆ False — When the object is clicked, it is not automatically in Drag mode. You have to manually put the object into Drag mode by using the Drag function.

UserObject property	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 object (the ICO file). The default icon is a box the size of the object.</p> <p>When the user drags the object, the icon displays when the object is over an area in which the object can be dropped (a valid drop area). When the object is over an area that is not a valid drop area, the No-Drop icon displays.</p> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p> <hr/>
Enabled	Boolean	<p>Specifies whether the object is enabled (can be selected). Values are:</p> <ul style="list-style-type: none"> ◆ True — Object can be selected ◆ False — Object cannot be selected
Height	Integer	<p>Specifies the height of the object, in PowerBuilder units.</p>
HScrollBar	Boolean	<p>Specifies whether a horizontal scrollbar displays. Values are:</p> <ul style="list-style-type: none"> ◆ True — Horizontal scrollbar displays ◆ False — Horizontal scrollbar does not display
LibraryName	String	<p>(External user objects only) The name of the dynamic-link library (DLL) that contains an external user object class.</p>
LinesPerPage	Integer	<p>Specifies the number of lines on a page. The default is 0 (10 lines per page). For information, see Scrolling in windows and user objects.</p> <p>PowerBuilder multiplies UnitsPerLine by LinesPerPage to determine the number of PowerBuilder units to scroll the object vertically when the user clicks in the scrollbar.</p>

UserObject property	Data type	Description
ObjectType	UserObjects (enumerated)	Specifies the type of user object. Values are: CustomVisual! ExternalVisual! VBXVisual!
PictureMaskColor	Long	Specifies the numeric value of the color in the picture that is changed to the background color. Applies only when PictureName is a bitmap and only when the UserObject is a tab page. Values can be: -2 to 16,777,215. For more information about color, see the <i>PowerScript Reference</i> .
PictureName	String	Specifies a value of the Pointer enumerated data type or the filename of the bitmap, cursor, or icon or to be displayed on the tab. This property applies only when the UserObject is a tab page.
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the object.
PowerTipText	String	Specifies text to be displayed as a PowerTip for the tab when the Tab control's PowerTips property is True. This property applies only when the UserObject is a tab page.
Style	Long	Specifies any additional style bits that you want to use to control how the object displays (external user object only).
TabBackColor	Long	Specifies the numeric value of the tab background color: -2 to 16,777,215. This property applies only when the UserObject is a tab page. For more information about color, see the <i>PowerScript Reference</i> .

UserObject property	Data type	Description
TabTextColor	Long	<p>Specifies the numeric value of the tab text color: -2 to 16,777,215.</p> <p>This property applies only when the UserObject is a tab page.</p> <p>For more information about color, see the <i>PowerScript Reference</i>.</p>
TabOrder	Integer	<p>Specifies tab value of the control within the user object (0 means the user cannot tab to the control).</p>
Tag	String	<p>Specifies the tag value assigned to the object.</p>
Text	String	<p>Specifies the text that displays in the object.</p>
UnitsPerColumn	Integer	<p>Specifies the number of PowerBuilder units to be scrolled right or left when a user clicks the left or right arrow in the horizontal scrollbar in a window or user object. The default is 0 (1/100 of the width of the window).</p> <p>To make the end of the scrollbar match the content, UnitsPerLine must be set according to the content width. For information, see <i>Scrolling in windows and user objects</i>.</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 scrollbar.</p>

UserObject property	Data type	Description
UnitsPerLine	Integer	<p>Specifies the number of PowerBuilder units to be scrolled up or down when a user clicks the up or down arrow in the vertical scrollbar in a window or user object. The default is 0 (1/100 of the window height).</p> <p>To make the end of the scrollbar match the content, UnitsPerLine must be set according to the content length. For information, see Scrolling in windows and user objects.</p> <p>PowerBuilder multiplies UnitsPerLine by LinesPerPage to determine the number of PowerBuilder units to scroll the window vertically when the user clicks in the scrollbar.</p>
Visible	Boolean	<p>Specifies whether the object is visible. Values are:</p> <ul style="list-style-type: none"> ◆ True — Object is visible ◆ False — Object is not visible
VScrollBar	Boolean	<p>Specifies whether a vertical scrollbar displays. Values are:</p> <ul style="list-style-type: none"> ◆ True — Vertical scrollbar displays ◆ False — Vertical scrollbar does not display
Width	Integer	Specifies the width of the object, in PowerBuilder units.
X	Integer	Specifies the X position (distance from the left edge of screen) of the object, in PowerBuilder units.
Y	Integer	Specifies the Y position (distance from the top of screen) of the object, in PowerBuilder units.

Events

UserObject 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 object.
DragEnter	When a dragged control enters the object.
DragLeave	When a dragged control leaves the object.
DragWithin	When a dragged control is within the object.
Other	When a Windows message occurs that is not a PowerBuilder event.
RButtonDown	When the right mouse button is pressed on the object.

Functions

UserObject 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 object.
DeleteItem	Integer	(VBX user objects supporting RemoveItem method only) Deletes item from list.
Drag	Integer	Starts or ends the dragging of the object.
EventParmDouble	Integer	Stores the specified numeric event parameter in a double variable.
		<p>Platform note Functions for VBX user objects are only supported by Windows 3.1 and Windows 95.</p>

UserObject function	Data type returned	Description
EventParmString	Integer	Stores the specified string event parameter in a string variable. Platform note Functions for VBX user objects are only supported by Windows 3.1 and Windows 95.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
Hide	Integer	Makes the object invisible.
InsertItem	Integer	(VBX user objects supporting AddItem method only) Inserts item in list.
Move	Integer	Places the object 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 object.
Print	Integer	Prints the object.
Resize	Integer	Changes the size of the object based on the width and height.
SetFocus	Integer	Sets focus to the object.
SetPosition	Integer	Specifies the position of the object in the front-to-back order of the window.
SetRedraw	Integer	Turns on or off automatic redrawing of the object after every change.
Show	Integer	Makes the object visible.

UserObject function	Data type returned	Description
TriggerEvent	Integer	Sends an event to the object and executes the script associated with the event.
TypeOf	Object	Returns the type of the object.

VariableCardinalityDefinition object

A class that provides information about the cardinality of a variable. It reports whether the associated variable is a single instance or an array. If it is an array, you can get information about the dimensions. VariableCardinalityDefinition is used in the VariableDefinition object. It has no events.

Properties

Variable Cardinality Definition property	Data type	Description
ArrayDefinition []	ArrayBounds	When the associated variable's Cardinality is BoundedArray!, an array with an ArrayBounds object for each dimension in the array being described. When Cardinality is UnboundedArray!, ArrayDefinition has a single ArrayBounds object with LowerBound and UpperBound properties both set to 0. The extent of the array is not part of the class definition. Not valid when Cardinality is ScalarType!
Cardinality	Variable CardinalityType	The cardinality of the associated variable. Values are: ScalarType! UnboundedArray! BoundedArray!
ClassDefinition	PowerObject	An object of type ClassDefinition describing the VariableCardinalityDefinition class.

Functions

Variable Cardinality Definition function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object	Returns the type of the object.

VariableDefinition object

A class describing the characteristics of a variable, property, or argument. VariableDefinition is used as a property in the ClassDefinition and ScriptDefinition objects. It has no events.

You can't start with a variable in your application and get a VariableDefinition object for it. Instead, you access the VariableDefinition instances that are elements of the VariableList array of a ClassDefinition instance or the ArgumentList array of a ScriptDefinition instance.

The VariableDefinition object has information about:

- ◆ The variable's name and type
- ◆ Whether the variable is a scalar or an array and information about the array
- ◆ The variable's initial value, whether the value overrides an ancestor's value, and whether the variable is a constant
- ◆ The read and write access levels for the variable
- ◆ The scope of the variable (global, shared, instance, local, argument), including whether the variable is an argument and how the argument is passed

Properties

Variable Definition property	Data type	Description
CallingConvention	ArgCalling Convention	The way an argument is passed when Kind is VariableArgument! Values are: ByReferenceArgument! ByValueArgument! ReadOnlyArgument! VarListArgument! VarListArgument! only applies to arguments for built-in PowerBuilder functions. They are shown as ellipses in the browser. For an example, see the ImportString function for DataWindow controls.

Variable Definition property	Data type	Description
Cardinality	Variable Cardinality Definition	Cardinality information for the variable.
ClassDefinition	PowerObject	An object of type ClassDefinition describing the VariableDefinition class.
InitialValue	Any	The initial value of the variable. Not valid when Kind is VariableArgument!
IsConstant	Boolean	Indicates whether the variable is a constant. Not valid when Kind is VariableArgument!
IsControl	Boolean	Indicates whether the variable is a control defined as a nested class within its parent, rather than an instance variable with a control class as its data type. Valid only when Kind is VariableInstance!
IsUserDefined	Boolean	Indicates whether the variable is a user-defined variable, instead of a property or variable defined by PowerBuilder. Always true for local variables. True for arguments if the function was also user-defined.
Kind	VariableKind	The scope of the variable. Values are: VariableGlobal! VariableShared! VariableInstance! VariableArgument! VariableLocal! Global variables are found only in the Application object. Argument and local variables are found only in scripts.
Name	String	The name of the variable.
OverridesAncestor Value	Boolean	Indicates if the current initial value overrides an ancestor's initial value. Valid only when Kind is VariableInstance!

Variable Definition property	Data type	Description
ReadAccess	VarAccess	The read access to the variable. Values are: Private! Public! Protected! System! Not valid when Kind is VariableArgument! or VariableLocal!
TypeInfo	TypeDefinition	Type information for the variable.
WriteAccess	VarAccess	The write access to the variable. Values are: Private! Public! Protected! System! Not valid for VariableArgument! or VariableLocal!

Functions

Variable Definition function	Data type returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
TypeOf	Object	Returns the type of the object.

VScrollBar control

A VScrollBar is a vertical bar with arrows at either end and a scroll box. Typically, you would use a VScrollBar control as a slider control for users to specify a value on a continuous scale, or as a way to graphically display information to the user.

Usage note

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

Properties

VScrollBar property	Data type	Description
BringToTop	Boolean	Specifies whether PowerBuilder moves the control to the top of the front-to-back order of the window. Values are: <ul style="list-style-type: none">◆ True — Control moved to top◆ False — Control not moved to top
DragAuto	Boolean	Specifies whether PowerBuilder puts the control automatically into Drag mode. Values are: <ul style="list-style-type: none">◆ 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.

VScrollBar property	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> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p>
Height	Integer	Specifies the height of the control, in PowerBuilder units.
MaxPosition	Integer	Specifies the value of the Position property when the scroll box is at the bottom of the scrollbar.
MinPosition	Integer	Specifies the value of the Position property when the scroll box is at the top of the scrollbar.
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 scrollbar width is used for the VScrollBar. Values are:</p> <ul style="list-style-type: none"> ◆ True — Standard width used ◆ False — Standard width not enforced 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.

VScrollBar property	Data type	Description
Visible	Boolean	Specifies whether the control is visible. Values are: <ul style="list-style-type: none"> ◆ True — Control is visible ◆ False — Control is not visible
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

VScrollBar 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 property 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.

VScrollBar event	Occurs
PageUp	When the open space above the scroll box is clicked.
RButtonDown	When the right mouse button is pressed on the control.

Functions

VScrollBar function	Data type returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetParent	PowerObject	Returns a reference to the name of the parent object.
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 properties.
Show	Integer	Makes the control visible.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.

VScrollBar function	Data type returned	Description
TypeOf	Object	Returns the type of the control.

Window object

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 properties, events, and functions. The properties 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.

Properties

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

Window property	Data type	Description
BackColor	Long	Specifies the numerical value of the background color of the window. Values are -2 to 16,777,215. For more information about color, see the <i>PowerScript Reference</i> .
Border	Boolean	Specifies whether the window has a border. Values are: ◆ True — Has a border ◆ False — Does not have a border
BringToTop	Boolean	Specifies whether PowerBuilder will move the window to the top of the front-to-back order. Values are: ◆ True — Will move to the top ◆ False — Will not move to the top

Window property	Data type	Description
ColumnsPerPage	Integer	<p>Specifies the number of columns on a page. The default is 0 (10 columns per page). For information, see <i>Scrolling in windows and user objects</i>.</p> <p>PowerBuilder multiplies <i>UnitsPerColumn</i> by <i>ColumnsPerPage</i> to determine the number of PowerBuilder units to scroll the window horizontally when the user clicks in the scrollbar.</p>
Control[]	WindowObject	<p>Contains the controls in the window. You should not change the contents of this array in a script.</p>
ControlMenu	Boolean	<p>Specifies whether the Control Menu box displays in the title bar. Values are:</p> <ul style="list-style-type: none"> ◆ True — Displays in title bar ◆ False — Does not display in title bar
Enabled	Boolean	<p>Specifies whether the window is enabled (can send and receive messages). Values are:</p> <ul style="list-style-type: none"> ◆ True — Can send/receive messages ◆ False — Cannot send/receive messages
Height	Integer	<p>Specifies the height of the window, in PowerBuilder units.</p> <p>You cannot resize minimized or maximized windows at runtime.</p>
HScrollBar	Boolean	<p>Specifies whether a horizontal scrollbar displays in the window. Values are:</p> <ul style="list-style-type: none"> ◆ True — A scrollbar displays ◆ False — A scrollbar does not display

Window property	Data type	Description
Icon	String	<p>Specifies a stock icon or the name of the ICO file containing the icon that will be displayed when the window is minimized.</p> <hr/> <p>Platform note ICO files are only supported on the Windows and Macintosh platforms.</p>
KeyboardIcon	Boolean	<p>This property is meaningful only for right-to-left versions (Hebrew and Arabic) of PowerBuilder.</p> <p>Specifies whether the keyboard icon appears on the window. The keyboard icon allows the user to change languages. Values are:</p> <ul style="list-style-type: none"> ◆ True — Keyboard icon appears on window ◆ False — Keyboard icon does not appear on window
LinesPerPage	Integer	<p>Specifies the number of lines on a page. The default is 0 (10 lines per page). For information, see Scrolling in windows and user objects.</p> <p>PowerBuilder multiplies UnitsPerLine by LinesPerPage to determine the number of PowerBuilder units to scroll the window vertically when the user clicks in the scrollbar.</p>
MaxBox	Boolean	<p>Specifies whether a Maximize Box displays in the title bar. Values are:</p> <ul style="list-style-type: none"> ◆ True — Maximize Box displays ◆ False — Maximize Box does not display

Window property	Data type	Description
MenuID	Menu	Specifies the ID of a menu. 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 Menu object.
MenuName	String	Specifies the name of a menu. 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 Menu object.
MinBox	Boolean	Specifies whether a Minimize Box displays in the title bar. Values are: <ul style="list-style-type: none"> ◆ True — Minimize Box displays ◆ False — Minimize Box does not display
Pointer	String	Specifies the name of the file containing the pointer that is used for the window.
Resizable	Boolean	Specifies whether the window is resizable. Values are: <ul style="list-style-type: none"> ◆ True — Window is resizable ◆ False — Window is not resizable

Window property	Data type	Description
RightToLeft	Boolean	Specifies that characters should be displayed in right-to-left order. The application must be running on a Hebrew or Arabic version of PowerBuilder under an operating system that supports right-to-left display. Values are: <ul style="list-style-type: none"> ◆ True — Characters display in right-to-left order ◆ False — Characters display in left-to-right order
Tag	String	Specifies the tag value assigned to the window.
Title	String	Specifies the text of the window title.
TitleBar	Boolean	Specifies whether a title bar displays. Values are: <ul style="list-style-type: none"> ◆ True — Title bar displays ◆ False — No title bar displays The user can move a window only if it has a title bar.
ToolBarAlignment	ToolBar Alignment (enumerated)	In an MDI frame window, specifies where the toolbar displays. Values are: <ul style="list-style-type: none"> AlignAtBottom! AlignAtLeft! AlignAtRight! AlignAtTop! Floating!
ToolBarHeight	Integer	In an MDI frame window, specifies the height of the toolbar when it is a floating toolbar.
ToolBarVisible	Boolean	In an MDI frame window, specifies whether the toolbar displays. Values are: <ul style="list-style-type: none"> ◆ True — Toolbar displays ◆ False — Toolbar does not display
ToolBarWidth	Integer	In an MDI frame window, specifies the width of the toolbar when it is a floating toolbar.

Window property	Data type	Description
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.
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 to be scrolled right or left when a user clicks the left or right arrow in the horizontal scrollbar in a window or user object. The default is 0 (1/100 of the width of the window).</p> <p>To make the end of the scrollbar match the content, UnitsPerLine must be set according to the content width. For information, see <i>Scrolling in windows and user objects</i>.</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 scrollbar.</p>

Window property	Data type	Description
UnitsPerLine	Integer	<p>Specifies the number of PowerBuilder units to be scrolled up or down when a user clicks the up or down arrow in the vertical scrollbar in a window or user object. The default is 0 (1/100 of the window height).</p> <p>To make the end of the scrollbar match the content, UnitsPerLine must be set according to the content length. For information, see <i>Scrolling in windows and user objects</i>.</p> <p>PowerBuilder multiplies UnitsPerLine by LinesPerPage to determine the number of PowerBuilder units to scroll the window vertically when the user clicks in the scrollbar.</p>
Visible	Boolean	<p>Specifies whether the window is visible. Values are:</p> <ul style="list-style-type: none"> ◆ True — Window is visible ◆ False — Window is not visible
VScrollBar	Boolean	<p>Specifies whether a vertical scrollbar displays. Values are:</p> <ul style="list-style-type: none"> ◆ True — Scrollbar displays ◆ False — Scrollbar does not display
Width	Integer	<p>Specifies the width of the window, in PowerBuilder units.</p> <p>You cannot resize minimized or maximized windows at runtime.</p>
WindowState	WindowState (enumerated)	<p>Specifies the state in which you want to run a window. Values are:</p> <p>Maximized! Minimized! Normal!</p> <p>Do not change the WindowState property in the Open event of a window opened as a sheet.</p>

Window property	Data type	Description
WindowType	WindowType (enumerated)	<p>Specifies the type of window. Values are:</p> <p>Child! Main! MDI! MDIHelp! Popup! Response!</p>
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 scrollbar, 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 scrollbar, if any).</p> <p>You cannot move a maximized window at runtime.</p>

Window property	Data type	Description
Y	Integer	<p>Specifies the Y position (distance from the top of the screen) of the window, in PowerBuilder units.</p> <p>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>The workspace is the area between the sides of the window (not including the thickness of the frame, toolbar, or scrollbar, 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 scrollbar, if any).</p> <p>You cannot move a maximized window at runtime.</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.

Platform note

DDE functions are available on the Windows platform. On UNIX platforms this and other DDE functions have effect only if the server and client applications are developed using PowerBuilder or compiled using Wind/U from Bristol Technology.

Window event	Occurs
Activate	<p>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.</p> <hr/> <p>Platform note This and other OLE functions are only available on the Windows platform.</p>
Clicked	<p>When the user clicks in an unoccupied area of the window (any area with no visible, enabled object).</p>
Close	<p>When the window is closed.</p>
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>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	<p>When the window becomes inactive.</p>
DoubleClickd	<p>When the user double-clicks in an unoccupied area of the window (any area with no visible, enabled object).</p>
DragDrop	<p>When a dragged control is dropped on the window.</p>
DragEnter	<p>When a dragged control enters the window.</p>
DragLeave	<p>When a dragged control leaves the window.</p>
DragWithin	<p>When a dragged control is within the window.</p>
Hide	<p>Just before the window is hidden.</p>
HotLinkAlarm	<p>After a Dynamic Data Exchange (DDE) server application has sent new (changed) data and the client DDE application has received it.</p> <p>See the platform note on page 441.</p>

Window event	Occurs
Key	When the user presses a key and the insertion point is not in a RichTextEdit or DataWindow edit control.
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.
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. See the platform note on page 441.
RemoteHotLinkStart	When a DDE client application wants to start a hot link. See the platform note on page 441.
RemoteHotLinkStop	When a DDE client application wants to end a hot link. See the platform note on page 441.
RemoteRequest	When a DDE client application requests data. See the platform note on page 441.
RemoteSend	When a DDE client application has sent data. See the platform note on page 441.
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 ALT or ALT plus another key.

Window event	Occurs
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.

PowerScript system functions

You can also use the PowerScript system functions in scripts for a window. For a list of the PowerScript system functions, see the [Browser](#).

Opening and closing a window

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

Window 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 the specified user object from view, closes it, and executes its Destructor event.
GetActiveSheet	Window	Returns the currently active sheet in the specified MDI frame window.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.

Window function	Data type returned	Description
GetFirstSheet	Window	Returns the top sheet in the MDI frame.
GetNextSheet	Window	Returns the sheet that is behind the specified sheet in the MDI frame.
GetParent	PowerObject	Returns a reference to the name of the parent object.
GetToolbar	Integer	Gets the values of the Visible, Alignment, and Title properties of the toolbar.
GetToolbarPos	Integer	Gets position information (coordinates) for a floating toolbar.
Hide	Integer	Makes the window invisible.
Move	Integer	Places the window in a new location specified by the X and Y arguments. The Move function does not move a maximized window.
OpenUserObject	Integer	Displays the specified user object, making its properties available to scripts.
OpenUserObjectWithParm	Integer	Displays the specified user object, making its properties available to scripts, and stores a parameter in the system's Message object.
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.

Window function	Data type returned	Description
Resize	Integer	Changes the size of the window to the size specified in the width and height arguments. The Resize function does not resize a minimized or maximized window.
SetFocus	Integer	Sets 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.
SetToolbar	Integer	Sets the values of the Visible, Alignment, and Title properties of the toolbar.
SetToolbarPosition	Integer	Sets the position of a fixed toolbar.
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.
WorkSpaceHeight	Integer	Returns the height of the workspace of the specified window. The workspace is the area between the sides of the window (not including the thickness of the frame, toolbar, or scrollbar, 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 scrollbar, if any).

Window function	Data type returned	Description
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.