

# PowerBuilder Data Types

D

PowerBuilder data types are broken down into two distinct types: standard data types and enumerated data types.

## Data Types

The standard data types are those that are used in most programming languages and include Char, Integer, Decimal, Long, and String (see Table D.1).

**Table D.1. PowerBuilder standard data types.**

<i>Data Type</i>	<i>Description</i>
Any	An undeclared data type that is only available in PowerBuilder 4.0 and above. You have to use the <code>ClassName()</code> function to determine the actual data type.
Blob	Binary large object.
Boolean	A Boolean value ( <code>TRUE</code> or <code>FALSE</code> ).
Character	A single string character. Often abbreviated to Char.
Date	Dates in the <code>yyyy-mm-dd</code> format. Hyphens and leading zeros are required and blanks are not allowed.
DateTime	Date and time combined into a single value.
Decimal	A signed decimal number with 18 digits of precision. Often abbreviated to Dec.
Double	A signed floating-point number with 15 digits of precision. The actual range varies from one platform to another.
Integer	A 16-bit signed whole number between <code>-32,768</code> and <code>+32,767</code> . Sometimes abbreviated Int.
Long	A 32-bit signed whole number between <code>-2,147,483,648</code> and <code>+2,147,483,647</code> .
Real	A signed floating-point number with 6 digits of precision. The actual range varies from one platform to another.
String	A string of 0 to 59,999 ASCII characters (or in the 32-bit environments, as large as you have virtual memory to hold).
Time	A time in 24-hour format: <code>hh:mm:ss</code> . Colons and leading zeros are required and blanks are not allowed.
UnsignedInteger	A 16-bit unsigned whole number between 0 and 65,535. Often abbreviated <code>UnsignedInt</code> or <code>UInt</code> .
UnsignedLong	A 32-bit unsigned whole number between 0 and 4,294,967,295. Often abbreviated <code>ULong</code> .

# PowerBuilder Enumerated Data Types

Enumerated data types have predefined sets of values. These data types can be used either as arguments in function calls or in the assignment of a value to an object attribute. The values of an enumerated data type always end with an exclamation point (!). Table D.2 is by no means an exhaustive list, but includes the commonly used ones. To see the full list, open the Object Browser in the Library painter and select the Enumerated Data Types radio button.

**Table D.2. Enumerated data types.**

<i>Enumerated Type</i>	<i>Description</i>	<i>Values</i>
Alignment!	Alignment of text.	Center! Justify! Left! Right!
ArrangeOpen!	The way a sheet will be arranged in an MDI frame when it is opened.	Cascaded! Layered! Original!
ArrangeType!	The way open sheets in an MDI frame will be arranged.	Cascade! Icons! Layer! Tile! TileHorizontal!
Border!	The style of the border that will be used for a DataWindow column.	Box! Lowered! NoBorder! Raised! ResizeBorder! ShadowBox! Underline!
Button!	Button that displays in a message box.	AbortRetryIgnore! Ok! OkCancel! RetryCancel! YesNo! YesNoCancel!

*continues*

Table D.2. continued

<i>Enumerated Type</i>	<i>Description</i>	<i>Values</i>
ConvertType!	How you want to convert units or pixels in the PixelsToUnits() and UnitsToPixels() functions.	XPixelsToUnits! YPixelsToUnits! XUnitsToPixels! YUnitsToPixels!
dwBuffer!	The DataWindow buffer.	Delete! Filter! Primary!
dwItemStatus!	The status of an item in a DataWindow.	DataModified! New! NewModified! NotModified!
FileLock!	The file locking mode, when opening a file.	LockRead! LockReadWrite! LockWrite! Shared!
FileMode!	A mode for reading and writing a file.	LineMode! StreamMode!
FillPattern!	The hatch pattern used to fill a drawing object.	bDiagonal! Diamond! fDiagonal! Horizontal! Solid! Square! Vertical!
FontCharSet!	Font character set.	ANSI! ChineseBig5! DefaultCharset! Hangeul! OEM! Shiftjis! Symbol!
FontFamily!	Font family.	AnyFont! Decorative! Modern! Roman! Script! Swiss!

<i>Enumerated Type</i>	<i>Description</i>	<i>Values</i>
FontPitch!	Font pitch.	Default! Fixed! Variable!
HelpCommand!	Type of command for the ShowHelp() function.	Index! Keyword! Topic!
Icon!	The icon for a message box.	Exclamation! Information! Question! StopSign!
KeyCode!	A key in the KeyDown() function. See the online help for the KeyDown() function for a listing of the valid codes.	
LibDirType!	The type of objects to be included in the directory list.	DirAll! DirApplication! DirDataWindow! DirFunction! DirMenu! DirPipeline! DirProject! DirQuery! DirStructure! DirUserObject! DirWindow!
LibExportType!	The type of object to be exported.	ExportApplication! ExportDataWindow! ExportFunction! ExportMenu! ExportPipeline! ExportProject! ExportQuery! ExportStructure! ExportUserObject! ExportWindow!

*continues*

Table D.2. continued

<i>Enumerated Type</i>	<i>Description</i>	<i>Values</i>
LibImportType!	The type of object to be imported.	ImportDataWindow!
ParmType!	Data type of input and output parameters.	TypeBoolean! TypeDate! TypeDateTime! TypeDecimal! TypeDouble! TypeInteger! TypeLong! TypeReal! TypeString! TypeTime! TypeUInt! TypeULong! TypeUnknown!
Pointer!	The pointer for the SetPointer() function.	Arrow! Beam! Cross! HourGlass! SizeNESW! SizeNS! SizeNWSE! SizeWE! UpArrow!
RowFocusInd!	The method that will be used to indicate that a row has focus.	FocusRect! Hand! Off!
SaveAsType!	The type of file to create when exporting the rows of a DataWindow.	Clipboard! CSV! dBase2! dBase3! DIF! Excel! PSReport! SQLInsert! Syk! Text!

<i>Enumerated Type</i>	<i>Description</i>	<i>Values</i>
		Wk1! Wks! WMF!
SeekType!	The position at which you want to begin a FileSeek().	FromBeginning! FromCurrent! FromEnd!
TextCase!	Text case.	AnyCase! Lower! Upper!
TrigEvent!	Type of event to be triggered by the Trigger function. (There is a long list for this enumerated type and you should refer to the Object Browser.)	
VTextAlign!	The alignment of text.	Bottom! MultiLine! Top! VCenter!
WindowState!	State in which a window will open.	Maximized! Minimized! Normal!
WindowType!	Type of window.	Child! Main! MDI! MDIHelp! Popup! Response!
WriteMode!	The mode for the FileOpen() function.	Append! Replace!

