

SYBASE®

Objects and Controls

PocketBuilder™

2.0.1

DOCUMENT ID: DC00133-01-0201-01

LAST REVISED: February 2005

Copyright © 2003-2005 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. 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.

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

11/04

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

About This Book	xv
 CHAPTER 1	 PocketBuilder System Objects and Controls..... 1
	What are system objects? 1
	System object properties, events, and functions..... 2
	System object inheritance hierarchy 2
	Base class objects not in this book 3
	Viewing system objects 4
 CHAPTER 2	 System Object Properties, Events, and Functions 5
	ADOResultSet object 6
	Application object 6
	ArrayBounds object 10
	BarcodeScanner base class..... 11
	BiometricScanner base class 13
	CallLog object..... 14
	CallLogEntry object 15
	Camera object 16
	CameraImageAttributes object..... 18
	CheckBox control 19
	ClassDefinition object..... 24
	CommandButton control..... 26
	Connection object..... 31
	ContextInformation object 31
	ContextKeyword object 33
	CORBACurrent object 34
	CORBAObject object..... 34
	DataStore object..... 35
	DataWindow control 44
	DataWindowChild object 60
	DialingDirectory object 66
	DialingDirectoryEntry object 67
	DropDownListBox control..... 69

DropDownPictureListBox control.....	76
DynamicDescriptionArea object	76
DynamicStagingArea object	76
EditMask control.....	77
EnumerationDefinition object	84
EnumerationItemDefinition object	86
Environment object.....	87
Error object.....	89
ErrorLogging object.....	91
Exception object.....	91
FileDirect	92
GPSCoordinate object.....	93
GPSFix object	94
GPSHeading object.....	95
GPSSatellitePosition object.....	96
GPSSatellitesInView object.....	96
Graph object.....	97
grAxis object.....	103
grDispAttr object.....	107
GroupBox control	109
HPBiometricScanner object	113
HProgressBar control	114
HScrollBar control	117
HTrackBar control	120
Inet object.....	124
InternetResult object	125
JaguarORB object	126
Line control.....	127
ListBox control.....	128
ListView control	135
ListViewItem object	145
mailFileDescription object	147
mailMessage object.....	148
mailRecipient object	149
mailSession object	150
MDIClient object.....	152
Menu object.....	152
MenuCascade object.....	157
Message object	157
MultiLineEdit control	159
NotificationBubble object.....	165
OLEControl control	167
OLECustomControl control (OCX)	167
OLEObject object	167

OLEStorage object	167
OLEStream object	168
OLETxnObject object	168
Oval control	168
PhoneCall object	170
Picture control	173
PictureButton control	177
PictureHyperLink control	182
PictureListBox control	186
Pipeline object	186
POOM object	187
POOMAppointment object	189
POOMContact object	191
POOMRecipient object	194
POOMRecurrence object	194
POOMTask object	196
ProfileCall object	199
ProfileClass object	200
ProfileLine object	201
ProfileRoutine object	203
Profiling object	205
RadioButton control	207
Rectangle control	211
ResultSet object	214
ResultSets object	214
RichTextEdit control	214
RoundRectangle control	215
RuntimeError object	217
ScriptDefinition object	219
SerialGPS object	222
Signature control	223
SimpleTypeDefinition object	226
SingleLineEdit control	227
SMSAddress object	232
SMSMessage object	233
SMSProtocol object	234
SMSProviderSpecificData object	235
SMSSession object	236
SocketBarcodeScanner object	237
SSLCallBack object	239
SSLServiceProvider object	239
StaticHyperLink control	240
StaticText control	245
SymbolBarcodeScanner object	249

Tab control	252
Throwable object	259
Timing object	260
Toolbar control	260
ToolStripItem object	262
TraceActivityNode object	264
TraceBeginEnd object	265
TraceError object	266
TraceESQL object	267
TraceFile object	269
TraceGarbageCollect object	270
TraceLine object	272
TraceObject object	273
TraceRoutine object	274
TraceTree object	276
TraceTreeError object	277
TraceTreeESQL object	278
TraceTreeGarbageCollect object	280
TraceTreeLine object	281
TraceTreeNode object	282
TraceTreeObject object	284
TraceTreeRoutine object	285
TraceTreeUser object	286
TraceUser object	288
Transaction object	289
TransactionServer object	291
TreeView control	291
TreeViewItem object	301
TypeDefinition object	303
UserObject object	305
VariableCardinalityDefinition object	310
VariableDefinition object	312
VProgressBar control	314
VScrollBar control	318
VTrackBar control	321
Window object	324

CHAPTER 3	Property Descriptions and Usage	337
	Accelerator	337
	Activation	338
	Address	338
	AddressType	339
	Alignment	340
	AllowEdit	340

Altitude	341
AutoArrange	342
AutoHScroll	342
Automatic	343
AutoScale	344
AutoSize	344
AutoSkip	345
AutoVScroll.....	346
Azimuth	347
BackColor	347
BeginX	349
BeginY	349
Body	350
BoldSelectedText	351
Border.....	352
BorderColor	353
BorderStyle.....	354
BottomMargin	354
BringToTop.....	355
ButtonHeader	355
CallState	356
CameraName	357
CameraType.....	357
Cancel	358
Caption	359
Category	359
CategorySort	360
Center.....	361
Checked	361
Close	362
ColumnsPerPage	363
ConfigParams.....	364
ContentsAllowed	366
ControlMenu	366
CornerHeight	366
CornerWidth	367
CreateOnDemand	368
DataObject	368
DataSource	369
DataType.....	370
Default	371
DefaultSize	372
Degree.....	373
Deleteltems	373

Depth.....	374
DeviceHandle.....	375
DisabledName.....	375
DisableDragDrop.....	376
DisableNoScroll.....	377
DisplayEveryNLabels.....	377
DisplayExpression.....	378
DisplayName.....	379
DisplayOnly.....	379
DisplayType.....	380
DocumentName.....	380
DragAuto.....	381
DragIcon.....	381
DriverName.....	382
DriverVersion.....	382
DropLines.....	383
Dropped.....	384
Duration.....	384
EditLabels.....	385
Elevation.....	386
Enabled.....	387
Encoding.....	387
EndTime.....	388
EndX.....	389
EndY.....	389
Escapement.....	390
ExtendedSelect.....	391
FaceName.....	392
FillColor.....	393
FillPattern.....	393
FixedLocations.....	394
FixedWidth.....	395
FixQuality.....	396
FixTime.....	397
Flags.....	397
FocusOnButtonDown.....	399
FocusRectangle.....	399
Folder.....	400
FontCharSet.....	401
FontFamily.....	402
FontPitch.....	402
Format.....	403
Frame.....	404
FreeDBLibraries.....	405

GeoidalHeight.....	405
GraphType	406
GroundSpeed	407
HasButtons.....	407
HasLines	408
HDOP	409
HeaderFooter	409
Heading	410
Hemisphere	410
Height	411
HideSelection	412
HScrollBar	412
HSplitScroll.....	413
HTextAlign.....	414
Icon.....	414
ID.....	415
IgnoreDefaultButton	415
Increment	416
Indent	417
InitiallyIconic	417
InputEditMode	418
InputFieldBackColor	419
InputFieldNamesVisible.....	420
Invert	420
IsFixValid	421
IsHeadingValid	421
Italic	422
InputFieldsVisible	422
Item[]	423
ItemGroup	423
ItemPictureIndex	424
ItemPictureIndex[]	425
ItemState	426
ItemStyle	427
Label.....	428
LabelWrap	430
LargePictureName[]	430
LargePictureWidth	432
LargePictureHeight.....	433
LargePictureMaskColor	434
Latitude.....	435
LeftMargin	436
LeftText	436
Legend	437

Limit.....	437
LineCallParamFlags	438
LineColor	439
LineMediaMode	440
LinesAtRoot.....	441
LinesPerPage.....	441
LineStyle.....	442
LinkUpdateOptions	443
LiveScroll.....	443
Longitude.....	444
MagneticVariation.....	445
MagneticVariationDirection	445
MajorGridLine.....	446
MajorDivisions	447
MajorTic.....	447
Map3DColors	448
Mask.....	449
MaskDataType	451
MaxBox	452
MaximumValue.....	452
MaxPosition	453
MaxValDateTime	453
MenuBar	454
MenuName	455
MessageClass.....	455
MessageOptions	456
MinBox	456
MinimumValue.....	457
MinMax.....	457
MinorGridLine.....	458
MinorDivisions	459
MinorTic.....	460
MinPosition	461
Minute.....	461
MinValDateTime	462
Mode	463
MultiSelect.....	463
Multiline	464
Name	465
NotificationID	466
NumberOfSatellites	467
Options	467
OriginalSize	468
OriginLine	469

OutlookCompatible	469
OverlapPercent	470
PageStyle	471
Password	472
PenMode	472
PerpendicularText	473
Perspective	474
PhoneNumber	474
PhoneNumberPlan	475
PhoneNumberType	476
PicturesAsFrame	477
PictureHeight	477
PictureIndex	478
PictureMaskColor	480
PictureName	481
PictureName[]	482
PictureOnRight	483
PictureWidth	484
Pointer	485
PopupMenu	485
Port	485
Position	486
PowerTipText	487
PowerTips	488
PrimaryLine	489
PRN	489
ProtocolType	490
RaggedRight	491
RawData	491
Resizable	493
ReturnsVisible	493
RightMargin	493
RightToLeft	494
Rotation	494
RulerBar	495
Roaming	495
RoundTo	496
RoundToUnit	497
Satellite[]	498
ScaleType	498
ScaleValue	499
ScannedData	500
ScannedSymbology	500
ScannedTimeStamp	501

ScannerType	501
Scrolling.....	502
SecondaryLine	502
SelectedTab	503
SerialPort.....	504
Series	505
SeriesSort.....	505
SetStep.....	506
ShadeBackEdge.....	507
ShowList.....	507
ShowHeader.....	508
ShowPicture	508
ShowSIPButton	509
ShowText	510
SIPOnFocus	510
SmallPictureHeight.....	511
SmallPictureMaskColor	512
SmallPictureName[].....	513
SmallPictureWidth.....	514
SmartMinimize.....	515
SNR.....	516
Sorted.....	516
SortType.....	517
SpacesVisible.....	518
Spacing	518
Spin	519
StartTime.....	519
StatePictureHeight	520
StatePictureMaskColor.....	521
StatePictureName[]	522
StatePictureWidth.....	523
Status	523
StdHeight.....	525
StdWidth.....	526
TabStop[]	527
TabOrder	527
TabPosition	528
TabTextColor.....	529
TabBackColor.....	530
TabsVisible.....	531
Tag	531
Tap_And_Hold_Indicator.....	532
Text	533
TextCase	534

TextColor	534
TextSize	535
ThreeState	536
ThirdState	537
Title	537
TitleBar	538
TodayDisplayApp	539
TodayDisplayText	540
TodayOrder	540
TodayRunApp	541
ToolBar	542
ToolBarAlignment	542
ToolBarHeight	542
ToolBarVisible	543
ToolBarWidth	543
ToolBarX	543
ToolBarY	544
TopMargin	544
Underline	544
UndoDepth	545
UnitsPerColumn	545
UnitsPerLine	546
UseCodeTable	547
ValidityPeriod	548
VDOP	549
Vendor	550
Version	550
View	551
ViewStyle	552
Visible	552
VScrollBar	553
VTextAlign	554
Weight	554
Width	555
WindowState	556
WindowType	557
WordWrap	557
Wrap	558
WrapMode	558
X	559
Y	560
ZoomPercent	561

CHAPTER 4	About Display Formats and Scrolling.....	563
	Using colors with display formats for PocketBuilder controls.....	563
	Using date display formats with PocketBuilder controls.....	564
	Using number display formats with PocketBuilder controls	565
	Using string display formats with PocketBuilder controls.....	567
	Using time display formats with PocketBuilder controls.....	567
	Scrolling in windows and user objects	569
Index		571

About This Book

Audience	This guide is for programmers building applications with PocketBuilder™.
How to use this book	This book describes the system-defined objects in PocketBuilder and their default properties, functions, and events.
Related documents	PocketBuilder reference set This manual is part of the PocketBuilder reference set, which is based on PowerBuilder® documentation. The reference set also includes the following manuals: • <i>Connection Reference</i> - Describes the database parameters and preferences you use to connect to a database in PocketBuilder. • <i>DataWindow Reference</i> - Lists the DataWindow® functions and properties and includes the syntax for accessing properties and data in DataWindow objects. • <i>PowerScript Reference</i> - Describes syntax and usage for the PowerScript® language including variables, expressions, statements, events, and functions. PocketBuilder documentation set The PocketBuilder documentation set includes the following manuals: • <i>Introduction to PocketBuilder</i> - Provides an overview of PocketBuilder features and the PocketBuilder development environment and a tutorial that leads the new user through the basic process of creating and deploying PocketBuilder applications. • <i>Resource Guide</i> - Presents advanced programming techniques and information about connecting to and synchronizing with a database. • <i>User's Guide</i> - Gives an overview of the PocketBuilder development environment and explains how to use the interface. Describes basic techniques for building the objects in a PocketBuilder application, including windows, menus, DataWindow objects, and user-defined objects. An appendix summarizes the differences between PocketBuilder and PowerBuilder.

Online Help Reference information for PowerScript properties, events, and functions is available in the online Help with annotations indicating which objects and methods are applicable to PocketBuilder.

SQL Anywhere® Studio documentation PocketBuilder is tightly integrated with Adaptive Server® Anywhere (ASA), UltraLite®, and MobiLink, which are components of SQL Anywhere Studio. You can install these products from the PocketBuilder setup program. Documentation for SQL Anywhere Studio is included in a separate collection on the PocketBuilder Technical Library CD and in online Help. For an introduction to these products, see Chapter 1 in the *Introduction to PocketBuilder*.

Other sources of information

Use the Sybase® Getting Started CD, the SyBooks™ CD, and the Technical Library Product Manuals Web site to learn more about your product.

- The Getting Started CD contains release bulletins and installation guides in PDF format and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD you need Adobe Acrobat Reader, which is downloadable at no charge from the Adobe Web site, using a link provided on the CD.
- The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access technical information about your product in an easy-to-use format.
- The Technical Library Product Manuals Web site is an HTML version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to the Technical Documents Web site (replacement for the Tech Info Library), the Solved Cases page, and Sybase newsgroups.

To access the Technical Library Product Manuals Web site, go to Product Manuals at <http://www.sybase.com/support/manuals/>.

Sybase EBFs and software updates

❖ Finding the latest information on EBFs and software maintenance

- 1 Point your Web browser to the Sybase Support Page at <http://www.sybase.com/support>.
- 2 Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.
- 3 Select a product.

- 4 Specify a time frame and click Go. A list of EBF/Maintenance releases is displayed.

Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as a Technical Support Contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click Edit Roles to add the “Technical Support Contact” role to your MySybase profile.

- 5 Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

Conventions

The formatting conventions used in this manual are:

Formatting example	To indicate
Retrieve and Update	When used in descriptive text, this font indicates: • Command, function, and method names • Keywords such as true, false, and null • Datatypes such as integer and char • Database column names such as emp_id and f_name • User-defined objects such as dw_emp or w_main
<i>variable or file name</i>	When used in descriptive text and syntax descriptions, oblique font indicates: • Variables, such as <i>myCounter</i> • Parts of input text that must be substituted, such as <i>pklname.pkd</i> • File and path names
File>Save	Menu names and menu items are displayed in plain text. The greater than symbol (>) shows you how to navigate menu selections. For example, File>Save indicates “select Save from the File menu.”
<code>dw_1.Update()</code>	Monospace font indicates: • Information that you enter in a dialog box or on a command line • Sample script fragments • Sample output fragments

If you need help

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online Help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.

PocketBuilder System Objects and Controls

About this chapter

This chapter provides overview information about PocketBuilder system objects and controls. This chapter also lists the PocketBuilder system objects not included in this book and the reason for not including them.

Contents

Topic	Page
What are system objects?	1
System object properties, events, and functions	2
System object inheritance hierarchy	2
Viewing system objects	4

What are system objects?

System objects

PocketBuilder **system class objects** are the built-in objects you use to develop your application. PocketBuilder 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

PocketBuilder **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

PocketBuilder system **structures** are a subset of system objects that contain properties that describe the state of other system objects or the system itself. For example, the Environment object is a structure that holds information about the computing platform the PocketBuilder 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	PocketBuilder 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 PocketBuilder language, that specify the processing that should happen when the event is triggered. PocketBuilder 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. 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, have no events associated with them.
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 PocketBuilder 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 its definition from its ancestor. The base class object typically implements generalized processing, and each descendant modifies the inherited processing as needed.

System object hierarchy

The PocketBuilder system objects compose 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 descend. The hierarchy also contains other (generic) base class objects that are not typically used in application development but are necessary parts of the logical organization of the hierarchy.

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:

- ClassDefinitionObject
- ConnectObject
- DragObject
- DrawObject
- DWObject
- ExtObject
- Function_Object
- GraphicObject
- GPS
- NonVisualObject
- OmControl
- OmCustomControl
- OmEmbeddedControl
- OmObject
- OmStorage
- OmStream
- ORB
- PBtoCPPObject
- PowerObject
- RemoteObject
- Service
- Structure
- WindowObject

Viewing system objects

Using the Browser

From within PocketBuilder, you can use the PocketBuilder 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 information about using the PocketBuilder Browser, see the *User's Guide*.

Using online Help

You can also use PocketBuilder 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 box to go directly to the correct topic.
- To see a list of the system objects and controls for which Help topics exist, select Objects and Controls from the Help contents list.
- In the Browser, select Help from the pop-up menu for the system object or control or one of its functions.

Help not available for base objects

If you select Help from the pop-up menu for a base object that descend from NonVisualObject, the Help topic for NonVisualObject displays. For other base objects, the Help topic for the Browser displays.

System Object Properties, Events, and Functions

About this chapter

This chapter lists the properties, events, and functions of PowerBuilder system objects and controls. This chapter does not include base class objects.

Contents

The objects and controls are listed alphabetically.

ADOResultSet object

The ADOResultSet object provides the ability to use ActiveX Data Object (ADO) record sets to return a result set to a client and to manipulate ADO Recordsets in PowerBuilder.

PocketBuilder	✗
PowerBuilder	✓

Application object

An application is a collection of PocketBuilder (or PowerBuilder) windows and objects that provide functionality for user activities, such as order entry or accounting activities. The Application object is the entry point into the applications.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

When a user runs an application, the Open event of the Application object is fired. The Open event triggers the script that initiates all the activity in the application.

Properties

Application property	Datatype	Description
AppName	String	Specifies the name of the Application object.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DDETimeOut	Integer	Not supported in PocketBuilder. 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).

Application property	Datatype	Description
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 do not specify a value, the value of AppName is used for DisplayName.
DWMessageTitle	String	Specifies the title of the message box for any runtime DataWindow errors encountered in the application. If you change the value of this property in script, the new value will be recognized only for DataWindows created (or painted) after the new value is set.
FreeDBLibraries	Boolean	Not supported in PocketBuilder. Specifies if you want PowerBuilder to free database interface libraries upon disconnecting from the database. The default is FALSE (PowerBuilder does not free the libraries upon disconnecting).
MicroHelpDefault	String	Not supported in PocketBuilder. 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	Not supported in PocketBuilder. 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 an operating system that supports right-to-left display. Values are: 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
TodayBackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215. The default value assigns the standard Today background color and watermark. This color is listed as Windows Background on the Today Item page of the Application object Properties view.
TodayDisplayApp	String	Specifies an application to control the display of a PocketBuilder Today item.
TodayDisplayText	String	Specifies the text that displays in a PocketBuilder slot on the Today screen.
TodayOrder	Integer	Specifies the position in which the custom item will appear on the Today screen.

Application property	Datatype	Description
TodayName	String	Specifies the name for the Today item that is saved in the device registry. This property is read-only. It cannot be updated after initialization.
TodayRunApp	String	Specifies an application that the user launches by clicking a PocketBuilder Today item.
TodayTextColor	Long	Specifies the numeric value of the color used for text: -2 to 16,777,215. The default value assigns the standard Today text color. This color is listed as Windows Text on the Today Item page of the Application object Properties view.
ToolbarFrameTitle	String	Not supported in PocketBuilder. Specifies the text that displays as the title for the FrameBar when it is floating.
ToolbarPopupMenuText	String	Not supported in PocketBuilder. Allows you to change the toolbar location text (Left, Top, Right, Bottom, Floating) in the Application's toolbar pop-up menu. Specify the text as a comma-separated list of items.
ToolbarSheetTitle	String	Not supported in PocketBuilder. Specifies the text that displays as the title for the SheetBar when it is floating.
ToolbarText	Boolean	Not supported in PocketBuilder. Specifies whether the text associated with the items in the toolbar displays. Values are: TRUE – Text displays in toolbar bar FALSE – Text does not display in toolbar
ToolbarTips	Boolean	Not supported in PocketBuilder. Specifies whether PowerTips display when text is not displayed on the buttons. Values are: TRUE – Power Tips are displayed FALSE – Power Tips are not displayed
ToolbarUserControl	Boolean	Not supported in PocketBuilder. Specifies whether users can use the toolbar pop-up menu to hide or show the toolbars, move toolbars, or show text. Values are: TRUE – Users can use pop-up menu FALSE – Users cannot use pop-up menu

Events

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

Functions

Application function	Datatype 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. This function can still be used but should be replaced by the system function SetLibraryList.
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.
TodaySave	Integer	Saves changes to the Today item to the registry and refreshes the Today screen. Changes must be saved to the registry to ensure that they are reflected in the Today screen after a reboot.
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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Windows CE platforms

This object can be used only in the development environment. It cannot be used in applications deployed to a Pocket PC or Smartphone device or emulator.

Properties

ArrayBounds property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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	Datatype 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.

BarcodeScanner base class

The BarcodeScanner is a base class for nonvisual objects that can read and process bar codes. The SymbolBarcodeScanner object and the SocketBarcodeScanner object implement all the methods and properties of this base class.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Properties

BarcodeScanner property	Datatype	Description
ScannerName	String	Name of the scanner you are using
ScannedData	String	Read-only data in the scanner's internal buffers at the time of the last RetrieveData call
ScannedSymbology	Integer	Read-only value with the decoder ID representing the symbology used by the last data read by the scanner
ScannedTimeStamp	DateTime	Read-only timestamp of the last scan

Events

BarcodeScanner event	Occurs
Constructor	When the object is created
Destructor	When the object is destroyed
ScannerInserted	When the interface layer first recognizes a physical scanner: either on initialization of the scanner or on physical insertion of the scanner
ScannerRemoved	When the scanner is physically removed
ScanTriggered	When asynchronous data is scanned

Functions

BarcodeScanner function	Datatype returned	Description
Close	Integer	Optional method that clears all buffers, detaches from the scanner firmware, and unloads all scanning DLLs. By default, this method is called by the BarcodeScanner object destructor.
DecoderName	String	Retrieves the short decoder name for the ID value passed as a function argument.
DeviceInfo	Integer	Retrieves device-specific settings, such as version numbers.
DeviceNames	Integer	Sets the names of the scanning devices.
EnableDecoder	Integer	Enables or disables the decoder whose ID value is passed in a function argument.
Flush	Integer	Flushes any old results in the scan buffers.
GetEnabledDecoders	Integer	Gets the array of enabled decoders.
GetSupportedDecoders	Integer	Gets the array of supported decoders.
Open	Integer	Loads the scanning DLLs and connects to the scanner firmware. This is typically the first method called on an object instance.
RetrieveData	Integer	Retrieves the data from the scanner internal buffers.
ScanAbort	Integer	Aborts all outstanding scan requests.
ScanNoWait	Integer	Sets the scan for asynchronous operation and an immediate return of scan data. In a typical implementation of this method, the ScanTriggered event sets the rearm flag to “true” for continuous scanning.
ScanWait	Integer	Sets the timeout period for a synchronous scan. The timeout period is the amount of time allowed to elapse before a scan resumes following a pause in the scanning.
SoftTrigger	Boolean	Sets or retrieves the soft trigger feature of a scanner.
Status	Integer	Returns the scanner status as an integer.

BiometricScanner base class

The BiometricScanner is a base class for nonvisual objects that can read and process biometric data. The HPBiometricScanner object implements the methods and properties of this base class for the purpose of scanning fingerprints.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Properties

BiometricScanner property	Datatype	Description
ScannerType	Integer	Bitset of authentication techniques used by the biometric scanner, including voice, fingerprint, and iris recognition

Functions

BiometricScanner function	Datatype returned	Description
Close	Integer	Optional method that clears all buffers, detaches from the scanner firmware, and unloads all scanning DLLs. By default, this method is called by the BiometricScanner object destructor.
FARPrecedence	Integer	Specifies whether the false acceptance rate (FAR) has precedence over the false rejection rate (FRR). This has meaning only if both the FAR and FRR have non-default values.
MaxFARRequested	Integer	Sets or retrieves the FAR.
MaxFRRRequested	Integer	Sets or retrieves the FRR.
Open	Integer	Loads the scanning DLLs and connects to the scanner firmware.
ScanCapture	Integer	Sets the timeout period for a (synchronous) scan. The timeout period is the amount of time allowed to elapse before a scan resumes following a pause in the scanning. Asynchronous biometric scans are not supported.
ScannedBitmap	Integer	Retrieves the image from the most recent scan. If height and width attributes are not provided in this function call, the bitmap file (BMP) representing the image uses the default image size.
ScannedMinutiae	Integer	Retrieves the encoded minutiae buffer from the most recent scan.

BiometricScanner function	Datatype returned	Description
ScannedQuality	Integer	Returns the quality rating from the most recent scan.
VerifyMatch	Integer	Compares the current candidate minutiae against the template (stored) minutiae records. A successful return value (+1) indicates they match within the specified FAR/FRR ratio.

CallLog object

The CallLog and CallLogEntry objects provide an interface to the entries in the call log on Smartphone and PocketPC - Phone Edition platforms. The call log lists information regarding all incoming and outgoing calls for a device, allowing a user to track and return missed calls, manage phone billing charges, and perform additional tasks.

PocketBuilder on PocketPC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

CallLog property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control

Events

CallLog event	Occurs
Constructor	When the object is created
Destructor	When the object is destroyed

Functions

CallLog function	Datatype returned	Description
GetEntry	CallLogEntry	Retrieves a call log entry based on an index value that you provide in the integer argument to this function
GetEntries	Integer	Retrieves the entire call log into an array of CallLogEntry objects

CallLogEntry object

The CallLog and CallLogEntry objects provide an interface to the entries in the call log on Smartphone and PocketPC - Phone Edition platforms. The CallLogEntry object is a system structure that holds information about an entry in a call log. CallLogEntry has no events. It inherits functions from the PowerObject base object.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

CallLogEntry property	Datatype	Description
CallState	Integer	Indicates the state of the call log entry. Missed calls can indicate an occupied phone line or no answer. Values are: • 1 Outgoing call that was answered • 2 Outgoing call that was missed • 12 Incoming call that was answered • 13 Incoming call that was missed
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.

CallLogEntry property	Datatype	Description
Dropped	Boolean	Indicates whether call was dropped by the service provider or ended normally. Values are: • true Call dropped by provider • false Call ended without service interruption
EndTime	DateTime	Indicates when the call ended.
Name	String	Text string associated with the phone number listed in the PhoneNumber property; typically a name.
PhoneNumber	String	Number of the incoming or outgoing call; if the caller ID of an incoming call is blocked or the number is otherwise unavailable, this entry is blank.
PhoneNumberPlan	String	Type of phone number; typically, “w” corresponds to a work phone number, “h” to a home phone number, and so on.
Roaming	Boolean	Indicates whether the call was placed or received locally or while roaming. Values are: • true Call made while roaming • false Call made locally
StartTime	DateTime	Indicates when the call started.

Camera object

A Camera object provides an interface to camera devices such as the HP Photosmart and VEO 130S SDIO cameras. You can preview an image in any visual control and save it as a JPEG file.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

Camera property	Datatype	Description
CameraName	String	Read-only name for the camera set with a specifier in the CameraType property.
CameraType	Integer	Specifier for the type of camera.

Camera property	Datatype	Description
ConfigParams	String	Not currently implemented. Reserved for future use.
DeviceHandle	UnsignedLong	Read-only file handle for the serial camera device.
DriverName	String	Read-only value optionally defined by the camera manufacturer.
DriverVersion	String	Read-only value optionally defined by the camera manufacturer.
Folder	String	Sets the folder for storing the snapped images. Required for HTC cameras.
Port	String	Sets the port for a camera. Required for the HP Photosmart and VEO 130S cameras.

Events

Camera event	Occurs
Constructor	When the object is created
Destructor	When the object is destroyed
Snapped	When an image has been captured by a digital camera device

Functions

Camera function	Datatype returned	Description
BeginPreview	Integer	Starts up the camera device's preview mode using the object of type GraphicObject that you specify in the function's only argument. For the HTC camera, launches the IA Camera Wizard.
CaptureImage	Integer	Saves the image as a file that you specify in this function's only argument.
Close	Integer	Closes a Camera communication channel if one is open and deactivates the data handlers.
EndPreview	Integer	Ends the camera device's preview mode.
GetAllowedImageAttributes	Integer	Passes the set of allowed image attributes for the camera device to the CameraImageAttributes structure object that you specify in the function's only argument.
GetOption	Integer	Gets the actual value of a device option that you specify in the function's only argument.
HasOption	Boolean	Queries the device to determine its support for an option that you specify in a CameraOption enumerated value.

Camera function	Datatype returned	Description
IsReadyToCapture	Boolean	Queries whether the device is set to capture an image to a file.
Open	Integer	Initializes data handlers and opens a communication channel to the camera device.
SetCaptureImageAttributes	Integer	Sets the image attributes from a CameraImageAttributes structure object that you specify in the function's only argument.
SetOption	Integer	Sets an option to a specific value.
SetPreviewImageAttributes	Integer	Sets the image attributes for preview mode.

CameraImageAttributes object

The CameraImageAttributes object is a read-only system structure that stores settings for a specific camera device. It is not used with cameras that depend on the IA Camera Wizard to configure camera settings and to preview and capture images. The CameraImageAttributes object has no events. It inherits functions from the PowerObject base object.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

CameraImageAttributes property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
Height	Integer	The height of the picture in pixels.
Identifier	Integer	Identifier for internal use.
IsForCapture	Boolean	Whether the data is for capture mode. Values are: - true – data is for capture mode - false – data is for preview mode
Width	Integer	The width of the picture in pixels
Zoom	Integer	The zoom factor for the image. Typically the zoom value is 1, 2, or 4.

CheckBox control

CheckBox controls 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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

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

Typically, check boxes have two states: on and off. You can also use a third state, unknown or unspecified. In the third state, the check is grayed.

Properties

CheckBox property	Datatype	Description
Automatic	Boolean	Specifies whether the control displays a mark when the user clicks it. Values are: 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 RGB function in the <i>PowerScript Reference</i> .
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleLowered! StyleRaised!
BringToTop	Boolean	Specifies whether PocketBuilder moves the control to the top of the front-to-back order. Values are: TRUE – Move to the top FALSE – Do not move to the top
Checked	Boolean	Specifies whether the control is selected. Values are: TRUE – Control is selected FALSE – Control is not selected
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.

CheckBox property	Datatype	Description
DragAuto	Boolean	Specifies whether PocketBuilder puts 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be clicked). Values are: TRUE – Control can be clicked FALSE – Control cannot be clicked
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. The application must be running on an appropriate version of PocketBuilder 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!

CheckBox property	Datatype	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	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.
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
RightToLeft	Boolean	Not supported in PocketBuilder. Specifies that characters should be displayed in right-to-left order. The application must be running on 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 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 RGB function in 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: TRUE – Control is in third state FALSE – Control is not in third state

CheckBox property	Datatype	Description
ThreeState	Boolean	Specifies whether the control has three states. Typically, CheckBox controls have only two states, such as on and off. Values are: 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: TRUE – Text is underlined FALSE – Text is not underlined
Visible	Boolean	Specifies whether the control is visible. Values are: 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.
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).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PocketBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

Functions

CheckBox function	Datatype 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.
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 PocketBuilder object that provides information about the class definition of a PocketBuilder object. You can examine a class in a PocketBuilder library or the class associated with an instantiated object.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Windows CE platforms

This object can be used only in the development environment. It cannot be used in applications deployed to a Pocket PC or Smartphone device or emulator.

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 PocketBuilder) or a user-defined object (defined in a PocketBuilder PKL)
- 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 and variables

Call FindFunctionDefinition to get a ScriptDefinition object describing the global function. Global variables are included in the VariableList array in the ClassDefinition object for the Application object.

Properties

ClassDefinition property	Datatype	Description
Ancestor	ClassDefinition	An object that represents the ancestor class. Ancestor is NULL when the ClassDefinition is describing PowerObject.
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 PowerObject containing information about the class definition of the object or control.
DataTypeOf	String	The system class name of the object. DataTypeOf is a string representation of a value of the Object enumerated datatype. Values are lowercase with no exclamation point. Sample values include: - window - string - any - dropdownlistbox For objects you have defined, the datatype 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 PocketBuilder as opposed to a class defined in a PKL by a user.
IsVariableLength	Boolean	Specifies whether the datatype has a fixed size. Values are: - TRUE – The datatype is variable length, meaning the datatype is a string, any, blob, or unbounded array. - FALSE – The datatype is a fixed length.
IsVisualType	Boolean	Indicates whether the class is a visual (displayable) or non-visual type. Values are: - 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 datatype.
LibraryName	String	The fully qualified name of the library the class was loaded from.
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.

ClassDefinition property	Datatype	Description
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.
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	Datatype returned	Description
ClassName	String	Returns the class of the object.
FindMatchingFunction	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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Using command buttons on Smartphone platforms

The Action key on a Smartphone device can trigger button events when a button has focus; however because there are no focus indicators or selection devices for command buttons on the Smartphone platform, applications should avoid using these controls.

Properties

Command Button property	Datatype	Description
BringToTop	Boolean	Specifies whether PocketBuilder moves the control to the top of the front-to-back order of the window. Values are: TRUE – Move to the top FALSE – Do not move to the top
Cancel	Boolean	Specifies whether the control acts as the Cancel button. (The Cancel button receives a Clicked event if the user presses Esc.) Values are: TRUE – Acts as the Cancel button FALSE – Does not act as the Cancel button
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
Default	Boolean	Specifies whether the control is the default control. The default control has a thick border and receives a Clicked event if the user presses Enter without selecting a control. Values are: TRUE – Acts as the default FALSE – Does not act as the default Editable controls Default behavior can be affected by editable controls on the window. For more information, see the <i>User's Guide</i> .
DragAuto	Boolean	Specifies whether PocketBuilder puts 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.

Command Button property	Datatype	Description
DragIcon	String	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. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be clicked). Values are: TRUE – Control is enabled FALSE – Control is not enabled
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. The application must be running on an appropriate version of PocketBuilder 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.

Command Button property	Datatype	Description
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.
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: TRUE – Text is underlined FALSE – Text is not underlined
Visible	Boolean	Specifies whether the control is visible. 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.

CommandButton event	Occurs
DragWithin	When a dragged control is within the control.
GetFocus	Before the control receives focus (before it is selected and becomes active).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PocketBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

Functions

CommandButton function	Datatype 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.
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 object

The Connection object specifies the parameters that PowerBuilder uses to connect to EAServer. 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.

PocketBuilder	✗
PowerBuilder	✓

ContextInformation object

The ContextInformation object provides information about an application's execution context, including current version information. Using this information, you can modify display characteristics and application behavior.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

ContextInformation property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
Handle	Long	Internal use only.

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	Datatype 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.
GetFixesVersion	Integer	Returns the fix level for the current PocketBuilder execution context.
GetHostObject	Integer	Provides a reference to the context's host object.
GetMajorVersion	Integer	Returns the major version for the current PocketBuilder execution context.
GetMinorVersion	Integer	Returns the minor version for the current PocketBuilder 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 PocketBuilder execution context.
GetVersionName	Integer	Returns complete version information for the current PocketBuilder 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 object

The ContextKeyword object provides environment information for the current context. In the default environment, the ContextKeyword object provides host workstation environment variables.

Pocket PC runtime

There are no environment variables on current Pocket PC platforms.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

ContextKeyword property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
Handle	Long	Internal use only.

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	Datatype returned	Description
ClassName	String	Returns the name assigned to the object.
GetContextKeywords	Integer	Retrieves one or more values associated with a specified keyword. Retrieves a blank string for Pocket PC platforms.

ContextKeyword function	Datatype 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 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.

CORBACurrent object

The CORBACurrent service object provides information about the EAServer transaction associated with a calling thread and enables the caller to control the transaction. The CORBACurrent object supports most of the methods defined by the CORBA Current interface.

PocketBuilder	✗
PowerBuilder	✓

CORBAObject object

The CORBAObject object gives PowerBuilder clients access to several standard CORBA methods. All proxy objects generated for EAServer components using the EAServer proxy generator are descendants of CORBAObject.

PocketBuilder	✗
PowerBuilder	✓

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. Because you can print DataStores, PocketBuilder provides some events and functions for DataStores that pertain to the visual presentation of the data.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

However, graph functions such as CategoryCount, CategoryName, GetData, SeriesCount, and so forth depend on the visual graph control, which is not created for a DataStore object. These functions return an error value or an empty string when used with DataStores.

Properties

DataStore property	Datatype	Description
DataObject	String	Specifies the name of the DataWindow or Report object associated with the control.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or 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.
ItemChanged	When the AcceptText and Update functions are called for the DataStore. Return codes: 0 - (Default) Accept the data value. 1 - Reject the data value and do not allow focus to change. 2 - Reject the data value but allow focus to change.
ItemError	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. Return codes: 0 - (Default) Reject the data value and show an error message box. 1 - Reject the data value with no message box. 2 - Accept the data value. 3 - Reject the data value but allow focus to change. If the return code is 0 or 1 (rejects the data), the field with the incorrect data regains the focus.
PrintEnd	When the printing of the DataStore ends.
PrintPage	Before each page of the DataStore is formatted for printing. Return codes: 0 - Do not skip a page. 1 - Skip a page.
PrintStart	When the printing of the DataStore starts.
RetrieveEnd	When the retrieval for the DataStore is complete.
RetrieveRow	After a row has been retrieved. Return codes: 0 - (Default) Continue. 1 - Stop the retrieval.
RetrieveStart	When the retrieval for the DataStore is about to begin. Return codes: 0 - (Default) Continue. 1 - Do not perform the retrieval. 2 - Do not reset the rows and buffers before retrieving the data from the database.

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

Functions

DataStore function	Datatype 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. (Returns an error value or an empty string for DataStores.)
CategoryName	String	Returns the name of the specified category in the specified graph. (Returns an error value or an empty string for DataStores.)
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.
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.
CreateFrom	Integer	Creates a DataStore object from the passed ResultSet object.
DataCount	Long	Returns the number of data points in the specified series in the specified graph. (Returns an error value or an empty string for DataStores.)

DataStore function	Datatype returned	Description
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. (Returns an error value or an empty string for DataStores.)
FindGroupChange	Long	Searches starting at a specified row for the first break for the specified group in the DataStore.
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. (Returns an error value or an empty string for DataStores.)
GenerateHTMLForm	Integer	Creates an HTML Form element containing columns for one or more rows. Also returns an HTML Style element containing style sheet information.
GenerateResultSet	Long	Not supported in PocketBuilder. Returns an EAServer result set from a PowerBuilder user object running as a component on EAServer.
GetBorderStyle	Border (enumerated)	Returns a Border enumerated datatype indicating the border style of the specified column in the DataStore. Border enumerated datatypes are: Box! Lowered! NoBorder! Raised! ResizeBorder! ShadowBox! Underline!
GetChanges	Long	Retrieves changes made to a DataStore into a blob. This function is used primarily in distributed applications.
GetChild	Integer	Stores in the specified variable the name of the child DataWindow in the specified column.

DataStore function	Datatype returned	Description
GetClickedColumn	Integer	Obtains the number of the column the user clicked or double-clicked in a DataStore.
GetClickedRow	Long	Obtains the number of the row the user clicked or double-clicked in a DataStore.
GetCurrentColumn	Integer	Returns the number of the current column in the DataStore.
GetCurrentColumnName	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. (Returns an error value or an empty string for DataStores.)
GetDataPieExplode	Integer	Reports the percentage that a pie slice is exploded in a pie graph. (Returns an error value or an empty string for DataStores.)
GetDataStyle	Integer	Finds out the appearance of a data point in a graph. (Returns an error value or an empty string for DataStores.)
GetDataValue	Integer	Returns the value of the specified data in the specified series in the specified graph. (Returns an error value or an empty string for DataStores.)
GetFormat	String	Returns the format used for display in the specified column of the DataStore.
GetFullState	Long	Retrieves the complete state of a DataStore into a blob. This function is used primarily in distributed applications.
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.

DataStore function	Datatype returned	Description
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.
GetRowFromRowId	Long	Gets the row number of a row in a DataStore from the unique row identifier associated with that row.
GetRowIdFromRow	Long	Gets the unique row identifier of a row in a DataStore from the row number associated with that row.
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. (Returns an error value or an empty string for DataStores.)
GetSQLSelect	String	Returns the current SELECT statement for the DataStore.
GetStateStatus	Long	Retrieves the current status of the internal state flags for a DataWindow and places this information in a blob. This function is used primarily in distributed applications.
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	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: • It can be inserted at the insertion point • It can replace all existing content This function only applies to DataStores whose content has the RichText presentation style.
InsertRow	Long	Inserts a new initialized row before the specified row in the DataStore.

DataStore function	Datatype returned	Description
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	Cancels the print job and deletes the spool file (if any) when the contents of the DataStore is sent to print. This function has two 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 DataStore.
Reset	Integer	Clears all the data from a DataStore.
Reset (graphs)	Integer	Deletes the data, the categories, or the series from a graph within a DataStore with an external data source.
ResetDataColors	Integer	Resets the color of a data point to the color specified for the series. (Returns an error value or an empty string for DataStores.)
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	Datatype 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	Saves the contents of the DataStore to the specified file, in the specified format, with or without column headings at the beginning
SaveAsAscii	Long	Saves the contents of a DataStore into a standard ASCII text file.
SelectRow	Integer	Selects or deselects the specified row of the DataStore.
SeriesCount	Integer	Returns the number of series in the specified graph. (Returns an error value or an empty string for DataStores.)
SeriesName	String	Returns the name of the specified series in the specified graph. (Returns an error value or an empty string for DataStores.)
SetBorderStyle	Integer	Sets the border style of the specified column in the DataStore.
SetChanges	Long	Applies changes captured with GetChanges to a DataStore. This function is used primarily in distributed applications.
SetColumn	Integer	Makes the specified column the current column in the DataStore.
SetDataPieExplode	Integer	Explodes a pie slice in a pie graph. (Returns an error value or an empty string for DataStores.)
SetDataStyle	Integer	For the specified data point in the specified series in the specified graph. (Returns an error value or an empty string for DataStores.)
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.
SetFullState	Long	Applies the contents of a DataWindow blob retrieved by GetFullState to a DataStore. This function is used primarily in distributed applications.
SetHTMLAction	Integer	Accepts action and context information about user interaction with the Web DataWindow client control in a web browser so that newly generated HTML can reflect any requested changes.
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. (Returns an error value or an empty string for DataStores.)

DataStore function	Datatype returned	Description
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.
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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

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

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

Properties

DataWindow property	Datatype	Description
Border	Boolean	Specifies whether the control has a border. Values are: 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 PocketBuilder moves the control to the top of the front-to-back order.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
ControlMenu	Boolean	Specifies whether the Control Menu box displays in the control title bar. Values are: TRUE – Control Menu box displays in the control title bar FALSE – Control Menu box does not display in the control title bar
DataObject	String	Specifies the name of the DataWindow object associated with the control.

DataWindow property	Datatype	Description
DragAuto	Boolean	Specifies whether PocketBuilder puts the control automatically into Drag Mode. DragAuto has these boolean values: TRUE – When the control is clicked, the control is automatically in Drag Mode. FALSE – When the control is clicked, the control is not automatically in Drag Mode. You have to manually put the control into Drag Mode by using the Drag function.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected). Values are: TRUE – Control is enabled FALSE – Control is not enabled
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: TRUE – Horizontal scrollbar is displayed FALSE – Horizontal scrollbar is not displayed
HSplitScroll	Boolean	Specifies whether the split bar displays in the control. Values are: 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.
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: 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: TRUE – Minimize Box displays FALSE – Minimize Box does not display

DataWindow property	Datatype	Description
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.
Resizable	Boolean	Specifies whether the DataWindow control is resizable. Values are: TRUE – DataWindow is resizable FALSE – DataWindow is not resizable
RightToLeft	Boolean	Not supported in PocketBuilder. Specifies that characters should be displayed in right-to-left order. The application must be running on 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 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: 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: 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
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
ButtonClicked	When the user clicks a button.
ButtonClicking	When the user clicks a button. This event occurs before the ButtonClicked event.
Clicked	When the user clicks 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.
DoubleClick	When the user double-clicks 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.
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).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
ItemChanged	When a field in the DataWindow has been modified and loses focus (for example, the user presses Enter, the Tab key, or an arrow key, or clicks the mouse on another field <i>within the DataWindow</i>). Return codes: 0 - (Default) Accept the data value. 1 - Reject the data value and do not allow focus to change. 2 - Reject the data value but allow focus to change.

DataWindow event	Occurs
ItemError	When a field has been modified, the field loses focus (for example, the user presses Enter, Tab, or an arrow key, or clicks the mouse on another field), and the field does not pass the validation rules for its column. Return codes: 0 - (Default) Reject the data value and show an error message box. 1 - Reject the data value with no message box. 2 - Accept the data value. 3 - Reject the data value but allow focus to change. If the Return code is 0 or 1 (rejects the data), the field with the incorrect data regains the focus.
ItemFocusChanged	When the current item in the control changes.
LoseFocus	When the DataWindow control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PocketBuilder event.
PrintEnd	When the printing of the DataWindow ends.
PrintPage	Before each page of the DataWindow is formatted for printing. Return codes: 0 - Do not skip a page. 1 - Skip a page.
PrintStart	When the printing of the DataWindow starts.
RButtonDown	When the right mouse button is pressed on the control. 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.
Resize	When the user or a script resizes a DataWindow control.
RetrieveEnd	When the retrieval for the DataWindow is complete.
RetrieveRow	After a row has been retrieved. Return codes: 0 - (Default) Continue. 1 - Stop the retrieval.
RetrieveStart	When the retrieval for the DataWindow is about to begin. Return codes: 0 - (Default) Continue. 1 - Do not perform the retrieval. 2 - Do not reset the rows and buffers before retrieving the data from the database.
RowFocusChanged	When the current row changes in the DataWindow.
RowFocusChanging	When the current row is about the change in the DataWindow. This event occurs before the RowFocusChanged event.
ScrollHorizontal	When the user scrolls right or left in the DataWindow control with the Tab or arrow keys or the scrollbar.

DataWindow event	Occurs
ScrollVertical	When the user scrolls up or down in the DataWindow control with the Tab or arrow keys or the scrollbar.
SQLPreview	After a Retrieve, Update, or ReselectRow function call and immediately before the SQL statement is submitted to the DBMS. The following return codes specify the action that takes place when the event occurs after an Update function call only: 0 - (Default) Continue. 1 - Stop. 2 - Skip this request and execute the next request.
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	Datatype 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.
ClearValues	Integer	Deletes all items from the value list associated with the specified column in the DataWindow control.

DataWindow function	Datatype returned	Description
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.
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. 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. 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.
FilteredCount	Integer	Returns the number of rows that are not visible because of the current filter.

DataWindow function	Datatype returned	Description
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.
GenerateHTMLForm	Integer	Creates an HTML Form element containing columns for one or more rows. Also returns an HTML Style element containing style sheet information.
GenerateResultSet	Long	Not supported in PocketBuilder. Returns an EAServer result set from a PowerBuilder user object running as a component on EAServer.
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 datatype indicating the border style of the specified column in the DataWindow control. Values are: Box! Lowered! NoBorder! Raised! ResizeBorder! ShadowBox! Underline!
GetChanges	Long	Retrieves changes made to a DataWindow into a blob. This function is used primarily in distributed applications.
GetChild	Integer	Stores in the specified variable the name of the child DataWindow in the specified column.

DataWindow function	Datatype returned	Description
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.
GetColumn	Integer	Returns the number of the current column in the DataWindow control.
GetColumnName	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 <code>GetDataValue</code> .
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.
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.
GetFullState	Long	Retrieves the complete state of a DataWindow into a blob. This function is used primarily in distributed applications.
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.

DataWindow function	Datatype returned	Description
GetItemTime	Time	Returns the time data in the specified row and column of the DataWindow control.
GetMessageText	String	Obtains the message text generated by a crosstab DataWindow object in the DataWindow control. Obsolete function GetMessageText is an obsolete function. 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.
GetRowFromRowId	Long	Gets the row number of a row in a DataWindow control from the unique row identifier associated with that row.
GetRowIdFromRow	Long	Gets the unique row identifier of a row in a DataWindow control from the row number associated with that row.
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.
GetSQLPreview	String	Returns the current SQL statement the DataWindow control is submitting to the database. Obsolete function GetSQLPreview is an obsolete function. SQL syntax is now available as an event argument.
GetSQLSelect	String	Returns the current SELECT statement for the DataWindow control.
GetStateStatus	Long	Retrieves the current status of the internal state flags for a DataWindow and places this information in a blob. This function is used primarily in distributed applications.
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.

DataWindow function	Datatype returned	Description
GetUpdateStatus	Integer	Stores the number of the row that will be updated in a variable and the dwBuffer enumerated datatype identifying the buffer containing the row in another variable. Obsolete function GetUpdateStatus is an obsolete function. 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.
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	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: • It can be inserted at the insertion point • It can replace all existing content This function only applies to DataWindow controls whose content has the RichText presentation style.
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.
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.

DataWindow function	Datatype returned	Description
OLEActivate	Integer	Not supported in PocketBuilder. Activates OLE for the OLE object in the specified row and column of the DataWindow control.
Paste	Integer	Inserts the contents of the clipboard into the edit control over the current row and column in the DataWindow control.
PasteRTF	Long	Not supported in PocketBuilder. 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.
Position	Integer	Syntax 1: Reports the insertion point's position in the edit control over the current row and column of the DataWindow control. 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.
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	Cancels the print job and deletes the spool file (if any) when the contents of the DataWindow control is sent to print. This function has two syntaxes. Use Syntax 1 when Syntax 1 of the Print function was used to send it to print.
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	Clears all the data from a DataWindow control.
Reset	Integer	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 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.

DataWindow function	Datatype returned	Description
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.
SaveAs	Integer	Saves the data represented in the specified graph in the DataWindow control to the specified file, in the specified format.
SaveAs	Integer	Saves the contents of the DataWindow control to the specified file, in the specified format, with or without column headings at the beginning.
SaveAsAscii	Long	Saves the contents of a DataWindow into a standard ASCII text file.
Scroll	Integer	Scrolls the edit control of a DataWindow control the specified direction the specified number of lines.
ScrollNextPage	Long	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). Syntax 2: Scrolls to the next page of the document in a DataWindow control whose contents has the RichText presentation style.
ScrollNextRow	Long	Scrolls the DataWindow control to the next row. ScrollNextRow changes the current row but does not change the current column.
ScrollPriorPage	Long	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). Syntax 2: Scrolls to the prior page of the document in a DataWindow control whose contents has the RichText presentation style.
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.

DataWindow function	Datatype returned	Description
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.
SelectText	Integer	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. 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.
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	Defines the action a DataWindow control takes following an event. Obsolete function SetActionCode is an obsolete function. You now set return codes in a return statement in the event script.
SetBorderStyle	Integer	Sets the border style of the specified column in the DataWindow control.
SetChanges	Long	Applies changes captured with GetChanges to a DataWindow. This function is used primarily in distributed applications.
SetColumn	Integer	Makes the specified column the current column in the DataWindow control.
SetDataPieExplode	Integer	Explodes a pie slice in a pie graph.

DataWindow function	Datatype returned	Description
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.
SetFullState	Long	Applies the contents of a DataWindow blob retrieved by GetFullState to a DataWindow. This function is used primarily in distributed applications.
SetHTMLAction	Integer	Accepts action and context information about user interaction with the Web DataWindow client control in a Web browser so that newly generated HTML can reflect any requested changes.
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.
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: Specifies that the series is an overlay.

DataWindow function	Datatype returned	Description
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.
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.
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 object

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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

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 events.

Properties

DataWindowChild property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.

Functions

DataWindowChild function	Datatype 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	String	Deletes all the items from a value list or code table associated with a DataWindow column.
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	Datatype returned	Description
DBErrorCode	Long	Returns the error code (number) generated by a database error. Obsolete function DBErrorCode is an obsolete function. 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. 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.
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.
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 datatype indicating the border style of the specified column in the DataWindowChild. Values are: Box! Lowered! NoBorder! Raised! ResizeBorder! ShadowBox! Underline!
GetChanges	Long	Retrieves changes made to a DataWindow into a blob. This function is used primarily in distributed applications.
GetChild	Integer	Provides a reference to a child DataWindow or to a report in a composite DataWindow, which you can use in DataWindow functions to manipulate that DataWindow or report.

DataWindowChild function	Datatype returned	Description
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.
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.
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.

DataWindowChild function	Datatype returned	Description
GetRowFromRowId	Long	Gets the row number of a row in a DataWindow control from the unique row identifier associated with that row.
GetRowIdFromRow	Long	Gets the unique row identifier of a row in a DataWindow control from the row number associated with that row.
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. 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.
GetUpdateStatus	Integer	Stores the number of the row that will be updated in a variable and the dwBuffer enumerated datatype identifying the buffer containing the row in another variable. Obsolete function GetUpdateStatus is an obsolete function. 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.
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.

DataWindowChild function	Datatype returned	Description
OLEActivate	Integer	Activates OLE for the OLE object in the specified row and column of the DataWindowChild.
ReselectRow	Integer	Accesses the database to reselect all columns that can be updated and refreshes all timestamp columns in a row in the DataWindowChild.
Reset	Integer	Clears all the data from a DataWindowChild.
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.
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.
SaveAs	Integer	Saves the contents of the DataWindowChild control to the specified file, in the specified format, with or without column headings at the beginning
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	Datatype returned	Description
SetBorderStyle	Integer	Sets the border style of the specified column in the DataWindowChild.
SetChanges	Long	Applies changes captured with GetChanges to a DataWindow. This function is used primarily in distributed applications.
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.
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.

DataWindowChild function	Datatype returned	Description
SetValue	Integer	Sets the value of the specified item in the value list or the code table of the specified column of the DataWindowChild.
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.

DialingDirectory object

The DialingDirectory object provides an interface to the entries of phone books on Smartphone and PocketPC - Phone Edition platforms. You can use this object with the DialingDirectoryEntry structure to merge multiple sources of phone numbers into a single logical entity.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

DialingDirectory property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control

Events

DialingDirectory event	Occurs
Constructor	When the object is created
Destructor	When the object is destroyed

Functions

DialingDirectory function	Datatype returned	Description
AcceptCall	Integer	Adds an entry to a dialing directory
AddEntry	Integer	Not implemented—reserved for future use
GetEntry	DialingDirectoryEntry	Retrieves an entry from a dialing directory
GetEntries	Integer	Retrieves the entire dialing directory into an array of DialingDirectoryEntry objects
UpdateEntry	Integer	Updates an entry in a dialing directory

DialingDirectoryEntry object

The DialingDirectoryEntry object is a system structure used with the DialingDirectory object to provide an interface to the entries of phone books on Smartphone and PocketPC - Phone Edition platforms. You can use these objects to merge multiple sources of phone numbers into a single logical entity. The DialingDirectoryEntry object has no events.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

DialingDirectoryEntry property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DataSource	Integer	Indicates the phone number source. Values are: • 0 Unspecified or unknown number • 1 Pocket Outlook contact list • 2 Pocket Outlook mobile number • 3 Pocket Outlook business number • 4 Pocket Outlook home number • 10 General SIM list • 11 Emergency dialing list • 12 Fixed dialing list • 13 Last dialing list • 14 Private numbers list
Index	Integer	Index of the entry in the phone book.
Name	String	Text string associated with the phone number listed in the PhoneNumber property; typically a name.
PhoneNumber	String	Number to dial. The number must be properly formatted for dialing to succeed.
PhoneNumberPlan	Integer	Indicates the calling (numbering) plan to use when dialing. Values are: • 0 Unspecified or unknown • 1 ISDN/Telephone numbering plan • 2 Data numbering plan (X.121) • 3 Telex numbering plan • 4 Private numbering plan • 5 ERMES numbering plan
PhoneNumberType	Integer	Indicates the protocol to use when dialing. Values are: • 0 Unspecified or unknown • 1 International • 2 National • 3 Network-specific • 4 Protocol- or subscriber-specific • 5 Alphanumeric address • 6 Abbreviated (speed dial) number

DropDownListBox control

A DropDownListBox control 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 text box. In other DropDownListBoxes, the user cannot modify the text box and must click the item or enter the first character of the item to select it.

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

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Smartphone platforms

Drop-down lists are automatically converted by PocketBuilder to spinner controls when deployed to Smartphone platforms. Edit functions for the drop-down list controls are not supported on these platforms.

Properties

DropDownList Box property	Datatype	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 text box portion of the control. Values are: TRUE – Can enter text in the text box FALSE – Cannot enter text in the text box AllowEdit must be TRUE when ShowList is TRUE.
AutoHScroll	Boolean	Specifies whether the text box portion of the control scrolls horizontally automatically when data is entered or deleted. Values are: TRUE – TextBox scrolls horizontally automatically FALSE – TextBox does not scroll horizontally automatically

DropDownList Box property	Datatype	Description
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215. For more information about color, see the RGB function in 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 PocketBuilder 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
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DragAuto	Boolean	Specifies whether PocketBuilder 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected). Values are: TRUE – Control is enabled FALSE – Control is not enabled
FaceName	String	Specifies the name of the typeface in which the text of the control displays (for example, ARIAL or COURIER).

DropDownList Box property	Datatype	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 PocketBuilder under an operating system that supports the selected character set. Values include: 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 is displayed in the control. Values are: TRUE – Horizontal scrollbar is displayed FALSE – Horizontal scrollbar is not displayed
InputEditMode	Integer	Specifies the input method edit mode. In PocketBuilder applications, you can use this property to set the SIP type on Pocket PC devices or the keypad entry mode on Smartphone devices.
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).

DropDownList Box property	Datatype	Description
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer that will be used for the control.
RightToLeft	Boolean	Not supported in PocketBuilder. Specifies that characters should be displayed in right-to-left order. The application must be running on 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
ShowList	Boolean	Specifies whether the option list always displays in the ListBox portion of the DropDownListBox when the control displays. Values are: TRUE – Option list always displays FALSE – Option list displays only when the user clicks the down arrow This property is usually set to FALSE. Note that AllowEdit must be TRUE when ShowList is TRUE.
Sorted	Boolean	Specifies whether the ListBox portion of the DropDownListBox is automatically sorted in ascending order. Values are: TRUE – ListBox automatically sorted FALSE – ListBox not sorted
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 in the control.
TextColor	Long	Specifies the numeric value of the color used for text: -2 to 16,777,215. For more information about color, see the RGB function in 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: TRUE – Text is underlined FALSE – Text is not underlined
Visible	Boolean	Specifies whether the control is visible. Values are: TRUE – Control is visible FALSE – Control is not visible

DropDownList Box property	Datatype	Description
VScrollBar	Boolean	Specifies whether a vertical scrollbar is displayed in the control. 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.
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).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
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 PocketBuilder 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	Datatype 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[] property of this control.
ClassName	String	Returns the name assigned to the control.
Clear	Integer	Clears the selected text from the control (but does not place it in the clipboard).
Copy	Integer	Copies (but does not delete) the selected text from the control to the clipboard.
Cut	Integer	Cuts (deletes) the selected text (if any) from the control and places it in the clipboard.
DeleteItem	Integer	Deletes the item indicated by the index from the ListBox portion of the control.
DirList	Boolean	Populates the ListBox portion of the DropDownListBox with a list of the files of the specified type that match the specified file pattern.
DirSelect	Boolean	Retrieves the current selection from the specified control and puts it in the specified variable.
Drag	Integer	Starts or ends the dragging of the control.
FindItem	Integer	Finds the first item in the ListBox portion of the control (after the specified index) that begins with a specified string.
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.
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.

DropDownListBox function	Datatype returned	Description
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.
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 control

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

PocketBuilder	✕
PowerBuilder	✓

DynamicDescriptionArea object

DynamicDescriptionArea is a PowerBuilder system object that stores information about the input and output parameters used in Format 4 of dynamic SQL.

PocketBuilder	✕
PowerBuilder	✓

DynamicStagingArea object

DynamicStagingArea is a PowerBuilder system object that stores information for use in dynamic SQL statements.

PocketBuilder	✕
PowerBuilder	✓

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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

EditMask property	Datatype	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 PocketBuilder 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
AutoVScroll	Boolean	Specifies whether PocketBuilder automatically scrolls up or down when data is entered into or deleted from the control. Values are: TRUE – Scrolls vertically automatically FALSE – Scrolling not automatic
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215. For more information about color, see the RGB function in the <i>PowerScript Reference</i> .

EditMask property	Datatype	Description
Border	Boolean	Specifies whether the control has a border. Values are: 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 PocketBuilder moves the control to the top of the front-to-back order in the window: TRUE – Control moved to top FALSE – Control not moved to top
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DisplayData	String	Specifies the data that initially displays in the control.
DisplayOnly	Boolean	Specifies whether the text in the control is display-only and cannot be changed by the user. Values are: TRUE – Text is display-only FALSE – Text can be changed by user
DragAuto	Boolean	Specifies whether PocketBuilder puts 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the <i>ICO</i> file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected): TRUE – Control is enabled FALSE – Control is not enabled
FaceName	String	Specifies the name of the typeface in which the text of the control displays (for example, ARIAL or COURIER).

EditMask property	Datatype	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 scroll bar displays in the control when all the data cannot be displayed at one time. Values are: TRUE – Horizontal scroll bar displayed FALSE – Horizontal scroll bar 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
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 control. FALSE – Trigger Clicked event; do not add new line in control (default).
Increment	Double	Specifies the increment of the spin.

EditMask property	Datatype	Description
InputEditMode	Integer	Specifies the input method edit mode. In PocketBuilder applications, you can use this property to set the SIP type on Pocket PC devices or the keypad entry mode on Smartphone devices.
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).
Mask	String	Specifies the mask to use to format and edit data in the control.
MaskDataType	MaskDataType (enumerated)	Specifies the datatype of the control. Values are: DateMask! DateTimeMask! DecimalMask! NumericMask! StringMask! TimeMask!
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.
RightToLeft	Boolean	Not supported in PocketBuilder. Specifies that characters should be displayed in right-to-left order. The application must be running on 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
SipOnFocus	Boolean	Whether to display the SIP when the control receives focus and minimize the SIP when the control loses focus. Values are: Yes — SIP opens and closes automatically. Yes — SIP opens and closes automatically. Painter: Show SIP on Focus option
Spin	Boolean	Specifies whether to scroll through the spin values. Values are: TRUE – Scroll through the spin values FALSE – Do not scroll through the spin values
TabOrder	Integer	Specifies the tab value of the control within the window (0 means the user cannot tab to the control).
TabStop[]	Integer	Specifies the positions of tab stops in the control.
Tag	String	Specifies the tag value assigned to the control.
Text	String	Specifies the text that displays in the control.

EditMask property	Datatype	Description
TextColor	Long	Specifies the color to be used for the text in the control. For more information about color, see the RGB function in the <i>PowerScript Reference</i> .
TextCase	TextCase (enumerated)	Specifies the case used to display text the user enters. Values are: AnyCase! Lower! Upper!
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 PocketBuilder 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 scroll bar displays in the control when not all the data can be displayed at one time. Values are: TRUE – Vertical scroll bar is displayed FALSE – Vertical scroll bar 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

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).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
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 PocketBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

Functions

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

EditMask function	Datatype returned	Description
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.
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 datatype 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 previous editing function performed in the control.

EnumerationDefinition object

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

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Windows CE platforms

This object can be used only in the development environment. It cannot be used in applications deployed to a Pocket PC or Smartphone device or emulator.

Properties

Enumeration Definition property	Datatype	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 PowerObject containing information about the class definition of the object or control.
DataTypeOf	String	The system class name or simple datatype of the variable. DataTypeOf is a string representation of a value of the Object enumerated datatype. Values are lowercase with no exclamation point. Sample values include: window string any dropdownlistbox For objects you have defined, the datatype 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.

Enumeration Definition property	Datatype	Description
IsSystemType	Boolean	Indicates whether the class is a system class, that is one of the classes defined within PocketBuilder as opposed to a class defined in a PKL by a user.
IsVariableLength	Boolean	Specifies whether the datatype has a fixed size. Always TRUE. Values are: TRUE – The datatype is variable length, meaning the datatype is a string, any, blob, or unbounded array. FALSE – The datatype 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	Datatype 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 datatype. It is used in the EnumerationDefinition class. It has no events.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Windows CE platforms

This object can be used only in the development environment. It cannot be used in applications deployed to a Pocket PC or Smartphone device or emulator.

Properties

Enumeration ItemDefinition property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
Name	String	The name of an enumerated value. For example, Left! is a named value of the enumerated datatype Alignment.
Value	Long	The numeric value associated with the name. For example, 0 is the value PocketBuilder associates with Left! The numeric value is only important if you are constructing source code for an object. Within PocketBuilder, you use the named value so that the datatype is correct.

Functions

Enumeration ItemDefinition function	Datatype 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.

Enumeration ItemDefinition function	Datatype returned	Description
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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

The Environment object has no events.

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

Properties

Environment property	Datatype	Description
CharSet	CharSet (enumerated)	The international character set used by PocketBuilder. Values include: • CharSetAnsi! • CharSetUnicode! • CharSetDBCS! • CharSetDBCSJapanese!
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
CPUType	CPUTypes (enumerated)	The type of CPU. For a complete list of CPUTypes values, see the Enumerated tab of the Browser.

Environment property	Datatype	Description
CurrentTrustMode	Integer	The assigned trust level of the process based on whether the certificat is a device's privileged store. Values are: 1 – The operating system trusts the program to run, but restricts certain function calls such as making phone calls or updating the Subscriber Identity Module (SIM) 2 – The operating system trusts the program to perform any function calls
DeviceID	String	Unique value for the device hosting the operating system.
Language	LanguageID (enumerated)	Specifies the value of the language setting for the machine. For a complete list of LanguageID values, see the Enumerated tab of the Browser.
MachineCode	Boolean	Specifies whether the application executable is machine code (compiled). Values are: TRUE – Executable is machine code FALSE – Executable is not machine code (pseudo-code)
NumberOfColors	Long	Number of colors on the screen.
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 CE 2002 and 4 for Windows CE 2003. On the desktop, this value would be 5 for Windows 2000 or XP.
OSMinorRevision	Integer	The point release of the operating system. For example, this value would be 0 for Windows CE 2002 and 20 for Windows CE 2003. On the desktop, this value would be 1 for Windows XP and 2 for Windows .NET.
OSType	OSTypes (enumerated)	Operating system or environment. For a complete list of OSType values, see the Enumerated tab of the Browser.
PBBuildNumber	Integer	The build number of this version of PocketBuilder.
PBFixesRevision	Integer	The maintenance version of PocketBuilder.
PBMajorRevision	Integer	The major version of PocketBuilder.
PBMinorRevision	Integer	The point release of PocketBuilder.
PBType	PBTypes (enumerated)	Version of PocketBuilder product. For a complete list of PBType values, see the Enumerated tab of the Browser.
PhoneCapable	Boolean	Indicates whether the device is capable of using the phone system, but does not determine whether subscriptions or phone accounts are enabled.
ScreenHeight	Long	Height of the screen in pixels.

Environment property	Datatype	Description
ScreenWidth	Long	Width of the screen in pixels.
Win16 (obsolete)	Boolean	Indicates the type of the operating system in which the application executable is running. Values are: TRUE – Executable is running under a 16-bit operating system FALSE – Executable is running under a 32-bit operating system

Functions

Environment function	Datatype 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.

Error object

The Error object is used to record execution-time errors. You can access the Error object from a script (typically in 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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

For more information about creating a custom Error object, see the chapter on user objects in the *User's Guide*. For information on using try-catch blocks to catch runtime and user-defined exceptions, see the *Resource Guide* and the *PowerScript Reference*.

Properties

Error property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
Line	Integer	Identifies the line in the script at which the error occurred.
Number	Integer	Identifies the PocketBuilder 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	Datatype 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.

ErrorLogging object

The ErrorLogging object provides the ability to write messages to the log file used by the object's container, such as the Jaguar server log for EAServer or the NT system application log for Microsoft Transaction Server.

PocketBuilder	✗
PowerBuilder	✓

Exception object

The Exception object inherits from the Throwable object and is the base class for user-defined or “checked” exceptions. A function or user-defined event must declare each checked exception it throws, and a caller to a function or event that throws a checked exception must either catch the exception or throw the exception itself.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Unlike RuntimeException objects, Exception objects do not have built-in properties that provide information about the location where the error occurred.

Properties

Exception property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
Text	String	Contains the text of the error message.

Events

Exception event	Occurs
Constructor	Immediately before the exception is thrown.
Destructor	Immediately after the exception is thrown.

Functions

Exception function	Datatype 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.
GetMessage	String	Returns the error message from objects of type Throwable.
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.
SetMessage	—	Sets an error message for an object of type Throwable.
TriggerEvent	Integer	Triggers a specific event in the object and executes the script for the event.
TypeOf	Object	Returns the type of the object.

FileDirect

The FileDirect object provides a nonvisual interface to the underlying file system on the Windows CE device.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

FileDirect property	Datatype	Description
FileHandle	Long	Identifier for the file you are handling with the FileDirect object

Functions

FileDirect function	Datatype returned	Description
Close	Integer	Closes a file
Open	Integer	Opens a file for processing
Read	Integer	Reads the requested number of bytes from a file
Seek	Integer	Moves the file pointer a specified distance
SetEndOfFile	Integer	Assigns the current position of a file pointer as the end of the file
Write	Integer	Writes the requested number of bytes to a file

GPSCoordinate object

The GPSCoordinate object is a system structure that stores latitude and longitude information used by the GPSFix object. The GPSCoordinate object has no events. It inherits functions from the PowerObject base object.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

GPSCoordinate property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
Degree	Integer	Position in degrees. The value can range from 0 to 179.
Hemisphere	Char	Hemispheric location from raw satellite feed. Values are: • N North • S South • E East • W West
Minute	Real	The minutes portion of the position in degrees. The value can range from 0 to 59.9999.

GPSFix object

The GPSFix object is a system structure that stores information about the current position fix. It is used by the SerialGPS object. The GPSFix object has no events. It inherits functions from the PowerObject base object.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

GPSFix property	Datatype	Description
Altitude	Real	Height above mean sea level (geoid).
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
FixQuality	Integer	Quality of the current position fix. Values are: • 0 Invalid • 1 GPS fix • 2 DGPS fix

GPSTime property	Datatype	Description
FixTime	Time	Universal Time Coordinate (UTC) at which the current position fix was read.
GeoidalHeight	Real	Height of geoid above WGS84 ellipsoid.
HDOP	Real	Relative accuracy of the horizontal position.
IsFixValid	Boolean	Whether there is valid fix data from the satellite.
Latitude	GPSCoordinate	North/South position in degrees.
Longitude	GPSCoordinate	East/West position in degrees.
NumberOfSatellites	Integer	Number of satellites used for the current position fix.

GPSHeading object

The GPSHeading object is a system structure that stores speed and directional information used by the SerialGPS object. The GPSHeading object has no events. It inherits functions from the PowerObject base object.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

GPSHeading property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
GroundSpeed	Real	Ground speed in knots.
Heading	Real	True heading without taking into account magnetic variation.
IsHeadingValid	Boolean	Whether the navigation receiver accepts the heading data.
MagneticVariation	Real	Magnetic variation in degrees. The value is 0 if magnetic variation information is not provided in the heading data.
MagneticVariationDirection	Char	Direction of magnetic variation. Possible values are E for East and W for West. The value is an empty character if magnetic variation information is not provided in the heading data.

GPSSatellitePosition object

The GPSSatellitePosition object is a system structure that stores information used by the GPSSatellitesInView object about the position of a satellite. The GPSSatellitePosition object has no events. It inherits functions from the PowerObject base object.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

GPSSatellitePosition property	Datatype	Description
Azimuth	Integer	Degrees from true north (values can range from 0 to 359)
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control
Elevation	Integer	Elevation in degrees (values can range from 0 to 90)
PRN	Integer	Satellite PRN number
SNR	Integer	Signal strength (values can range from 0 to 99)

GPSSatellitesInView object

The GPSSatellitesInView object is a system structure that stores information used by the SerialGPS object about the satellites in view. The GPSSatellitesInView object has no events. It inherits functions from the PowerObject base object.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

GPSSatellitesInView property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control
HDOP	Real	Relative accuracy of horizontal position (Horizontal Dilution of Position)
Satellite[]	GPSSatellitePosition	Array of satellite information
VDOP	Real	Relative accuracy of vertical position (Vertical Dilution of Position)

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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

Graph property	Datatype	Description
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215. For more information about color, see the RGB function in 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 PocketBuilder 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.

Graph property	Datatype	Description
CategorySort	grSortType	Specifies how the categories are sorted.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
Depth	Integer	Specifies the percent the depth is of the width of the graph.
DragAuto	Boolean	Specifies whether PocketBuilder will put the graph 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags a control (the <i>ICO</i> file). The default icon is a box the size of the control. When the user drags a control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Elevation	Integer	Specifies the angle of front-to-back elevation.
Enabled	Boolean	Specifies whether the control is enabled (can be selected). Values are: TRUE – Control can be selected FALSE – Control cannot be selected
FocusRectangle	Boolean	Specifies whether a dotted rectangle (the focus rectangle) will frame the control when it has focus. Values are: TRUE – Control will be framed when it has focus FALSE – Control will not be framed when it has focus

Graph property	Datatype	Description
GraphType	grGraphType (enumerated)	Specifies the type of the graph. Values are: 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: 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.
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.

Graph property	Datatype	Description
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: TRUE – Control is visible FALSE – Control is not visible
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.
DoubleClicked	When the control is double-clicked (selected or unselected).
DragDrop	When a dragged control is dropped on the control.
DragEnter	When a dragged control enters the control.
DragLeave	When dragged control leaves the control.
DragWithin	When a dragged control is within the control.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PocketBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

Functions

Graph function	Datatype 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 PocketBuilder 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.
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.
GetParent	PowerObject	Returns a reference to the name of the parent object.

Graph function	Datatype returned	Description
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.
Hide	Integer	Makes the control invisible.
ImportClipboard	Long	Copies the contents of the clipboard to the graph starting in the specified column.
ImportFile	Long	Copies the contents of the specified file to the graph starting in the specified column.
ImportString	Long	Imports the contents from the specified string to the graph starting in the specified column.
InsertCategory	Integer	Inserts a new category before an existing category in the graph.
InsertData	Long	Inserts a new data point into the graph before a specified data point and moves existing data points to the right.
InsertSeries	Integer	Inserts a new series before an existing series in the graph.
ModifyData	Integer	Changes the value of the specified data point in the specified series in the graph to the specified value, and optionally modifies the data for the specified tick mark.
Move	Integer	Moves the control to a specified location.
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.
SeriesCount	Integer	Determines how many series there are in the graph.
SeriesName	String	Obtains the name of the specified series in the graph.
SetDataPieExplode	Integer	Explodes a pie slice in a pie graph.

Graph function	Datatype returned	Description
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 PocketBuilder class grAxis is used as part of a graph control. Each graph has three grAxis objects: Category, Series, and Values.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

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	Datatype	Description
AutoScale	Boolean	Specifies whether PocketBuilder scales the axis automatically. Values are: TRUE – Automatically scale FALSE – Do not automatically scale
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DataType	grAxisDataType (enumerated)	Specifies the datatype 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.
DropLines	LineStyle (enumerated)	Specifies the type of drop line for the axis. Values are: 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!

grAxis property	Datatype	Description
MaximumValue	Double	Specifies the maximum value for the axis when its datatype is numeric.
MaxValDateTime	DateTime	Specifies the maximum value for the axis when its datatype is date or time.
MinimumValue	Double	Specifies the minimum value for the axis when its datatype is numeric.
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 datatype 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.
RoundToUnit	grRoundToType (enumerated)	Specifies the units for the rounding value. The units must be appropriate for the axis datatype. Values are listed below. For an axis of any datatype: RndDefault! For an axis of any numeric datatype: RndNumber! For an axis of type date or DateTime: RndYears! RndMonths! RndDays! For an axis of type time or DateTime: RndHours! RndMinutes! RndSeconds! RndMicroseconds!

grAxis property	Datatype	Description
ScaleType	grScaleType (enumerated)	Specifies the type of scale used for the axis. Values are: Linear! Log10! Loge!
ScaleValue	grScaleValue (enumerated)	Specifies the scale of values on the axis. Values are: Actual! Cumulative! Percentage! CumPercent!
SecondaryLine	LineStyle (enumerated)	Specifies the type of secondary line for the axis. See DropLines in this table for the list of values.
ShadeBackEdge	Boolean	Specifies whether the back edge of the axis is shaded. Values are: TRUE – Shaded FALSE – Not shaded

Functions

grAxis function	Datatype 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 PocketBuilder 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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

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	Datatype	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 RGB function in the <i>PowerScript Reference</i> .
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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.

grDispAttr property	Datatype	Description
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!
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 RGB function in the <i>PowerScript Reference</i> .

grDispAttr property	Datatype	Description
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	Datatype 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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

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

Properties

GroupBox property	Datatype	Description
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215. For more information about color, see the RGB function in 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 PocketBuilder moves the control to the top of the front-to-back order of the window.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DragAuto	Boolean	Specifies whether PocketBuilder puts 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the <i>ICO</i> file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the text of the GroupBox is visible or grayed. Values are: TRUE – Text is visible FALSE – Text is grayed
FaceName	String	Specifies the name of the typeface in which the text of the control displays (for example, ARIAL or COURIER).

GroupBox property	Datatype	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	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	Not supported in PocketBuilder. Specifies that characters should be displayed in right-to-left order. The application must be running on 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 RGB function in the <i>PowerScript Reference</i> .

GroupBox property	Datatype	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	Indicates whether the text in the control is underlined. Values are: TRUE – Text is underlined FALSE – Text is not underlined
Visible	Boolean	Specifies whether the control is visible. Values are: 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

GroupBox event	Occurs
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
Other	When a Windows message occurs that is not a PocketBuilder event.

Functions

GroupBox function	Datatype 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.

GroupBox function	Datatype 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 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.

HPBiometricScanner object

The HPBiometricScanner object implements the methods and properties of the BiometricScanner base class that involve scanning fingerprints. It dynamically loads Hewlett-Packard support DLLs for the hp iPAQ h5500 and h5550 Pocket PC devices.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Implementation details for the HPBiometricScanner object include the following:

- The ScannerType property value is implemented as a read-only value set to 8 for fingerprint scanning.
- The maximum acceptable values for the FAR and the FRR are set by low-level drivers and cannot be modified by MaxFARRequested and MaxFRRRequested function calls.
- An internal threshold value is used for any comparison made by a VerifyMatch function call. The FARAchieved parameter of VerifyMatch is always -1 because the FAR/FRR ratio is not used in determining a candidate match.

HPProgressBar control

You can use a progress bar to indicate the progress of a lengthy operation, such as an installation program that copies a large number of files. The HPProgressBar control is a horizontal rectangle that fills with the system highlight color as the operation progresses.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

HPProgressBar property	Datatype	Description
BringToTop	Boolean	Specifies whether PocketBuilder moves the control to the top of the front-to-back order. Values are: TRUE – Control moved to top FALSE – Control not moved to top
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.

HProgressBar property	Datatype	Description
DragAuto	Boolean	Specifies whether PocketBuilder puts 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Height	Integer	Specifies the height of the control, in PowerBuilder units.
MaxPosition	Unsigned Integer	Specifies the value of the Position property when the progress bar is at the right edge of the control. This value can be different from the end of the control's range, set with the SetRange function.
MinPosition	Unsigned Integer	Specifies the value of the Position property when the progress bar is at the left edge of the control. This value can be different from the start of the control's range, set with the SetRange function.
Pointer	String	Specifies the name of the stock pointer or file containing the pointer used for the control.
Position	Integer	Specifies the value of the current position within the range of the control (set with the SetRange function). The control uses the range and the current position to determine the percentage of the progress bar to fill with the highlight color.
SetStep	Integer	Specifies a step increment for the progress bar. The default is 10.
SmoothScroll	Boolean	Specifies that the control displays as a smooth scrolling bar instead of the default segmented bar.
TabOrder	Integer	Specifies the tab value of the control within the window (0 means the user cannot tab to the control).
Tag	String	Specifies the tag value assigned to the control.
Visible	Boolean	Specifies whether the control is visible. Values are: TRUE – Control is visible FALSE – Control is not visible
Width	Integer	Specifies the width of the control, in PowerBuilder units.

HProgressBar property	Datatype	Description
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units).
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.

Events

HProgressBar event	Occurs
Clicked	When the left mouse button is pressed on the control.
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
DoubleClicked	When the left mouse button is double-clicked on the 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.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PocketBuilder event.
RightClicked	When the right mouse button is pressed on the control.

Functions

HProgressBar function	Datatype 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.
OffsetPos	Integer	Moves the control's current position by the amount specified.

HProgressBar function	Datatype returned	Description
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.
SetRange	Integer	Sets the range of the control. The control uses the range and the current position to determine the percentage of the progress bar to fill with the highlight color.
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its properties.
Show	Integer	Makes the control visible.
StepIt	Integer	Moves the control's current position by the amount specified by the value of the SetStep property.
TriggerEvent	Integer	Triggers a 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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Usage note

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

Properties

HScrollBar property	Datatype	Description
BringToTop	Boolean	Specifies whether PocketBuilder moves the control to the top of the front-to-back order. Values are: TRUE – Control moved to top FALSE – Control not moved to top
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DragAuto	Boolean	Specifies whether PocketBuilder puts 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the <i>ICO</i> file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Height	Integer	Specifies the height of the control, in PowerBuilder units.
MaxPosition	Integer	Specifies the value of the Position 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.

HScrollBar property	Datatype	Description
StdHeight	Boolean	Specifies whether PocketBuilder uses the standard height for the control. Values are: 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.
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.

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).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
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 PocketBuilder event.
PageLeft	When the open space to the left of the scroll box is clicked.
PageRight	When the open space to the right of the scroll box is clicked.
RButtonDown	When the right mouse button is pressed on the control.

Functions

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

HTrackBar control

Like a scroll bar, a trackbar is used as a scrolling control, but clicking on the trackbar slider moves it in discrete increments instead of continuously. The HTrackBar control has a series of tick marks along the bottom of the trackbar channel.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

To enable this control to be used properly from the keyboard, you must add code to the `LineLeft`, `LineRight`, `PageLeft`, and `PageRight` events. The code you add should change the slider `Position` property by the appropriate value and then pass the new slider position to the object or objects you associate with the trackbar control. You must code the `Moved` event if you want the trackbar control to pass on the slider position after the slider is dragged with a mouse.

Usage note

Use a trackbar when you want the user to select a discrete value. For example, you might use a trackbar to enable a user to select a timer interval or the size of a window.

Properties

HTrackBar property	Datatype	Description
BringToTop	Boolean	Specifies whether PocketBuilder moves the control to the top of the front-to-back order. Values are: TRUE – Control moved to top FALSE – Control not moved to top
ClassDefinition	PowerObject	An object of type <code>PowerObject</code> containing information about the class definition of the object or control.
DragAuto	Boolean	Specifies whether PocketBuilder puts 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 <code>Drag</code> function.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the <i>ICO</i> file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Height	Integer	Specifies the height of the control, in PowerBuilder units.
LineSize	Integer	Specifies how far the slider moves in response to keyboard input from the arrow keys. Setting <code>LineSize</code> to 1 indicates moving 1 increment in the range of values specified by the <code>MaxPosition</code> and <code>MinPosition</code> properties.

HTrackBar property	Datatype	Description
MaxPosition	Integer	Specifies the value of the Position property when the slider is at the right edge of the control.
MinPosition	Integer	Specifies the value of the Position property when the slider is at the left edge of the control.
PageSize	Integer	Specifies how far the slider moves in response to a click in the slider channel area or pressing the Page keys. The default size is the difference between the MaxPosition and MinPosition properties divided by 5.
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 slider.
Slider	Boolean	Specifies whether or not the trackbar contains a slider.
SliderSize	Integer	Specifies the size of the slider on the trackbar. A value of 0 makes the slider the default size.
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.
TickFrequency	Integer	Specifies tick mark frequency. Setting TickFrequency to 1 indicates 1 tick mark for each increment in the trackbar range of values.
TickMarks	HTickMarks (enumerated)	Specifies where tickmarks should be displayed. Values are: HTicksOnRight! HTicksOnLeft! HTicksOnBoth! HTicksOnNeither!
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.

Events

HTrackBar 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).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
LineLeft	When the left arrow key is clicked.
LineRight	When the right arrow key is clicked.
LoseFocus	When the control loses focus (becomes inactive).
Moved	When the slider is moved (use the Position property to determine the new location).
Other	When a Windows message occurs that is not a PocketBuilder event.
PageLeft	When the Page Up key is clicked or when mouse clicks are made to the left of the slider in the trackbar channel.
PageRight	When the Page Down key is clicked or when mouse clicks are made to the right of the slider in the trackbar channel.
RButtonDown	When the right mouse button is pressed on the control.

Functions

HTrackBar function	Datatype 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.

HTrackBar function	Datatype returned	Description
Print	Integer	Prints the control.
Resize	Integer	Changes the size of the control.
SelectionRange	Integer	Sets a selection range for the trackbar. When you select a range, a blue line is drawn in the channel of the trackbar and two arrows are drawn where the tickmarks are placed to indicate the beginning and end of the selection range.
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.
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 object

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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

Inet property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.

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	Datatype 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.
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 object

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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

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

Properties

InternetResult property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.

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	Datatype 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.
InternetData	Integer	Processes the HTML data returned by a GetURL or PostURL function.
InternetStatus	Integer	Not used.
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.

JaguarORB object

The JaguarORB object allows PowerBuilder clients to access EAServer in the same manner as C++ clients and take advantage of CORBA features that are not available through the Connection object.

PocketBuilder	✗
PowerBuilder	✓

Line control

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

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

Line property	Datatype	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.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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 more information about color, see the RGB function in 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.
Visible	Boolean	Specifies whether the control is visible. Values are: TRUE – Control is visible FALSE – Control is not visible

Events

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

Functions

Line function	Datatype 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 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) scroll bars.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Smartphone platforms

The Action key on Smartphone devices expands the list and the arrow keys navigate within the list, but you must program a menu item or soft key to move the focus from the list box to another control in the same main window.

Properties

ListBox property	Datatype	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 RGB function in 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
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PocketBuilder moves the control to the top of the front-to-back order. Values are: TRUE – Control moved to top FALSE – Control not moved to top
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DisableNoScroll	Boolean	Specifies behavior of a scroll bar. Values are: TRUE – The scroll bar will always be visible but will be disabled when all the items can be accessed without it. FALSE – The scroll bar will be displayed only if it is necessary (based on the number of items and the height of the list box).
DragAuto	Boolean	Specifies whether PocketBuilder puts 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.

ListBox property	Datatype	Description
Enabled	Boolean	Specifies whether the control is enabled (can be selected). Values are: TRUE – Control can be selected FALSE – Control cannot be selected
ExtendedSelect	Boolean	Specifies whether users can select multiple items in the listbox at one time. Values are: TRUE – Users can select multiple items by clicking on an item and dragging the mouse up or down to select items; using Click or Shift+ Click to select a sequential group of items; or using Ctrl+ Click on multiple items. FALSE – Users cannot select multiple items. Used with MultiSelect The MultiSelect property allows users to select multiple items in a list box by simply clicking on the items. If MultiSelect = TRUE and ExtendedSelect = TRUE, then the behavior of ExtendedSelect takes precedence.
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.

ListBox property	Datatype	Description
HScrollBar	Boolean	Specifies whether a horizontal scroll bar displays. Values are: TRUE – Horizontal scroll bar displays FALSE – Horizontal scroll bar does not display
InputEditMode	Integer	Specifies the input method edit mode. In PocketBuilder applications, you can use this property to set the SIP type on Pocket PC devices or the keypad entry mode on Smartphone devices.
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 Used with ExtendedSelect The MultiSelect property allows users to select multiple items in a list box by simply clicking on the items. If MultiSelect = TRUE and ExtendedSelect = TRUE, then the behavior of ExtendedSelect takes precedence.
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
RightToLeft	Boolean	Not supported in PocketBuilder. Specifies that characters should be displayed in right-to-left order. The application must be running on 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
Sorted	Boolean	Specifies whether the items in the ListBox are sorted. Values are: 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.

ListBox property	Datatype	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 RGB function in 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: TRUE – Text is underlined FALSE – Text is not underlined
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 scroll bar is displayed on the right of the ListBox. Values are: TRUE – Vertical scroll bar is displayed FALSE – Vertical scroll bar 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).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.

ListBox event	Occurs
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Controls message occurs that is not a PocketBuilder event.
RButtonDown	When the right mouse button is pressed on the control.
SelectionChanged	When an item in the control is selected.

Functions

ListBox function	Datatype 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.
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.

ListBox function	Datatype returned	Description
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.
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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Smartphone platforms

The arrow keys on a Smartphone device navigate within the list, but you must program a menu item or soft key to move the focus from the list view to another control in the same main window.

Properties

ListView property	Datatype	Description
Accelerator	Integer	Specifies the ASCII value of the accelerator key assigned for the control.
AutoArrange	Boolean	Specifies whether PocketBuilder arranges icons automatically in large and small icon views.
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215. For more information about color, see the RGB function in 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
BorderStyle	BorderStyle (enumerated)	Specifies the border style of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PocketBuilder moves the control to the top of the front-to-back order.
ButtonHeader	Boolean	Specifies whether the column titles in report view appear as buttons.

ListView property	Datatype	Description
CheckBoxes	Boolean	Specifies whether the state images are replaced by check boxes. The check boxes are set to unchecked by default. The ListView control will process mouse and keyboard input to toggle the checked state. Values are: - TRUE – Check boxes are displayed. - FALSE – Check boxes are not displayed. The state of an item's check box can be determined by checking the state picture index for the item: Unchecked = 1 Checked = 2
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DeleteItems	Boolean	Specifies whether the user can delete a ListView item from a ListView control by pressing Delete.
DragAuto	Boolean	Specifies whether PocketBuilder puts 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
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.
Enabled	Boolean	Specifies whether the control is enabled (can be clicked). Values are: TRUE – Control can be clicked FALSE – Control cannot be clicked

ListView property	Datatype	Description
ExtendedSelect	Boolean	Specifies whether users can select multiple items in the listbox at one time. Values are: TRUE – Users can select multiple items by selecting outside all items and dragging to create a rectangle enclosing the desired items; by using Click or 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 PocketBuilder 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 used for the text in the control. Values are: Default! Fixed! Variable!
FullRowSelect	Boolean	Specifies whether an entire row may be selected in report view. Values are: • TRUE – In report view, an entire row may be selected • FALSE – In report view, just the item in the first column may be selected

ListView property	Datatype	Description
GridLines	Boolean	Specifies whether the report view displays gridlines: • TRUE – In report view, gridlines are displayed • FALSE – In report view, gridlines are not displayed
HeaderDragDrop	Boolean	Specifies whether column headers may be dragged to move columns in report view: • TRUE – In report view, column headers may be dragged • FALSE – In report view, dragging column headers will not move the columns
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
InputEditMode	Integer	Specifies the input method edit mode. In PocketBuilder applications, you can use this property to set the SIP type on Pocket PC devices or the keypad entry mode on Smartphone devices.
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.
LabelWrap	Boolean	Specifies whether long labels wrap under the ListView item in a large icon view. Values are: TRUE – Labels will wrap FALSE – Labels will not wrap LabelWrap does not apply to list, report, or small icon views.
LargePictureHeight	Integer	Specifies the size, in pixels, for the height of the picture used in the large icon view. In a script, this value can only be set before a large picture has been added to the large picture index list. If the large picture height is 0, PocketBuilder uses the height of the first picture added to the large picture index.
LargePictureMaskColor	Long	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.

ListView property	Datatype	Description
LargePictureName[]	String	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 PocketBuilder library. Not updated after initialization.
LargePictureWidth	Integer	Specifies the size, in pixels, for the width of the picture used in the large icon view. In a script, this value can only be set before a large picture has been added to the large picture index list. If the large picture width is 0, PocketBuilder uses the width of the first picture added to the large picture index.
OneClickActivate	Boolean	Specifies whether one click initiates the ItemActivate event: • TRUE – One click fires the ItemActivate event, causes the item to change color as the mouse moves over it (hot tracking), and causes the mouse to change to a hand cursor when it is over the item • FALSE – The item does not turn color as the mouse moves over it (assuming that TrackSelect = FALSE) and the mouse does not change to a hand cursor when it is over the item (assuming that TwoClickActivate = FALSE). However, the ItemActivate event is always initiated when an item is double clicked, even though OneClickActivate = FALSE and TwoClickActivate = FALSE.
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: TRUE – Scrolling is enabled FALSE – Scrolling is disabled
ShowHeader	Boolean	Specifies whether column titles appear in a report view. Values are: 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, PocketBuilder uses the height of the first picture added to the small picture index.
SmallPictureMaskColor	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.

ListView property	Datatype	Description
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 PocketBuilder library. Not updated after initialization.
SmallPictureWidth	Integer	Specifies the size, in pixels, for the width 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 width is 0, PocketBuilder uses the width of the first picture added to the small picture index.
SortType	grSortType	Specifies whether items are sorted alphabetically based on the item label. Values are: Ascending! Descending! UserDefined! Unsorted!
StatePictureHeight	Integer	Specifies the size, in pixels, for the height of the state picture. In a script, this value can only be set before a state picture has been added to the state picture index list. If the state picture height is 0, PocketBuilder uses the height of the first picture added to the state picture index list.
StatePictureMaskColor	Long	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.
StatePictureName[]	String	Specifies the name of the picture used as the state picture. The picture can be an icon, cursor, or bitmap supplied by the user or a stock picture from the PocketBuilder library. Not updated after initialization.
StatePictureWidth	Integer	Specifies the size, in pixels, for the width of the state picture. In a script, this value can only be set before a state picture has been added to the state picture index list. If the state picture width is 0, PocketBuilder uses the width of the first picture added to the state picture index list.
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.
TextColor	Long	Specifies the numeric value of the color used for text: -2 to 16,777,215.

ListView property	Datatype	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.
TrackSelect	Boolean	Specifies whether items appear in a different color when the mouse moves over them (hot tracking) and whether an item is selected if the mouse pauses over it. Values are: TRUE – An item changes color when the mouse moves over it, and an item is selected if the mouse pauses over it FALSE – An item does not change color nor is it selected when the mouse moves over or pauses on it
TwoClickActivate	Boolean	Specifies whether two clicks initiates the ItemActivate event: TRUE – Clicking twice (one click to select the item, one click to activate) fires the ItemActivate event, causes the item to change color as the mouse moves over it (hot tracking), and causes the mouse to change to a hand cursor when it is over the item FALSE – The item does not turn color as the mouse moves over it (assuming that TrackSelect = FALSE) and the mouse does not change to a hand cursor when it is over the item (assuming that OneClickActivate = FALSE). However, the ItemActivate event is always initiated when an item is double clicked, even though TwoClickActivate = FALSE.
Underline	Boolean	Specifies whether the text in the control is underlined. Values are: TRUE – Text is underlined FALSE – Text is not underlined
UnderlineCold	Boolean	When the OneClickActivate property is TRUE, this property specifies whether hot tracking (color of items changes when mouse moves over them) is turned on and items not highlighted are underlined. TRUE – Hot tracking is turned on and non-highlighted items are underlined FALSE – Non-highlighted items are not underlined
UnderlineHot	Boolean	When either the OneClickActivate or TwoClickActivate property is TRUE, this property specifies whether hot tracking (color of items changes when mouse moves over them) is turned on and items that are highlighted are underlined. TRUE – Hot tracking is turned on and highlighted items are underlined FALSE – Highlighted items are not underlined

ListView property	Datatype	Description
View	ListViewView	Specifies the layout of the ListBox. Valid values are: 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	Specifies whether the control is visible. Values are: 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.

ListView event	Occurs
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.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
InsertItem	When an item is being inserted.
ItemActivate	When an item is double clicked, or when the item is single clicked if the property OneClickActivate = TRUE, or when the item is clicked twice if the property TwoClickActivate = TRUE.
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 PocketBuilder 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	Datatype returned	Description
AddColumn	Integer	Adds a column to a ListView control report view.
AddItem	Integer	Adds an item to a ListView control.

ListView function	Datatype returned	Description
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.
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.
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.

ListView function	Datatype returned	Description
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.
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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

ListViewItem property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
CutHighlighted	Boolean	Specifies whether the item is the target of a cut operation. Values are: 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 ListView item.
DropHighlighted	Boolean	Specifies if the item is the target of a DragDrop operation. Values are: 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: 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.
PictureIndex	Integer	Identifies the large and small picture associated with the item.
Selected	Boolean	Specifies if the item is selected. Values are: TRUE – The item is selected FALSE – The item is not selected
StatePictureIndex	Integer	Identifies the state picture associated with the item.

Functions

ListViewItem function	Datatype 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 object

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

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

mailFileDescription property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
FileType	mailFileType (enumerated)	Specifies the type of the attachment file. The only valid value for PocketBuilder is: mailAttach! – data file
Filename	String	Specifies the name of the attachment file.
Pathname	String	Specifies the full path of the attachment file including the file name.
Position	Unsignedlong	Specifies the position of the attachment file within the message body. Required when sending multiple attachments.

Functions

mailFileDescription function	Datatype returned	Description
ClassName	String	Returns the class of an 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.

mailMessage object

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

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

mailMessage property	Datatype	Description
AttachmentFile[]	mailFileDescription	Specifies the file attachment for the current message. A mailFileDescription array contains information about an attachment file.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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.) 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.)
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.) TRUE – Return receipt requested FALSE – Return receipt not requested
Recipient[]	mailRecipient	Specifies the recipient of the current message. For mailSend, mailOriginator! is not a valid value for the Recipient property. The valid values are mailto!, mailcc!, and mailbcc!. To specify that the sender receive a copy of the message, use mailcc!.

mailMessage property	Datatype	Description
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.) TRUE – Message has not been read FALSE – Message has been read

Functions

mailMessage function	Datatype 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 object

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

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

mailRecipient property	Datatype	Description
Address	String	Specifies the electronic mail address of the current mail recipient. (Runtime only.)
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.

mailRecipient property	Datatype	Description
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	Datatype 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.

mailSession object

The mailSession nonvisual object signs on and is used for subsequent calls to the Windows CE messaging application program interface (CEMAPI) session.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

mailSession property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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	Datatype 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.
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 PocketBuilder application and mail.
mailLogon	mailReturnCode	Establishes a mail session for the PocketBuilder 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.

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

MDIClient object

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

PocketBuilder	✕
PowerBuilder	✓

Menu object

Typically, menus are lists of items (usually commands or options) that a user can select in the currently active window. Menus can display in a menu bar, in a drop-down or cascading menu, or as a pop-up menu.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

PocketBuilder 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 drop-down or cascading menu. When the user clicks a Menu object, a Clicked event is triggered. If there is a drop-down 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.

Smartphone platforms

On Smartphone devices, the main menu bar must contain exactly two menu bar items. Although you can add as many cascading menu items as you want, typically nine menu items is the maximum you can fit on a Smartphone platform screen at one time. Shortcut keys and tool tips are not supported.

Many of the properties and functions do not work on a Smartphone platform for a menu while it is loaded, however, you can use these same properties and functions to modify the menu when it is not the current menu or you can use the window object `ChangeMenu` function to replace the current window's menu.

Properties

Menu property	Datatype	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: TRUE – Object is selected FALSE – Object is not selected
ClassDefinition	PowerObject	An object of type <code>PowerObject</code> containing information about the class definition of the object or control.
Default	Boolean	Not supported in <code>PocketBuilder</code> . Specifies whether the menu item is the default and appears in a bold typeface. Only one item within a menu should be set to default. In context menus, the <code>Default</code> property is used to indicate the action that would have been performed if the user had double-clicked on the object rather than right-clicked on it. The property is also used to indicate which operation would have been performed if the item had been dragged with the left mouse button rather than the right mouse button. Values are: TRUE – Menu item is bolded FALSE – Menu item is not bolded

Menu property	Datatype	Description
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: TRUE – Object is enabled FALSE – Object is not enabled
Item[]	Menu	Specifies the Menu objects under a Menu object.
MenuItemType	MenuItemType (enumerated)	Not supported in PocketBuilder. Allows you to identify special Menu objects that are used differently on different platforms. Values are: MenuItemTypeAbout! MenuItemTypeExit! MenuItemTypeHelp! MenuItemTypeNormal!
MergeOption	MenuMerge Option (enumerated)	Not supported in PocketBuilder. Specifies how the object is affected when an OLE object is activated. Values are: EditMenu! Exclude! FileMenu! HelpMenu! Merge! WindowMenu! For more information about MergeOption, see the chapter on using OLE in <i>Application Techniques</i> in the PowerBuilder documentation set.
MicroHelp	String	Not supported in PocketBuilder. Specifies the text of the MicroHelp for the object.
ParentWindow	Window	Specifies the window that owns the Menu object.
ShiftToRight	Boolean	Not supported on Smartphone platforms. Specifies whether the Menu object shifts down or to the right when other Menu objects are added in a descendent menu. Values are: TRUE – Object shifts to right FALSE – Object shifts down
Shortcut	Integer	Not supported in PocketBuilder. 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	Datatype	Description
ToolBarItemDown	Boolean	Not supported in PocketBuilder. Specifies how the toolbar button appears. Values are: 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	Not supported in PocketBuilder. Specifies the name of the toolbar icon associated with the Menu object when it is down.
ToolBarItemBar Index	Integer	Not supported in PocketBuilder. 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	Not supported in PocketBuilder. 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	Not supported in PocketBuilder. Specifies the order of the item in the toolbar.
ToolBarItemSpace	Integer	Not supported in PocketBuilder. Specifies the amount of empty space before the item in the toolbar.
ToolBarItemText	String	Not supported in PocketBuilder. Specifies the text that displays in the toolbar item when the display text option is on for toolbars.
ToolBarItemVisible	Boolean	Not supported in PocketBuilder. Specifies whether the toolbar item displays. Values are: TRUE – The toolbar item is visible FALSE – The toolbar item is not visible
Visible	Boolean	Specifies whether the Menu object is visible. Values are: TRUE – Object is visible FALSE – Object is not visible

Events

Menu event	Occurs
Clicked	When the Menu object is clicked (selected or unselected).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
Selected	When the user moves to the Menu object using the arrow keys or the mouse.

Functions

Menu function	Datatype returned	Description
Check	Integer	Displays a check mark 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.
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 check mark next to the Menu object and sets the Checked property to FALSE.

MenuCascade object

Menu objects contained within a MenuCascade object appear as a drop-down button palette.

PocketBuilder	✕
PowerBuilder	✓

Message object

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

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

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 more information about creating a custom Message object, see the chapter on user objects in the *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	Datatype	Description
Handle	Long	The handle of the window or control.
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.

Message property	Datatype	Description
LongParm	Long	The long parameter for the event (this number comes from Windows). The parameter's value and meaning are determined by the event.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DoubleParm	Double	A numeric or a numeric variable.
StringParm	String	A string or a string variable.
PowerObjectParm	PowerObject	Any PocketBuilder object type including structures.
Processed	Boolean	A boolean value set in the script for the user-defined event or the Other event. Values are: TRUE – The script processed the event; do not call the default window process (DefWindowProc) after the event has been processed. FALSE – (Default) Call DefWindowProc after the event has been processed.
ReturnValue	Long	When Message.Processed is TRUE, specifies the value you want returned to Windows. This property is ignored when Message.Processed is FALSE.

Events

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

Functions

Message function	Datatype 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.

MultiLineEdit control

A MultiLineEdit control 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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

MultiLineEdit property	Datatype	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: 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: 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 RGB function in 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
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!

MultiLineEdit property	Datatype	Description
BringToTop	Boolean	Specifies whether PocketBuilder moves 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
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DisplayOnly	Boolean	Specifies whether the text is display-only and cannot be changed by the user. Values are: TRUE – Text cannot be changed by user FALSE – Text can be changed by user
DragAuto	Boolean	Specifies whether PocketBuilder 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected). Values are: TRUE – Control can be selected FALSE – Control cannot be selected
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!

MultiLineEdit property	Datatype	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.
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 scroll bar displays. Values are: TRUE – Horizontal scroll bar displays FALSE – Horizontal scroll bar 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.
InputEditMode	Integer	Specifies the input method edit mode. In PocketBuilder applications, you can use this property to set the SIP type on Pocket PC devices or the keypad entry mode on Smartphone devices.
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).
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.

MultiLineEdit property	Datatype	Description
RightToLeft	Boolean	Not supported in PocketBuilder. Specifies that characters should be displayed in right-to-left order. The application must be running on 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
SipOnFocus	Boolean	Whether to display the SIP when the control receives focus and minimize the SIP when the control loses focus. Values are: Yes — SIP is opened and closed automatically. No — SIP is not opened or closed automatically. Painter: Show SIP on Focus option.
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: AnyCase! Lower! Upper!
TextColor	Long	Specifies the numeric value of the color used for text: -2 to 16,777,215. For more information about color, see the RGB function in 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: TRUE – Text is underlined FALSE – Text is not underlined
Visible	Boolean	Specifies whether the control is visible. Values are: TRUE – Control is visible FALSE – Control is not visible

MultiLineEdit property	Datatype	Description
VScrollBar	Boolean	Specifies whether a vertical scroll bar is displayed on the right of the control. Values are: TRUE – Vertical scroll bar is displayed FALSE – Vertical scroll bar 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).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
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 PocketBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

Functions

MultiLineEdit function	Datatype 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.

MultiLineEdit function	Datatype 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 to 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.
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.
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.

MultiLineEdit function	Datatype 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.
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.

NotificationBubble object

The NotificationBubble object adds an icon and text to the navigation bar (notification tray) near the top of the Pocket PC device screen. The icon opens a notification bubble with text in HTML format.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Properties

NotificationBubble property	Datatype	Description
Body	String	HTML formatted text that is the body of the message in the notification bubble.
Caption	String	Text caption (title) for the notification bubble.
Duration	Integer	Length of time in seconds that the notification bubble displays before the system removes the icon and the bubble.

NotificationBubble property	Datatype	Description
Flags	UnsignedLong	Settings that change the behavior of the notification bubble. Values are: • 0 – (Default) No changes • 1 – Hides initial display of the notification bubble, displaying it only as an icon in the notification tray • 2 – Highlights notification bubble borders and title • 16 – Forces the display to turn on for the notification • 32 – Forces the notification to be silent and not vibrate regardless of system settings Values are additive. A value of 1, 3, 17, or 33 overrides the InitiallyIconic property. Values of 16 or greater are valid only for Pocket PC 2003 operating systems.
InitiallyIconic	Boolean	Determines whether the notification bubble is first displayed as an icon in the notification tray. Values are: • TRUE – Displays initially as an icon • FALSE – (Default) Displays initially as a notification bubble, unless overridden by a Flags property setting
NotificationID	UnsignedLong	Unique identifier number for the notification bubble.

Functions

NotificationBubble function	Datatype returned	Description
Icon	Integer	Specifies the notification icon to display in the notification tray.
Remove	Integer	Removes the notification bubble from the system.
SetMessageSink	Integer	Specifies the object that will receive event notifications from user events having the pbm_command event ID.
Update	Integer	Informs Windows CE that there is a new object or an update to an existing object in the notification tray.

OLEControl control

An OLEControl placed in a window can contain 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).

PocketBuilder	✗
PowerBuilder	✓

OLECustomControl control (OCX)

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

PocketBuilder	✗
PowerBuilder	✓

OLEObject object

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

PocketBuilder	✗
PowerBuilder	✓

OLEStorage object

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

PocketBuilder	✗
PowerBuilder	✓

OLEStream object

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

PocketBuilder	✗
PowerBuilder	✓

OLETxnObject object

The OLETxnObject object provides explicit control of MTS transactions to PowerBuilder clients using the SetComplete and SetAbort functions.

OLETxnObject inherits from the OLEObject object.

PocketBuilder	✗
PowerBuilder	✓

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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

Oval property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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 RGB function in 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 more information about color, see the RGB function in 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.

Events

Oval event	Occurs
Constructor	When the control is created.
Destructor	When the control is destroyed.

Functions

Oval function	Datatype 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.
PostEvent	Boolean	Adds an event to the end of the message queue for the control.
Resize	Integer	Changes the size of the control.
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.

PhoneCall object

The PhoneCall object provides an interface that allows a PocketBuilder application user to place a voice or data phone call from a Smartphone or PocketPC-Phone Edition platform.

Properties

PhoneCall property	Datatype	Description
ApplicationName	String	Name of the application placing the call. The application name is stored in the call log and displayed by some platforms/applications in user dialog boxes.
CallComment	String	Comment that is stored in the call log and displayed by some platforms in user dialog boxes.
CalledParty	String	Name of the party associated with the PhoneNumber property for outgoing calls.
CallerID	String	Caller ID associated with the PhoneNumber for incoming calls.
CallerIDName	String	Name of the party associated with the PhoneNumber property for incoming calls.
CallStatus	Unsigned long	The status of the current call.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
LineCallParamFlags	Unsigned long	A value that specifies a set of boolean parameters used to set options when making a phone call using the MakeCall function and for data transmission.
LineMediaMode	Unsigned long	A value that specifies the media type of a phone call.
LineStatus	Unsigned long	The status of the current line.
NoAnswerTimeout	Integer	Time in seconds to wait in the PROCEEDING or RINGBACK state after completion of dialing and before abandoning the call with a “no answer” and “disconnected” line call state.
PhoneNumber	String	Phone number of incoming or outgoing calls.
PromptUser	Boolean	Specifies whether to prompt user for confirmation before a call is placed. Values are: • true Prompt user • false Do not prompt user (default)
RingTone	String	Specifies the name and location of a sound file to be used as the ring tone for incoming calls.
TransmitCallerID	Integer	Specifies whether to block transmission of caller ID information: • 0 Use the system default setting (default) • 1 Block transmission of caller ID • 2 Send caller ID
VoiceCall	Boolean	Specifies whether the phone call is a voice call or a data call. Values are: • true Voice call (default) • false Data call

Events

PhoneCall event	Description
CallConnected	When the call, either inbound or outbound, is connected
CallDisconnected	When the call, either inbound or outbound, is disconnected
CallerID	When a caller ID associated with an incoming call has been detected
Constructor	When the object is created
Destructor	When the object is destroyed
IncomingCall	When notification of an incoming call, such as the ringing of the phone, is received
LineReply	Not currently implemented—reserved for future use

Functions

PhoneCall function	Datatype returned	Description
AcceptCall	Integer	Accepts a new incoming call.
AllowReceivingCalls	Integer	Allows reception of incoming calls. Typically called immediately after PhoneCall object is constructed. Takes a boolean argument: true = object events will be fired, false = no object events will be fired.
DropCall	Integer	Disconnects the current call.
MakeCall	Integer	Places a call using the properties of the PhoneCall object.
SetHold	Integer	Allows the user to place the call on hold or retrieve a call placed on hold.
SetMute	Integer	Allows the user to mute or unmute the line. Takes a boolean argument: true = mute the line, false = unmute the line. The return value indicates the previous state: 1 = muted, 2 = not muted.
SetRingTone	Integer	Sets the ring tone for the PhoneCall object. If you specify an empty string for the function's only argument, the default ring tone for the device is used.

Picture control

Picture controls can contain images in the following formats:

- Bitmaps, with *.BMP* extensions
- GIF or animated GIF files, with the *.GIF* extension
- JPEG files, with *.JPEG* or *.JPG* extensions
- PNG files, with *.PNG* extensions

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

You can create the image in another application or use a scanner to create it.

Picture format restrictions

Unlike BMP and GIF images and ICO icons, JPEG and PNG images must reside in the file system, not in a PocketBuilder resource file (PKR) or in a database blob. You must deploy JPEG or PNG files to the current application directory or include the full path of the deployed image files that you want to use in a Pocket PC device or emulator. PNG files are not supported on the desktop, only on Pocket PC devices and emulators, and can only be added at runtime.

Properties

Picture property	Datatype	Description
Border	Boolean	Specifies whether the control has a border. Values are: 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!

Picture property	Datatype	Description
BringToTop	Boolean	Specifies whether PocketBuilder 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
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DragAuto	Boolean	Specifies whether PocketBuilder puts 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the <i>ICO</i> file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected). Values are: TRUE – Control can be selected FALSE – Control cannot be selected
FocusRectangle	Boolean	Specifies whether a dotted rectangle (focus rectangle) frames the picture when it has focus. Values are: TRUE – Control will be framed when it has focus FALSE – Control will not be framed when it has focus
Height	Integer	Specifies the height of the control, in PowerBuilder units.
Invert	Boolean	Specifies whether the control displays with its colors inverted. Values are: TRUE – Colors are inverted FALSE – Colors are not inverted
Map3DColors	Boolean	Specifies whether the system 3D colors are mapped to the control. Values are: TRUE – Colors are mapped FALSE – Colors are not mapped

Picture property	Datatype	Description
OriginalSize	Boolean	Specifies whether the width and height properties of a bitmap image (picture) will be set to their original values. Values are: TRUE – Width and height set to original values FALSE – Existing width and height not changed 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.
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
PowerTipText	Long	Specifies a PowerTip 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: 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.
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.
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).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.

Picture event	Occurs
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PocketBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

Functions

Picture function	Datatype returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.
Draw	Integer	Draws a picture in the parent window at a specified location.
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.

PictureBox control

A PictureBox displays a picture 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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

The picture image can be in the following formats:

- Bitmaps, with *.BMP* extensions
- GIF or animated GIF files, with the *.GIF* extension
- JPEG files, with *.JPEG* or *.JPG* extensions
- PNG files, with *.PNG* extensions

Picture format restrictions

Unlike BMP and GIF images and ICO icons, JPEG and PNG images must reside in the file system, not in a PocketBuilder resource file (PKR) or in a database blob. You must deploy JPEG or PNG files to the current application directory or include the full path of the deployed image files that you want to use in a Pocket PC device or emulator. PNG files are not supported on the desktop, only on Pocket PC devices and emulators, and can only be added at runtime.

Properties

PictureBox property	Datatype	Description
BringToTop	Boolean	Specifies whether PocketBuilder moves the control to the top of the front-to-back order of the window.
Cancel	Boolean	Specifies whether the control acts as the Cancel button (the Cancel button receives a Clicked event if the user presses Esc). Values are: TRUE – Control acts as Cancel button FALSE – Control does not act as Cancel button
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.

PictureButton property	Datatype	Description
Default	Boolean	Specifies whether the control is the default PictureButton (the default PictureButton has a thick border and receives a clicked event if the user presses Enter without selecting an control). Values are: TRUE – Control is default PictureButton FALSE – Control is not default PictureButton Editable controls Default behavior can be affected by editable controls on the window. For more information, see the <i>User's Guide</i>.
DisabledName	String	Specifies the name of the picture (bitmap image) that displays when the control is disabled. If the string has no extension, PocketBuilder adds an appropriate extension.
DragAuto	Boolean	Specifies whether PocketBuilder puts 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the <i>ICO</i> file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected). Values are: TRUE – Control is enabled FALSE – Control is not enabled
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!

PictureButton property	Datatype	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.
HTextAlign	Alignment (enumerated)	Specifies how the text in the control is aligned. Values are: Center! Justify! Left! Right!
Italic	Boolean	Specifies whether the text in the control is italic. Values are: TRUE – Text is italic FALSE – Text is not italic
Map3DColors	Boolean	Specifies whether the system 3D colors are mapped to the control. Values are: TRUE – Colors are mapped FALSE – Colors are not mapped
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.
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
PowerTipText	Long	Specifies a PowerTip for the control
TabOrder	Integer	Specifies the tab value of the control within the window (0 means the user cannot tab to the control).

PictureButton property	Datatype	Description
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: TRUE – Text is underlined FALSE – Text is not underlined
Visible	Boolean	Specifies whether the control is visible. Values are: TRUE – Control is visible FALSE – Control is not visible
VTextAlign	VTextAlign (enumerated)	Specifies how the text in the control is aligned. Values are: Bottom! MultiLine! Top! VCenter! All these values except MultiLine! assume there is only one line of text.
Weight	Integer	Specifies the stroke weight of the text in the control; for example, 400 for normal or 700 for bold.
Width	Integer	Specifies the width of the control, in PowerBuilder units.
X	Integer	Specifies the X position (the distance from left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from top of the window), in PowerBuilder units.

Events

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.

PictureButton event	Occurs
GetFocus	Just before the control receives focus (before it is selected and becomes active).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PocketBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

Functions

PictureButton function	Datatype 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.
TypeOf	Object	Returns the type of the control.

PictureHyperLink control

The PictureHyperLink control is a descendant of the Picture control. The URL property of the PictureHyperLink control enables you to provide a hot link to a Web page. When the user clicks the control, the user's Web browser opens to display the page you specify.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

PictureHyperLink controls can contain images in the following formats:

- Bitmaps, with *.BMP* extensions
- GIF or animated GIF files, with the *.GIF* extension
- JPEG files, with *.JPEG* or *.JPG* extensions
- PNG files on devices and emulators only, not on the desktop

Usage note

If you know that your users have browsers that support URL completion, you can enter a partial address, such as:

`sybase.com`

You can, of course, enter a complete address, such as:

`http://www.sybase.com`

Properties

PictureHyperLink property	Datatype	Description
Border	Boolean	Specifies whether the control has a border. Values are: 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 PocketBuilder 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
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DragAuto	Boolean	Specifies whether PocketBuilder puts 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the <i>ICO</i> file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected). Values are: TRUE – Control can be selected FALSE – Control cannot be selected
FocusRectangle	Boolean	Specifies whether a dotted rectangle (focus rectangle) frames the picture when it has focus. Values are: TRUE – Control will be framed when it has focus FALSE – Control will not be framed when it has focus
Height	Integer	Specifies the height of the control, in PowerBuilder units.

PictureHyperLink property	Datatype	Description
Invert	Boolean	Specifies whether the control displays with its colors inverted. Values are: TRUE – Colors are inverted FALSE – Colors are not inverted
Map3DColors	Boolean	Specifies whether the system 3D colors are mapped to the control. Values are: TRUE – Colors are mapped FALSE – Colors are not mapped
OriginalSize	Boolean	Specifies whether the width and height properties of a bitmap image (picture) will be set to their original values. Values are: TRUE – Width and height set to original values FALSE – Existing width and height not changed 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 <i>BMP</i> , <i>JPG</i> , <i>JPEG</i> , <i>GIF</i> , or <i>PNG</i> is required. <i>PNG</i> files can only be added in code at runtime.
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
PowerTipText	Long	Specifies a PowerTip 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.
URL	String	Specifies the URL to open in the user's Web browser when the picture is clicked, provided no clicked event is coded. The status text displays the URL when the mouse passes over 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 (distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (distance from the top of the window), in PowerBuilder units.

Events

PictureHyperLink event	Occurs
Clicked	When the control is clicked (selected).
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
DoubleClicked	When the control is double-clicked (selected and activated).
DragDrop	When a dragged control is dropped on the control.
DragEnter	When a dragged control enters the control.
DragLeave	When a dragged control leaves the control.
DragWithin	When a dragged control is within the control.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PocketBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

Functions

PictureHyperLink function	Datatype returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.
Draw	Integer	Draws a picture in the parent window at a specified location.
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.

PictureHyperLink function	Datatype returned	Description
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.

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) scroll bars.

PocketBuilder	✗
PowerBuilder	✓

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.

PocketBuilder	✗
PowerBuilder	✓

POOM object

The POOM (Pocket Outlook Object Manager) object can be used to integrate a PocketBuilder application more tightly with the Pocket PC or Smartphone. It serves as an object store for the device contact manager and appointment calendar. The POOM object can be used to manipulate tasks, contacts, and appointments programmatically.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

POOM property	Datatype	Description
OutlookCompatible	Boolean	Specifies compatibility with Microsoft Outlook on the desktop. Values are: • TRUE – For synchronization with Outlook • FALSE – For synchronization with Microsoft Schedule+ This is a readonly property whose value is determined by system settings.
Version	String	Specifies the version of the Pocket Outlook object model. This is a readonly property whose value is determined by system settings.

Events

POOM event	Occurs
Constructor	When the object is created
Destructor	When the object is destroyed

Functions

POOM function	Datatype returned	Description
Add	Appointment, contact, or task object	Adds an appointment, contact, or task.
AddToInfraredQueue	Integer	Adds an appointment, contact, or task to the infrared queue.
GetAppointment	POOMAppointment	Obtains a single appointment using an index.
GetAppointmentFromOID	POOMAppointment	Obtains a single appointment using a specific object identifier (OID).
GetAppointments	Integer	Obtains an array of appointments. This function is overloaded. Syntax 1 obtains an array of appointments meeting specified criteria. Syntax 2 obtains an array of all appointments.
GetContact	POOMContact	Obtains a single contact using an index.
GetContactFromOID	POOMContact	Obtains a single contact using a specific object identifier (OID).
GetContacts	Integer	Obtains an array of contacts. This function is overloaded. Syntax 1 obtains an array of contacts meeting specified criteria. Syntax 2 obtains an array of all contacts.
GetTask	POOMTask	Obtains a single task using an index.
GetTaskFromOID	POOMTask	Obtains a single task using a specific object identifier (OID).
GetTasks	Integer	Obtains an array of tasks. This function is overloaded. Syntax 1 obtains an array of tasks meeting specified criteria. Syntax 2 obtains an array of all tasks.
Login	Integer	Logs in to a Pocket Outlook session. Required to add or remove appointments, contacts, and tasks. This function has an optional argument that specifies the parent window of the Outlook session.
Logout	Integer	Logs out of the Pocket Outlook session. Failure to log out can result in an out-of-memory error.
ReceiveFromInfrared	Integer	Receives items over an infrared link and distributes them to destination folders.
Remove	Integer	Removes an appointment, contact, or task.
SendToInfrared	Integer	Sends the entire infrared queue.

POOMAppointment object

You can add or remove appointments only after you log in using the general POOM object. The POOMAppointment object lets you add or remove recipients, and get or set appointment recurrences, but you must use functions on the general POOM object to add or remove appointments.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

POOMAppointment property	Datatype	Description
AppointmentDuration	Integer	Specifies duration of an appointment in minutes. This is a readonly property that is calculated if you set the AppointmentStart and AppointmentEnd properties.
AppointmentEnd	DateTime	Specifies the end of an appointment. The AppointmentEnd property must always be greater than the AppointmentStart property.
AppointmentStart	DateTime	Specifies the start of an appointment.
Body	String	Specifies annotations for an appointment in text format. The text limit is 60 K. For more information, see the BodyInk property.
BodyInk	Blob	Specifies annotations for an appointment in a blob in Pocket Word Ink (PWI) format that can be set to an Ink control such as the PocketBuilder Signature control. You must call the Add function on the POOM object to add the appointment to the repository before you set or get the Body and BodyInk properties. When you set one of these properties, its value is automatically updated in the repository.
BusyStatus	POOMBusyStatus (enumeration)	Specifies the availability of the user. Values are: • BusyFree! (default) • BusyOutOfOffice! • BusyTentative!
Categories	String	Specifies the categories to which an appointment is assigned. The text limit for the assigned categories is 1023 characters.

POOMAppointment property	Datatype	Description
IsAllDayEvent	Boolean	Specifies whether the appointment is for the whole day. The default is false.
IsMeeting	Boolean	Indicates whether there is a meeting request. This is a readonly property. If there are recipients for the appointment, the appointment is considered to be a meeting.
IsRecurring	Boolean	Indicates whether an appointment recurs. This is a readonly property that is set if you specify a recurrence pattern.
IsReminderSet	Boolean	Specifies whether a reminder for an appointment is set.
IsSensitive	Boolean	Specifies whether the appointment is normal or private.
Location	String	Specifies the location of an appointment. The text limit for the location is 1023 characters.
OID	Long	Specifies the object ID for an appointment. This is a direct, read-only property.
ReminderMinutesBeforeStart	Integer	Specifies the time in minutes before an appointment reminder is displayed.
ReminderOptions	Long	Specifies how the reminders for a task are transmitted. Values are: • 1 – activates the light-emitting diode (LED) • 2 – activates the vibration indicator if one exists • 4 – displays a dialog box • 8 – plays the sound specified in the ReminderSoundFile property • 16 – repeats the reminder Values are additive. For example, a value of 24 specifies that the designated sound file is played and the reminder is repeated.
ReminderSoundFile	String	Specifies the path to a sound file. Can be used only if IsReminderSet is set to true and the ReminderOptions property is set.
Subject	String	Specifies a subject (topic) for an appointment. The text limit for the subject is 4095 characters.

Functions

POOMAppointment function	Datatype returned	Description
AddRecipient	Integer	Adds the specified recipient for an appointment.
Cancel	Integer	Cancels the appointment, but does not remove it from the repository. Call the Remove function on the POOM object to remove the appointment from the repository.
ClearRecurrencePattern	Integer	Clears the recurrence pattern for an appointment and sets it as an appointment with a single instance.
Display	Integer	Displays the appointment using the default display in Pocket Outlook or the window specified as an optional argument to the POOM Login function.
GetRecipients	Integer	Gets an array of recipients from the POOMRecipient object.
GetRecurrence	POOMRecurrence	Returns the POOMRecurrence object associated with the appointment.
RemoveRecipient	Integer	Removes the specified recipient for the appointment.
Send	Integer	Sends the appointment (meeting request) to all recipients.
SetRecurrence	Integer	Sets a recurrence pattern for an appointment.
Update	Integer	Updates the existing appointment in the repository.

POOMContact object

Only after you log in using the general POOM object can you add or remove contacts. The POOMContact object lets you copy and display contacts, but you must use functions on the general POOM object to add or remove contacts.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

Unless otherwise noted, properties of the String datatype have a limit of 1023 characters, and properties of the Date datatype require values greater than 12/31/1899 and less than 1/1/3000.

POOMContact property	Datatype	Description
Anniversary	Date	Specifies an anniversary date for a contact.
AssistantName	String	Specifies the name of an assistant to a contact.
AssistantTelephoneNumber	String	Specifies the telephone number of an assistant to a contact.
Birthday	Date	Specifies the birthday of a contact.
Body	String	Specifies annotations for a contact in text format. The text limit is 60K. For more information, see the BodyInk property.
BodyInk	Blob	Specifies annotations for a contact in a blob in Pocket Word Ink (PWI) format that can be set to an Ink control such as the PocketBuilder Signature control. You must call the Add function on the POOM object to add the contact to the repository before you set or get the Body and BodyInk properties. When you set one of these properties, its value is automatically updated in the repository.
BusinessTelephoneNumber	String	Specifies a business telephone number for a contact.
Business2TelephoneNumber	String	Specifies a second business telephone number for a contact.
BusinessFaxNumber	String	Specifies a business fax number for a contact.
BusinessAddressStreet	String	Specifies a business street address for a contact.
BusinessAddressCity	String	Specifies the city in which a contact's business is located.
BusinessAddressState	String	Specifies the state in which a contact's business is located.
BusinessAddressCountry	String	Specifies the country in which a contact's business is located.
BusinessAddressPostalCode	String	Specifies the postal code for a contact's business location.
CarTelephoneNumber	String	Specifies a car telephone number for a contact.
Categories	String	Specifies the categories to which a contact is assigned.
Children	String	Specifies the names of the children of a contact.
CompanyName	String	Specifies the company name of a contact.
Department	String	Specifies the department of a contact.
Email1Address	String	Specifies an e-mail address for a contact.
Email2Address	String	Specifies a second e-mail address for a contact.
Email3Address	String	Specifies a third e-mail address for a contact.
FileAs	String	Specifies a filing string for a contact.
FirstName	String	Specifies the first name of a contact.
HomeTelephoneNumber	String	Specifies a home telephone number for a contact.
Home2TelephoneNumber	String	Specifies a second home telephone number for a contact.

POOMContact property	Datatype	Description
HomeFaxNumber	String	Specifies a home fax for a contact.
HomeAddressStreet	String	Specifies a home street address for a contact.
HomeAddressCity	String	Specifies a home city for a contact.
HomeAddressState	String	Specifies the home state or province of a contact.
HomeAddressCountry	String	Specifies the home country of a contact.
HomeAddressPostalCode	String	Specifies the home postal code of a contact.
JobTitle	String	Specifies the job title of a contact.
LastName	String	Specifies the last name of a contact.
MiddleName	String	Specifies the middle name of a contact.
MobileTelephoneNumber	String	Specifies a mobile telephone number for a contact.
OfficeLocation	String	Specifies the location of a contact.
OID	Long	Specifies the object identifier (OID) of a contact. This is a read-only property.
OtherAddressStreet	String	Specifies a second street address for a contact.
OtherAddressCity	String	Specifies the city of a contact's secondary residence.
OtherAddressState	String	Specifies the state or province of a contact's secondary residence.
OtherAddressCountry	String	Specifies the country of a contact's secondary residence.
OtherAddressPostalCode	String	Specifies the postal code of a contact's secondary residence.
PagerNumber	String	Specifies a page number for a contact.
RadioTelephoneNumber	String	Specifies a radio telephone number for a contact.
Spouse	String	Specifies the name of the spouse of a contact.
Suffix	String	Specifies the suffix of a contact's name.
Title	String	Specifies the title of a contact.
WebPage	String	Specifies the Web page of a contact.
YomiCompanyName	String	Specifies Japanese phonetic rendering of a contact's company name.
YomiFirstName	String	Specifies Japanese phonetic rendering of a contact's first name.
YomiLastName	String	Specifies Japanese phonetic rendering of a contact's last name.

Functions

POOMContact function	Datatype returned	Description
Display	Integer	Displays the contact using the default display in Pocket Outlook or the window specified as an optional argument to the POOM Login function
Update	Integer	Updates the existing contact in the repository

POOMRecipient object

The POOMRecipient object is used by the POOMAppointment object.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

POOMRecipient property	Datatype	Description
Name	String	Required property that returns the name of the recipient. This is a readonly property.
Address	String	Required property that sets or returns the e-mail address of the recipient.

POOMRecurrence object

The POOMRecurrence object is used by the POOMAppointment and POOMTask objects.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

POOMRecurrence property	Datatype	Description
DayOfMonth	Integer	Specifies the day of the month on which an appointment recurs. Values must be in the range 1 to 31. This property is valid only if the RecurrenceType property is set to RecursMonthly! or RecursYearly!.
DayOfWeekMask	Long	Specifies the day or days of the week on which an appointment or task occurs. Values are: • 1 – Recurs on Sunday • 2 – Recurs on Monday • 4 – Recurs on Tuesday • 8 – Recurs on Wednesday • 16 – Recurs on Thursday • 32 – Recurs on Friday • 64 – Recurs on Saturday Values are additive. For example, a value of 20 specifies that the appointment or task recurs on Tuesdays and Thursdays. This property is valid only if the RecurrenceType property is set to RecursMonthNth!, RecursWeekly!, or RecursYearNth!.
Duration	Long	Specifies the duration for a recurring appointment. This is used to calculate the duration if either StartTime or EndTime is not set and is ignored if both are set.
EndTime	DateTime	Specifies the end time of an appointment recurrence. The date portion is ignored. If you set a value for an appointment recurrence using this property, it overrides the AppointmentEnd property of the appointment object.
HasNoEndDate	Boolean	Specifies whether the recurrence pattern has an end date. The default is true (no end date).
Instance	Long	Specifies the week of any month in which the recurrence occurs. Values must be in the range 1 to 5. This property is valid only if the RecurrenceType property is set to RecursMonthNth! or RecursYearNth!.
Interval	Long	Specifies the interval between recurrences. This property is valid only if the RecurrenceType property is set to RecursDaily!, RecursMonthly!, RecursMonthNth!, or RecursWeekly!. The interval must be between 1 and 999 and is in units of days, weeks, or months.

POOMRecurrence property	Datatype	Description
MonthOfYear	Integer	Specifies the month of a recurrence. This property is valid only if the RecurrenceType property is set to RecursYearly! or RecursYearNth!. The value must be in the range 1 to 12.
Occurrences	Long	Specifies the number of occurrences.
PatternStartDate	DateTime	Specifies the start time of a recurring appointment. The date portion is ignored. If you set a value for an appointment recurrence using this property, it overrides the AppointmentStart property of the appointment object.
PatternEndDate	DateTime	Specifies the end date of the recurrence. You must set the HasNoEndDate property to false to use this property. Setting and saving this property sets the value of the Occurrences property.
RecurrenceType	POOMRecurrenceType (enumeration)	Specifies the type of recurrence. Values are: • RecursDaily! – Recurs every day • RecursWeekly! – Recurs every week • RecursMonthly! – Recurs every month • RecursMonthNth! – Recurs every N months • RecursYearly! – Recurs every year • RecursYearNth! – Recurs every N years
StartTime	DateTime	Specifies the start time of a recurring appointment. The date portion is ignored. If you set a value for an appointment recurrence using this property, it overrides the AppointmentStart property of the appointment object.

POOMTask object

Only after you log in using the general POOM object can you add or remove tasks. The POOMTask object lets you copy and display tasks, and get and set task recurrences, but you must use functions on the general POOM object to add or remove tasks.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

Unless otherwise noted, properties of the String datatype have a limit of 1023 characters, and properties of the Date datatype require values greater than 12/31/1899 and less than 1/1/3000.

POOMTask property	Datatype	Description
Body	String	Specifies annotations for a task in text format. The text limit is 60K. For more information, see the BodyInk property.
BodyInk	Blob	Specifies annotations for a task in a blob in Pocket Word Ink (PWI) format that can be set to an Ink control such as the PocketBuilder Signature control. You must call the Add function on the POOM object to add the task to the repository before you set or get the Body and BodyInk properties. When you set one of these properties, its value is automatically updated in the repository.
Categories	String	Specifies the categories to which a task is assigned.
CompletedDate	Date	Specifies the date when the task was completed.
DueDate	Date	Specifies the date when the task is scheduled to be completed.
Importance	POOMTaskPriority (enumeration)	Specifies the importance or priority of the task. Values are: • ImportanceLow! • ImportanceNormal! • ImportanceHigh!
IsComplete	Boolean	Specifies whether the task is complete. Values are: • true – To indicate that the task is complete • false – (Default) To indicate that the task is not complete
IsRecurring	Boolean	Indicates whether a task recurs. This is a readonly property that is set if you specify a recurrence pattern.
IsReminderSet	Boolean	Specifies whether the user should be reminded of a task.
IsSensitive	Boolean	Specifies whether the task is private.
IsTeamTask	Boolean	Specifies whether the task is a team task.
OID	Long	Specifies the object identifier for a task. This is a read-only property.

POOMTask property	Datatype	Description
ReminderOptions	Long	Specifies how the reminders for a task are transmitted. Values are: • 1 – activates the light-emitting diode (LED) • 2 – activates the vibration indicator if one exists • 4 – displays a dialog box • 8 – plays the sound specified in the ReminderSoundFile property • 16 – repeats the reminder Values are additive. For example, a value of 24 specifies that the specified sound file is played and the reminder is repeated.
ReminderSoundFile	String	Specifies the path to a sound file. Can be used only if IsReminderSet is set to true and ReminderOptions property is set.
ReminderTime	DateTime	Specifies the date and time of a reminder for the task.
StartDate	Date	Specifies the date when the task is scheduled to start. A value must be assigned to this property before you can add a task to the POOM repository. (You can add a task by calling Add on the POOM object.)
Subject	String	Specifies the subject for a task. The text limit for the subject is 4095 characters.

Functions

POOMTask function	Datatype returned	Description
ClearRecurrencePattern	Integer	Clears the recurrence pattern for a task and sets it as a task with a single instance.
Display	Integer	Displays the task using the default display in Pocket Outlook or the window specified as an optional argument to the POOM Login function.
GetRecurrence	POOMRecurrence	Returns the POOMRecurrence object associated with the appointment.
SetRecurrence	Integer	Sets a recurrence pattern for a task.
SkipRecurrence	Integer	Moves to the next occurrence in a recurring task.
Update	Integer	Updates the existing task in the repository.

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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The ProfileCall object has no events.

Properties

ProfileCall property	Datatype	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 datatype ProfileRoutine containing the destination of the call.
CallingLine	ProfileLine	An object of datatype 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 datatype ProfileRoutine containing the routine that initiated the call.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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.
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	Datatype 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.

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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The ProfileClass object has no events.

Properties

ProfileClass property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
LibraryName	String	The name of the library that contains the class. The value is " " for system classes and embedded SQL statements.
Name	String	The name of the class or the string <code>Embedded SQL</code> 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	Datatype 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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The ProfileLine object has no events.

Properties

ProfileLine property	Datatype	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.

ProfileLine property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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.
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	Datatype 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.
OutgoingCallList	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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The ProfileRoutine object has no events.

Properties

ProfileRoutine property	Datatype	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 embedded SQL activities, the value is <code>Embedded SQL</code> . For global and system functions, the value is an invalid object.
ClassDefinition	PowerObject	An object of type <code>PowerObject</code> containing information about the class definition of the object or control.
HitCount	Long	The number of times this routine was called or the number of objects created/destroyed.

ProfileRoutine property	Datatype	Description
Kind	ProfileRoutine Kind (enumerated)	The kind of routine node. Values are: 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.
Name	String	The name of the routine including the argument datatypes and return value. For embedded SQL 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	Datatype 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.

ProfileRoutine function	Datatype returned	Description
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 PocketBuilder 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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The Profiling object has no events.

Properties

Profiling property	Datatype	Description
ApplicationName	String	The name of the application used to generate the trace file.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.

Profiling property	Datatype	Description
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 (ie, the proper amount of time has been subtracted from the timestamps 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	Datatype 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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

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	Datatype	Description
Automatic	Boolean	Specifies whether the control becomes dark when it is clicked. Values are: 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>RGB</i> function in 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 PocketBuilder moves 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
Checked	Boolean	Specifies whether the item is selected (the center is dark). Values are: TRUE – Control is selected FALSE – Control is not selected

RadioButton property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DragAuto	Boolean	Specifies whether PocketBuilder puts 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the <i>ICO</i> file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected). Values are: TRUE – Control is enabled FALSE – Control is not enabled
FaceName	String	Specifies the name of the typeface in which the text of the control displays (for example, HELV or COURIER).
FontCharSet	FontCharSet (enumerated)	Specifies the font character set used for the text in the control. 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!

RadioButton property	Datatype	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	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.
RightToLeft	Boolean	Specifies that characters should be displayed in right-to-left order. The application must be running on 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 (0 means the user cannot tab to the control). In a GroupBox, the up and down arrow keys are used to move among RadioButtons in a specified sequence. To permit tabbing in a GroupBox, change the tab value of the GroupBox to 0 and assign nonzero tab values to the RadioButtons (the default tab value for the RadioButtons in a GroupBox is 0).
Tag	String	Specifies the tag value assigned to the control.
Text	String	Specifies the text that displays next to the control.
TextColor	Long	Specifies the numeric value of the color used for text: -2 to 16,777,215. For more information about color, see the RGB function in 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: TRUE – Text is underlined FALSE – Text is not underlined

RadioButton property	Datatype	Description
Visible	Boolean	Specifies whether the control is visible. Values are: 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).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PocketBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

Functions

RadioButton function	Datatype returned	Description
ClassName	String	Returns the name assigned to the control.
Drag	Integer	Starts or ends the dragging of the control.

RadioButton function	Datatype 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.
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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

Rectangle property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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 RGB function in 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 more information about color, see the RGB function in the <i>PowerScript Reference</i> .
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.

Rectangle property	Datatype	Description
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.

Events

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

Functions

Rectangle function	Datatype 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.
PostEvent	Boolean	Adds an event to the end of the message queue for the control.
Resize	Integer	Changes the size of the control.
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.

ResultSet object

The ResultSet object provides the ability to use EAServer result sets or ActiveX Data Object (ADO) record sets to return a result set to a client. Use ResultSet objects with the CreateFrom and GenerateResultSet DataStore functions.

PocketBuilder	✗
PowerBuilder	✓

ResultSets object

The ResultSets object provides the ability to handle multiple result sets returned from EAServer.

PocketBuilder	✗
PowerBuilder	✓

RichTextEdit control

A RichTextEdit control contains a document that 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.

PocketBuilder	✗
PowerBuilder	✓

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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

Round Rectangle property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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 RGB in 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 more information about color, see RGB in the <i>PowerScript Reference</i> .

Round Rectangle property	Datatype	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.

Events

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

Functions

RoundRectangle function	Datatype 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.

RoundRectangle function	Datatype returned	Description
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.

RuntimeError object

The `RuntimeError` object inherits from the `Throwable` object and is used by the PocketBuilder virtual machine (PBVM) to throw runtime errors. Runtime errors are also called unchecked exceptions. You do not need to declare where they might be thrown and you do not need to catch them as you do checked exceptions.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

When a `RuntimeError` is thrown, its properties are populated automatically with the runtime information associated with the line where the error occurred. If a `RuntimeError` is not handled, the Application object `SystemError` event is triggered and the global `Error` object is populated with the runtime information.

The following derived types provide more robust error-handling capabilities:

- `DivideByZeroError` – thrown when an attempt is made to divide by zero.
- `NullObjectError` – thrown when an attempt is made to access an object using a null reference.
- `DWRuntimeError` – thrown when a DataWindow error occurs that is not handled by an Error event script.

The descendants of `RuntimeError` allow you to handle specific runtime errors. For example, you could catch only `NullObjectError` exceptions in a specific block of code. Alternatively, you can catch all runtime errors with a single `CATCH` statement.

Properties

RuntimeError property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control
Class	String	Name of the class where the exception occurred
DLLName	String	(PBXRuntimeError only) Name of the PocketBuilder extension DLL where the exception occurred
Line	Integer	Line number where the exception occurred
Number	Integer	Identifies the PocketBuilder error
ObjectName	String	Name of the object where the exception occurred
RoutineName	String	Name of the event or routine where the exception occurred
Text	String	Text associated with the type of exception

Events

RuntimeError event	Occurs
Constructor	When the exception is thrown
Destructor	Immediately after the exception is thrown

Functions

RuntimeError function	Datatype 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.
GetMessage	String	Returns the error message from objects of type Throwable.
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.
SetMessage	—	Sets an error message for an object of type Throwable.
TriggerEvent	Integer	Triggers a specific event in the object and executes the script for the event.
TypeOf	Object	Returns the type of the object.

ScriptDefinition object

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

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Windows CE platforms

This object can be used only in the development environment. It cannot be used in applications deployed to a Pocket PC or Smartphone device or emulator.

You cannot 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	Datatype	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.
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 PowerObject containing information about the class definition of the object or control.
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: 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.
IsLocallyScripted	Boolean	Indicates whether the script is implemented at this level in the inheritance hierarchy. Values are: 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

ScriptDefinition property	Datatype	Description
IsRPCFunction	Boolean	Whether this is an RPC function. Values are: TRUE – Is an RPC function FALSE – Is not an RPC function Corresponds to the RPCFUNC keyword in a declaration for a stored procedure.
IsScripted	Boolean	Whether the event has a definition but no code at any level of the collapsed inheritance hierarchy. Values are: TRUE – The event has a script at some level of the object's inheritance hierarchy FALSE – The event does not have a script Only events can be defined but not scripted. For functions, IsScripted is always TRUE.
Kind	ScriptKind	Whether the script is a function or event. Values are: ScriptEvent! ScriptFunction!
LocalVariableList	VariableDefinition	An unbounded array whose elements are VariableDefinition objects, one object per local variable. The array is empty if there are no local variables.
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 PocketBuilder functions, the filename of the DLL containing the function. The value is an empty string ("") for scripts that are not built-in PocketBuilder functions.

Functions

ScriptDefinition function	Datatype 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.

SerialGPS object

The SerialGPS object provides an interface to Bluetooth unit global positioning system (GPS) devices manufactured by Pharos and TomTom. The SerialGPS object inherits from the GPS base class.

In addition to the functions listed, this object inherits functions from the NonVisualObject object.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

SerialGPS property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
ConfigParams	String	Customization property that is currently undefined.
DeviceHandle	UnsignedLong	Read-only value containing a device-specific handle.
RawData	String	Read-only, raw, NMEA-formatted data without any parsing for validity.
SerialPort	String	Specifies the serial port used by the GPS device. For Bluetooth devices, the value of this property is typically COM8.
Vendor	String	Customization property that is currently undefined.

Events

SerialGPS event	Occurs
Constructor	When the object is created
Destructor	When the object is destroyed

Functions

SerialGPS function	Datatype returned	Description
Close	Integer	Closes a GPS communications channel if one is open and deactivates the data handlers.
GetFix	GPSTFix	Populates the GPSTFix structure with values from the current fix.
GetHeading	GPSTHeading	Populates the GPSTHeading structure with values from the current heading.
GetSatellitesInView	GPSSatellitesInView	Populates the GPSSatellitePosition and GPSSatellitesInView structures with position information from the current satellites in view.
Open	Integer	Initializes GPS data handlers and opens a communications channel. You can use an optional string argument to force-feed a single set of GPS data without opening a communications channel. If this argument is used, the data acquisition behavior of derived classes (SerialGPS) remains undefined.

Signature control

The Signature visual control is typically used for the capture of user signatures, but can also be used to capture freehand drawing and typed input.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

Signature property	Datatype	Description
BackColor	Long	Specifies the numeric value of the background color, with valid values from -2 to 16,777,215.
Border	Boolean	Specifies whether the control has a border. Values are: - TRUE – Control has a border - FALSE – Control does not have a border

Signature property	Datatype	Description
BorderStyle	BorderStyle (enumeration)	Specifies the style of the border of the control. Values are: • StyleBox! • StyleLowered! • StyleRaised!
BringToTop	Boolean	Specifies whether PocketBuilder moves the control to the front of the window.
Enabled	Boolean	Specifies whether the control is visible or grayed. Values are: • TRUE – Control is visible • FALSE – Control is grayed
Height	Integer	Specifies the height of the control, in PowerBuilder units.
PageStyle	SigPageStyle (enumeration)	Specifies the style of the control, which can be one of the following values: • SIGPageStyleNone! • SIGPageStyleDottedLines! • SIGPageStyleGridLines! • SIGPageStyleLeftMargin! • SIGPageStyleRuledLines! (default) • SIGPageStyleTopMargin! • SIGPageStyleTopLeftMargin! • SIGPageStyleYellowBackground!
PenMode	SigPenMode (enumeration)	Specifies the drawing mode. Values are: • SIGPenModePen! – User can write or draw (default) • SIGPenModeSelect! – User can select text • SIGPenModeSpace! – User can collapse and expand white space
Tag	String	Specifies the tag value assigned to the control.
Tap_And_Hold_Indicator	Boolean	Specifies whether red or blue dots in a circular animation display to confirm that the user has performed a tap and hold action with the stylus on the control. This action on a Pocket PC simulates a right mouse-click. The property is enabled by default to conform with typical Pocket PC behavior.
ViewStyle	SigViewStyle (enumeration)	Specifies the type of user input expected. Values are: • SIGViewStyleWriting! – Words in user input are converted to text • SIGViewStyleTyping! – User inputs data using the SIP • SIGViewStyleDrawing! – No attempt is made to convert user input into text and the SIP is disabled

Signature property	Datatype	Description
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.
WrapMode	SigWrapMode (enumeration)	Specifies whether text is wrapped to the window or page. Values are: • SigWrapModePage! – Wraps text to page • SigWrapModeWindow! – Wraps text to window (default)
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.
ZoomPercent	Integer	Specifies the zoom percent for the control. The default value is 100.

Events

Signature event	Occurs
Clicked	Immediately after the control is tapped or clicked
Constructor	Immediately before the Open event occurs in the window
Destructor	Immediately after the Close event occurs in the window
GetFocus	Just before the control receives focus (before it is selected and becomes active)
LoseFocus	When the control loses focus (becomes inactive)
RButtonDown	When the user performs a tap and hold action on the control

Functions

Signature function	Datatype returned	Description
Clear	Integer	Clears the contents of the control.
GetDataAsBitmap	Integer	Retrieves the data in the control as a standard Windows bitmap that is compatible with the Picture control and Windows desktop applications.
GetDataAsInk	Integer	Retrieves the data in the control in Pocket Word Ink (PWI) format. This format is compatible with Pocket Word.

Signature function	Datatype returned	Description
GetDataAsRTF	Integer	Retrieves the text data in the control as an RTF ANSI text block in a blob or Unicode string. This function does not return graphic signature data.
GetDataAsText	Integer	Retrieves the text data in the control as a string. This function does not return graphic signature data.
SetDataAsInk	Integer	Sets data in Pocket Word Ink (PWI) format in the control. This format is compatible with Pocket Word.
SetDataAsRTF	Integer	Sets text data in the control as an RTF ANSI text block in a blob or Unicode string. This function does not set graphic signature data.
SetDataAsText	Integer	Sets the text data in the control as a string. This function does not return graphic signature data.
SetFocus	Integer	Sets focus to the control.

SimpleTypeDefinition object

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

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Windows CE platforms

This object can be used only in the development environment. It cannot be used in applications deployed to a Pocket PC or Smartphone device or emulator.

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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

SingleLineEdit property	Datatype	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: 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 RGB in 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
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PocketBuilder moves 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
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.

SingleLineEdit property	Datatype	Description
DisplayOnly	Boolean	Specifies whether the text in the control is display-only and cannot be changed by the user. Values are: TRUE – Text cannot be changed by user FALSE – Text can be changed by user
DragAuto	Boolean	Specifies whether PocketBuilder puts 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the <i>ICO</i> file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected). Values are: TRUE – Control can be selected FALSE – Control cannot be selected
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!

SingleLineEdit property	Datatype	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
InputEditMode	Integer	Specifies the input method edit mode. In PocketBuilder applications, you can use this property to set the SIP type on Pocket PC devices or the keypad entry mode on Smartphone devices.
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).
Password	Boolean	Specifies whether the control is a password field (whether asterisks appear when the user types characters). Values are: 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 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
SIPOnFocus	Boolean	Whether to display the SIP when the control receives focus and minimize the SIP when the control loses focus. Values are: Yes — SIP is opened and closed automatically. No — SIP is not opened or closed automatically. Painter: Show SIP on Focus option.
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.

SingleLineEdit property	Datatype	Description
TextCase	TextCase (enumerated)	Specifies the case in which text entered in the control displays. Values are: AnyCase! Lower! Upper!
TextColor	Long	Specifies the numeric value of the color used for text: -2 to 16,777,215. For more information about color, see the RGB function in 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: TRUE – Text is underlined FALSE – Text is not underlined
Visible	Boolean	Specifies whether the control is visible. Values are: 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
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
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 PocketBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

Functions

SingleLineEdit function	Datatype 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.
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.

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

SMSAddress object

The SMSAddress object is a system structure that stores the address and address type of an SMSMessage object. The SMSAddress object has no events. It inherits functions from the PowerObject base object.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

SMSAddress property	Datatype	Description
Address	String	Identifies an address for the SMS message.
AddressType	SMSAddrType	Specifies the SMS address type. Values are: • smsat_abbreviated! • smsat_alphanumeric! • smsat_international! • smsat_national! • smsat_networkspecific! • smsat_subscriber! • smsat_unknown!
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.

SMSMessage object

The SMSMessage object is a system structure that stores the text of an SMSMessage object. The SMSMessage object has no events. It inherits functions from the PowerObject base object.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

SMSMessage property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
ID	Long	Identifies an SMS message.
Options	SMSMsgOptions	Indicates the retry options if an SMS message is not delivered by the message router. Values are: • SMS_OPTION_DELIVERY_NONE! No special handling specified. • SMS_OPTION_DELIVERY_NO_RETRY! No attempt to resend the message will be made. When this option is not specified, the message is resent according to a short-term retry schedule.
Status	SMSMsgStatus	Specifies the SMS message status.
Text	String	Text data portion of the message; this can be null.
ValidityPeriod	DateTime	Defines the duration for which the SMS message is valid, beginning when the SMS service center receives the message; this can be null.

SMSProtocol object

The SMSProtocol object is a system structure that stores the data encoding and message mode of an SMS message. The SMSProtocol object has no events. It inherits functions from the PowerObject base object.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

SMSProtocol property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
Encoding	SMSDataEncoding	Indicates the text encoding of the SMS message. Values are: • SMSDE_OPTIMAL! (recommended for most purposes) • SMSDE_GSM! • SMSDE_UCS2!
Mode	SMSMessageModes	Indicates whether the message mode is send or receive. Values are: • SMSMODERECEIVE! (not implemented) • SMSMODESEND!
ProtocolType	SMSProtocolType	Indicates which SMS protocol to use. Values are: • SMS_MSGTYPE_TEXT! • SMS_MSGTYPE_NOTIFICATION! • SMS_MSGTYPE_WDP! • SMS_MSGTYPE_WCMP! • SMS_MSGTYPE_STATUS! • SMS_MSGTYPE_BROADCAST! • SMS_MSGTYPE_RAW!

SMSProviderSpecificData object

The SMSProviderSpecificData object is a system structure that stores provider-specific information for an SMS message. It inherits functions from the PowerObject base object.

PocketBuilder	×
PowerBuilder	×

Object not implemented

The SMSProviderSpecificData object and its properties are not implemented in the current release of PocketBuilder. They are reserved for future use.

Properties

SMSPProviderSpecificData property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control
ClassSpecificValue	Any	Specifies values associated with the MessageClass
MessageClass	SMSPProviderMsgClass	Specifies how messages should be handled when calling the Send function on an open SMSSession object
MessageOptions	Long	Specifies provider-specific options

SMSSession object

The SMSSession object provides the interface for a PocketBuilder application and the SMS messaging system on a Pocket PC or SmartPhone device. It opens and closes an SMS message session and has functions that let you send and get the status of an SMS message. It also provides access to the SMSAddress, SMSMessage, and SMSProtocol objects.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

SMSSession property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.

Events

SMSSession event	Occurs
Constructor	When the object is created
Destructor	When the object is destroyed
IncomingMessage	When an SMS message is received

Functions

SMSSession function	Datatype returned	Description
Close	Integer	Closes the SMS session.
GetMessageStatus	Integer	Checks the status of an SMS message sent during the current session.
Open	Integer	Opens the SMS session for send access. Read access will be supported in a future release.
Send	Integer	Sends an SMS message.

SocketBarcodeScanner object

The SocketBarcodeScanner object provides an interface to Socket SDIO Series Barcode scanners.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Properties

SocketBarcodeScanner property	Datatype	Description
ScannerName	String	Name of the scanner you are using
ScannedData	String	Read-only data in the scanner's internal buffers at the time of the last RetrieveData call

SocketBarcodeScanner property	Datatype	Description
ScannedSymbology	Integer	Read-only value with the decoder ID representing the symbology used by the last data read by the scanner
ScannedTimeStamp	DateTime	Read-only timestamp of the last scan

Events

SocketBarcodeScanner event	Occurs
Constructor	When the object is created
Destructor	When the object is destroyed
ScannerInserted	When the interface layer first recognizes a physical scanner: either on initialization of the scanner or on physical insertion of the scanner
ScannerRemoved	When the scanner is physically removed
ScanTriggered	When asynchronous data is scanned

Functions

SocketBarcodeScanner function	Datatype returned	Description
Close	Integer	Optional function that clears all buffers, detaches from the scanner firmware, and unloads all scanning DLLs. By default, this function is called by the BarcodeScanner object destructor.
DecoderName	String	Retrieves the short decoder name for the ID value passed as a function argument.
DeviceInfo	Integer	Retrieves device-specific settings, such as version numbers.
DeviceNames	Integer	Sets the names of the scanning devices.
EnableDecoder	Integer	Enables or disables the decoder whose ID value is passed in a function argument.
Flush	Integer	Flushes any old results in the scan buffers.
GetEnabledDecoders	Integer	Gets the array of enabled decoders.
GetSupportedDecoders	Integer	Gets the array of supported decoders.
Open	Integer	Loads the scanning DLLs and connects to the scanner firmware. This is typically the first function called on an object instance.
RetrieveData	Integer	Retrieves the data from the scanner internal buffers.
ScanAbort	Integer	Aborts all outstanding scan requests.

SocketBarcodeScanner function	Datatype returned	Description
ScanNoWait	Integer	Sets the scan for asynchronous operation and an immediate return of scan data. In a typical implementation of this function, the ScanTriggered event sets the rearm flag to “true” for continuous scanning.
ScanWait	Integer	Sets the timeout period for a synchronous scan. The timeout period is the amount of time allowed to elapse before a scan resumes following a pause in the scanning.
SoftTrigger	Boolean or Integer	Sets or retrieves the soft trigger feature of a scanner.
Status	Integer	Returns the scanner status as an integer.

SSLCallBack object

The SSLCallBack object provides PowerBuilder clients with the ability to handle SSL callbacks from EAServer.

PocketBuilder	✗
PowerBuilder	✓

SSLServiceProvider object

The SSLServiceProvider object allows you to establish a Secure Sockets Layer (SSL) connection from a PowerBuilder client to EAServer.

PocketBuilder	✗
PowerBuilder	✓

StaticHyperLink control

The StaticHyperLink control is a descendant of the StaticText control. The URL property of the StaticHyperLink control enables you to provide a hot link to a web page. When the user clicks the control, the user's web browser opens to display the page you specify.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Usage note

If you know that your users have browsers that support URL completion, you can enter a partial address, such as `sybase.com`. You can always enter a complete address, such as `http://www.sybase.com`.

Properties

StaticHyperLink property	Datatype	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 RGB function in 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.
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!

StaticHyperLink property	Datatype	Description
BringToTop	Boolean	Specifies whether PocketBuilder 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
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DisabledLook	Boolean	Specifies whether the control appears to be enabled.
DragAuto	Boolean	Specifies whether PocketBuilder puts 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected). Values are: TRUE – Control can be selected FALSE – Control cannot be selected
FaceName	String	Specifies the name of the typeface in which the text of the control will display (for example, ARIAL or COURIER).
FillPattern	FillPattern (enumerated)	Specifies the hatch pattern used to fill the control. 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.

StaticHyperLink property	Datatype	Description
FocusRectangle	Boolean	Specifies whether a dotted rectangle (focus rectangle) frames the control when it has focus. Values are: TRUE – Control framed when it has focus FALSE – Control not framed when it has focus
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 rectangular box that contains 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 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 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 is the default and means the user cannot tab to the control).

StaticHyperLink property	Datatype	Description
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 RGB function in 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: TRUE – Text is underlined FALSE – Text is not underlined
URL	String	Specifies the URL to open in the user's web browser when the text control is clicked, provided no clicked event is coded. The status text displays the URL when the mouse passes over the control.
Visible	Boolean	Specifies whether the control is visible. Values are: 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

StaticHyperLink 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.

StaticHyperLink 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).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PocketBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

Functions

StaticHyperLink function	Datatype 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.
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.

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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

StaticText property	Datatype	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 RGB function in 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.
BorderStyle	BorderStyle (enumerated)	Specifies the style of the border of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PocketBuilder 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
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DisabledLook	Boolean	Specifies whether the control appears to be enabled.

StaticText property	Datatype	Description
DragAuto	Boolean	Specifies whether PocketBuilder puts 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the <i>ICO</i> file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected). Values are: TRUE – Control can be selected FALSE – Control cannot be selected
FaceName	String	Specifies the name of the typeface in which the text of the control will display (for example, ARIAL or COURIER).
FillPattern	FillPattern (enumerated)	Specifies the hatch pattern used to fill the control. 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.
FocusRectangle	Boolean	Specifies whether a dotted rectangle (focus rectangle) frames the control when it has focus. Values are: TRUE – Control framed when it has focus FALSE – Control not framed when it has focus

StaticText property	Datatype	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 rectangular box that contains 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 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 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 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 RGB function in the <i>PowerScript Reference</i> .

StaticText property	Datatype	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: TRUE – Text is underlined FALSE – Text is not underlined
Visible	Boolean	Specifies whether the control is visible. Values are: 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.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PocketBuilder event.
RButtonDown	When the right mouse button is pressed on the control.

Functions

StaticText function	Datatype 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.
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.

SymbolBarcodeScanner object

The SymbolBarcodeScanner object implements all the methods and properties of the BarcodeScanner base class. It dynamically loads Symbol support DLLs for the PPT 2800 and 8800 series scanners.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Properties

SymbolBarcodeScanner property	Datatype	Description
ScannerName	String	Name of the scanner you are using
ScannedData	String	Read-only data in the scanner's internal buffers at the time of the last RetrieveData call
ScannedSymbology	Integer	Read-only value with the decoder ID representing the symbology used by the last data read by the scanner
ScannedTimeStamp	DateTime	Read-only timestamp of the last scan

Events

SymbolBarcodeScanner event	Occurs
Constructor	When the object is created
Destructor	When the object is destroyed
ScannerInserted	When the interface layer first recognizes a physical scanner: either on initialization of the scanner or on physical insertion of the scanner
ScannerRemoved	When the scanner is physically removed
ScanTriggered	When asynchronous data is scanned

Functions

SymbolBarcodeScanner function	Datatype returned	Description
Close	Integer	Optional function that clears all buffers, detaches from the scanner firmware, and unloads all scanning DLLs. By default, this function is called by the BarcodeScanner object destructor.
DecoderName	String	Retrieves the short decoder name for the ID value passed as a function argument.
DeviceInfo	Integer	Retrieves device-specific settings, such as version numbers.
DeviceNames	Integer	Sets the names of the scanning devices.
EnableDecoder	Integer	Enables or disables the decoder whose ID value is passed in a function argument.
Flush	Integer	Flushes any old results in the scan buffers.
GetEnabledDecoders	Integer	Gets the array of enabled decoders.

SymbolBarcodeScanner function	Datatype returned	Description
GetSupportedDecoders	Integer	Gets the array of supported decoders.
Open	Integer	Loads the scanning DLLs and connects to the scanner firmware. This is typically the first function called on an object instance.
RetrieveData	Integer	Retrieves the data from the scanner internal buffers.
ScanAbort	Integer	Aborts all outstanding scan requests.
ScanNoWait	Integer	Sets the scan for asynchronous operation and an immediate return of scan data. In a typical implementation of this function, the ScanTriggered event sets the rearm flag to “true” for continuous scanning.
ScanWait	Integer	Sets the timeout period for a synchronous scan. The timeout period is the amount of time allowed to elapse before a scan resumes following a pause in the scanning.
SoftTrigger	Boolean or Integer	Sets or retrieves the soft trigger feature of a scanner.
Status	Integer	Returns the scanner status as an integer.

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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

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	Datatype	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 in the <i>PowerScript Reference</i> .
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 PocketBuilder 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
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.

Tab property	Datatype	Description
CreateOnDemand	Boolean	Specifies whether PocketBuilder creates graphical representations of controls on all tab pages when the Tab control is created. Values are: 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.
Control[]	UserObject	Specifies the array of tab pages within the Tab control.
DragAuto	Boolean	Specifies whether PocketBuilder 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Enabled	Boolean	Specifies whether the control is enabled (can be selected). Values are: TRUE – Can be selected FALSE – Cannot be selected
FaceName	String	Specifies the name of the typeface used for the text labels on tabs (for example, ARIAL or COURIER). Only TrueType fonts display correctly on vertical tabs.
FixedWidth	Boolean	Specifies whether tabs have a fixed width, meaning they do not shrink to the length of their text labels. Values are: 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

Tab property	Datatype	Description
FocusOnButtonDown	Boolean	Specifies whether each tab gets focus when the user clicks on it. Values are: TRUE – The tab the user clicks on gets focus; a dotted rectangle marks the tab FALSE – The tab does not get focus In either case, the selected tab page comes to the front. 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.
FontCharSet	FontCharSet (enumerated)	Specifies the character set for the text labels on the tabs. Values are: ANSI! ChineseBig5! DefaultCharset! HangEul! OEM! Shiftjis! Symbol!
FontFamily	FontFamily (enumerated)	Specifies the font family (type style) for the text labels on the tabs. Values are: AnyFont! Decorative! Modern! Roman! Script! Swiss!
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

Tab property	Datatype	Description
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 do not 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.
PowerTips	Boolean	Specifies whether PowerTipText for a tab page is displayed as a PowerTip (a pop-up label for the tab) when the mouse pointer pauses over the tab. PowerTips are useful if the tabs are labeled with pictures. Values are: TRUE – PowerTipText, if any, displays as a pop-up label for each tab FALSE – No PowerTips are displayed
RaggedRight	Boolean	Specifies whether tabs are stretched so that they fill space along the edge of the control. Values are: TRUE – Tabs remain the size determined by their label text and the FixedWidth property FALSE – Tabs are stretched to fill the edge
SelectedTab	Integer	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
ShowPicture	Boolean	Specifies whether the picture selected for each tab is displayed. Values are: TRUE – The picture for each tab, if any, is displayed FALSE – No pictures are displayed

Tab property	Datatype	Description
ShowText	Boolean	Specifies whether the text specified for each tab label is displayed. Values are: TRUE – The text for each tab is displayed FALSE – The text for each tab is not displayed
TabOrder	Integer	Specifies the tab value of the control within the window (0 means the user cannot tab to the control).
TabPosition	TabPosition (enumerated)	Specifies where the tabs appear around the Tab control. Values are: • 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	Specifies whether the text on the tabs is underlined. Values are: TRUE – Text labels are underlined FALSE – Text labels are not underlined
Visible	Boolean	Specifies whether the control is visible. Values are: 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.
DoubleClick	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.
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).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
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 PocketBuilder 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.
RightDoubleClick	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 RightDoubleClick 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	Datatype 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.
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.
SelectTab	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.
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.

Throwable object

The Throwable datatype is the base class for all throwable objects. These include exceptions and error objects.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

Throwable property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
Text	String	Contains the text of the error message.

Events

Throwable event	Occurs
Constructor	When the exception is thrown.
Destructor	Immediately after the exception is thrown.

Functions

Throwable function	Datatype 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.
GetMessage	String	Returns the error message from objects of type Throwable.
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.
SetMessage	—	Sets an error message for an object of type Throwable.

Throwable function	Datatype returned	Description
TriggerEvent	Integer	Triggers a specific event in the object and executes the script for the event.
TypeOf	Object	Returns the type of the object.

Timing object

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.

PocketBuilder	✗
PowerBuilder	✓

Toolbar control

The Toolbar control is typically used as a visual extension to a Pocket PC application menu.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

Toolbar property	Datatype	Description
Enabled	Boolean	Specifies whether the control is enabled. Values are: • TRUE – (Default) Control is enabled. • FALSE – Control is disabled. Tapping items in control causes device to beep.
Height	Integer	Specifies the height of the control, in PowerBuilder units.

Toolbar property	Datatype	Description
Item[]	String array	Specifies the initial item strings in the ToolbarItem portion of the Toolbar control. After initialization, this array can be updated only through function calls, such as AddItem or DeleteItem.
ItemCount	Integer	Specifies the number of toolbar items in the Toolbar control. After initialization, this property can be updated only through function calls, such as AddItem or DeleteItem.
PictureHeight	Integer	Specifies the height of the picture, in pixels. This property can be set only 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.
PictureName[]	String array	Specifies the picture name for each item in the Item property array. After initialization, this array can be updated only through the AddPicture function call.
PictureWidth	Integer	Specifies the width of the picture, in pixels. This property can 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.
Position	ToolbarAlignment (enumeration)	Specifies whether the toolbar is docked at the top or the bottom. Values are: • AlignAtBottom! • AlignAtTop! • Floating!
Visible	Boolean	Specifies whether the control is visible. Values are: • TRUE – (Default) Control is visible • FALSE – Control is not visible
Width	Integer	Read-only property that specifies the width of the control, in PowerBuilder units.
Wrap	Boolean	Specifies whether toolbar items wrap onto a second line. Values are: • TRUE – (Default) Toolbar items wrap • FALSE – Toolbar items do not wrap
X	Integer	Read-only property that specifies the X position (the distance from the left edge of the parent window), in PowerBuilder units.
Y	Integer	Read-only property that specifies the Y position (the distance from the top of the parent window), in PowerBuilder units.

Events

ToolBar event	Occurs
Clicked	Immediately after a toolbar item is tapped or clicked

Functions

ToolBar function	Datatype returned	Description
AddItem	Integer	Adds a toolbar item to the toolbar
AddPicture	Integer	Adds a picture to the toolbar picture array
DeleteItem	Integer	Removes a toolbar item from the toolbar
Destroy	Integer	Removes the toolbar
GetItem	Integer	Returns a toolbar item
GetItemPictureIndex	Integer	Returns the picture index that corresponds to the item index for the toolbar item
GetItemState	Integer	Returns the state of a toolbar item
InsertItem	Long	Inserts a toolbar item in the toolbar
SetItemPictureIndex	Integer	Specifies the picture to be used for a toolbar item
SetItemState	Boolean	Sets the state of a toolbar item

ToolBarItem object

You can add toolbar items to a toolbar on the Items page of the Properties view for a Toolbar object. You must enter a number for the item in the CurrentBarItemIndex text box, then select or clear the ItemEnabled check box for the item and assign it an enumerated value for ItemStyle.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Properties

ToolStripItem property	Datatype	Description
ItemGroup	Integer	Specifies the group of items to which a toolbar item belongs. By default a toolbar item does not belong to a group, in which case its ItemGroup value is 0. If you select stylecheckgroup! for the ItemStyle property of a toolbar item in the Toolbar control Properties view, the ItemGroup value for that item defaults to 1.
ItemPictureIndex	Integer	Specifies the picture you want to associate with a toolbar item. You can add pictures to the picture array in the Toolbar control Properties view or by using the AddPicture function. You can also change the picture associated with a toolbar item by calling SetItemPictureIndex on the Toolbar control. The ItemPictureIndex property is similar to the PictureIndex property on a ListViewItem object.
ItemState	Integer	Specifies the state of the ToolStrip item.
ItemStyle	ToolStripItemStyle (enumeration)	Specifies the style of the ToolStripItem. Values are: • StyleButton! Specifies a standard button that automatically returns to the up position after a user taps it. • StyleCheck! Specifies a button that remains depressed when a user taps it. The user must tap the button a second time to raise it. • StyleCheckGroup! Specifies a group of buttons. A button of this style remains depressed after a user taps it, but returns to the up position when the user taps it again or taps a toolbar button of the same style. • StyleSeparator! Specifies a separator bar.

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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceActivityNode object has no events.

Properties

TraceActivityNode property	Datatype	Description
ActivityType	TraceActivity (enumerated)	A value of the enumerated datatype TraceActivity that identifies the activity represented by the node. Values are: 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
Category	TraceCategory (enumerated)	The category of the activity represented by the node. Values are: 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.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceActivityNode function	Datatype 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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceBeginEnd object has no events.

Properties

TraceBeginEnd property	Datatype	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: 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.

TraceBeginEnd property	Datatype	Description
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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	Datatype 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.

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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceError object has no events.

Properties

TraceError property	Datatype	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.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
Message	String	The system error message or the message passed to the TraceError function.
Severity	Long	The system error severity or the severity argument passed to the TraceError function.
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceError function	Datatype 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.

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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceESQL object has no events.

Properties

TraceESQL property	Datatype	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: 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.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
Name	String	The name of the ESQL statement.
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceESQL function	Datatype 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.

TraceFile object

The TraceFile object is used to access the contents of a trace file created from a PocketBuilder 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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceFile object has no events.

Properties

TraceFile property	Datatype	Description
ApplicationName	String	The name of the application used to generate the trace file.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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, that is, the proper amount of time has been subtracted from the timestamps 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	Datatype 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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceGarbageCollect object has no events.

Properties

TraceGarbageCollect property	Datatype	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: 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.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceGarbageCollect function	Datatype 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.

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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceLine object has no events.

Properties

TraceLine property	Datatype	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.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
LineNumber	UnsignedLong	The line number.
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceLine function	Datatype 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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceObject object has no events.

Properties

TraceObject property	Datatype	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: 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.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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.
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceObject function	Datatype 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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceRoutine object has no events.

Properties

TraceRoutine property	Datatype	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: 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.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.

TraceRoutine property	Datatype	Description
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 datatypes and return value.
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	Datatype 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 PocketBuilder 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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceTree object has no events.

Properties

TraceTree property	Datatype	Description
ApplicationName	String	The name of the application used to generate the trace file.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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	Datatype 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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceTreeError object has no events.

Properties

TraceTreeError property	Datatype	Description
ActivityType	TraceActivity (enumerated)	The value ActError! which identifies the activity represented by the node as an occurrence of a system error or warning.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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 severity argument passed to the TraceError function.
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

TraceTree-Error function	Datatype 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.

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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceTreeESQL object has no events.

Properties

TraceTreeESQL property	Datatype	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.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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	Datatype 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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceTreeGarbageCollect object has no events.

Properties

TraceTreeGarbageCollect property	Datatype	Description
ActivityType	TraceActivity (enumerated)	The value ActGarbageCollect! which identifies the activity represented by the node as garbage collection start and finish.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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	Datatype 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.

TraceTreeGarbage Collect function	Datatype 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.

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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceTreeLine object has no events.

Properties

TraceTreeLine property	Datatype	Description
ActivityType	TraceActivity (enumerated)	The value ActLine! which identifies the activity represented by the node as an occurrence of a routine line hit.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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	Datatype 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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceTreeNode object has no events.

Properties

TraceTreeNode property	Datatype	Description
ActivityType	TraceActivity (enumerated)	A value of the enumerated datatype TraceActivity that identifies the activity represented by the node. Values are: 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
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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	Datatype 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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceTreeObject object has no events.

Properties

TraceTreeObject property	Datatype	Description
ActivityType	TraceActivity (enumerated)	The value ActObjectCreate! or ActObjectDestroy! which identifies the activity represented by the node as object creation or destruction.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
ClassName	String	The name of the class that is the object type.
EnterTimerValue	Decimal	The timer value (in seconds) when the activity began.
ExitTimerValue	Decimal	The timer value (in seconds) when the activity ended.
IsCreate	Boolean	TRUE if the node represents the creation of an object and FALSE if it represents the destruction of an object.
ObjectID	UnsignedLong	The internal identifier for the object.
ParentNode	TraceTreeNode	The parent of this node.

Functions

TraceTreeObject function	Datatype returned	Description
ClassName	String	Returns the name assigned to the object.

TraceTreeObject function	Datatype returned	Description
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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceTreeRoutine object has no events.

Properties

TraceTreeRoutine property	Datatype	Description
ActivityType	TraceActivity (enumerated)	The value ActRoutine! which identifies the activity represented by the node as an occurrence of a routine entry and exit.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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.

TraceTreeRoutine property	Datatype	Description
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 datatypes 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.
ParentNode	TraceTreeNode	The parent of this node.

Functions

TraceTreeRoutine function	Datatype 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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceTreeUser object has no events.

Properties

TraceTreeUser property	Datatype	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.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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	Datatype 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.

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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

The TraceUser object has no events.

Properties

TraceUser property	Datatype	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.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
Message	String	The message passed to the TraceUser function.
TimerValue	Decimal	The timer value (in seconds) when the activity occurred.

Functions

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

Transaction object

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

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

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

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

Properties

Transaction property	Datatype	Description
AutoCommit	Boolean	The automatic commit indicator (SQL Server only). Values are: TRUE – Commit automatically after every database activity FALSE – Do not commit automatically after every database activity
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
Database	String	The name of the database with which you are connecting.
DBMS	String	PocketBuilder 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.

Transaction property	Datatype	Description
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.
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 might 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	Datatype returned	Description
ClassName	String	Returns the name assigned to the user object.
DBHandle	Long	Returns the handle for your DBMS.
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.

TransactionServer object

The TransactionServer object provides information about the current transaction context and enables a component running in a transaction server to control the transaction and its own lifecycle.

PocketBuilder	✗
PowerBuilder	✓

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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Smartphone platforms

Arrow keys on a Smartphone device navigate within the tree view, but you must program a menu item or soft key to move the focus from the tree view to another control in the same main window.

Properties

TreeView property	Datatype	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. For more information about color, see the RGB function in 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

TreeView property	Datatype	Description
BorderStyle	BorderStyle (enumerated)	Specifies the border style of the control. Values are: StyleBox! StyleLowered! StyleRaised! StyleShadowBox!
BringToTop	Boolean	Specifies whether PocketBuilder moves the control to the top of the front-to-back order. Values are: TRUE – Control is on top of other controls FALSE – Control is not on top of other controls
CheckBoxes	Boolean	Specifies whether the state images are replaced by check boxes. The check boxes are set to unchecked by default. The TreeView control will process mouse and keyboard input to toggle the checked state. Values are: TRUE – Check boxes are displayed. FALSE – Check boxes are not displayed. The state of an item's check box can be determined by checking the state picture index for the item: Unchecked = 1 Checked = 2
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DeleteItems	Boolean	Specifies whether the user can delete a TreeView item from a TreeView control using the Delete key. Values are: TRUE – The user can delete items from the control FALSE – The user cannot delete items from the control
DisableDragDrop	Boolean	Disable Drag Drop determines whether events for dragging, such as BeginDrag, are triggered when the user clicks on an item and drags. Values are: TRUE – Drag events are not triggered FALSE – Drag events are triggered
DragAuto	Boolean	Specifies whether PocketBuilder puts 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. In either case, DisableDragDrop must be set to FALSE for dragging to occur.

TreeView property	Datatype	Description
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the <i>ICO</i> file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
EditLabels	Boolean	Specifies whether the user can edit the item labels in a control by clicking on a selected item. Values are: TRUE – The user can edit item labels FALSE – The user can not edit item labels
Enabled	Boolean	Specifies whether the control is enabled (can be clicked). Values are: TRUE – Control can be clicked FALSE – Control cannot be clicked
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. The application must be running on an appropriate version of PocketBuilder 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!

TreeView property	Datatype	Description
FullRowSelect	Boolean	Specifies whether full row selection is enabled. Values are: TRUE – Clicking anywhere on a row causes the entire row to be selected, and selecting any item in the row causes the entire row to be highlighted. FALSE – Selecting one item in a row does not cause the entire row to be highlighted or selected. This property cannot be used in conjunction with <code>HasLines = TRUE</code>.
HasButtons	Boolean	Specifies whether TreeView parent items have + and - buttons associated with them to indicate if they are expanded (-) or collapsed (+). Values are: TRUE – Parent items have buttons FALSE – Parent items do not have buttons Specifies whether only the selected TreeView item is allowed to be expanded. A single mouse click selects an item. TRUE – When an item is selected, it is automatically expanded and the previously selected item is automatically collapsed. FALSE – More than one item can be expanded at a time. If <code>SingleExpand</code> is set to <code>TRUE</code> to specify that only one item can be expanded, more than one item can be expanded by clicking on the item buttons if <code>HasButtons</code> is also <code>TRUE</code>.
HasLines	Boolean	Specifies whether TreeView items will be connected by lines. Values are: TRUE – Items will be connected by lines FALSE – Items will not be connected by lines
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
Indent	Integer	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. 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 <code>PictureWidth</code> to 0 to remove extra space.

TreeView property	Datatype	Description
InputEditMode	Integer	Specifies the input method edit mode. In PocketBuilder applications, you can use this property to set the SIP type on Pocket PC devices or the keypad entry mode on Smartphone devices.
Italic	Boolean	Specifies whether the text in the control is italic. Values are: TRUE – Text is italic FALSE – Text is not italic
LinesAtRoot	Boolean	Specifies whether PocketBuilder will connect TreeView root items with lines. Values are: TRUE – TreeView control connects root items with lines when HasLines is also TRUE FALSE – Control does not connect root items with lines
PictureHeight	Integer	Specifies the size, in pixels, for the height of the TreeView item picture. In a script, this value can only be set before a picture has been added to the picture index list. If the picture height is 0, PocketBuilder uses the height of the first picture added to the picture index list.
PictureMaskColor	Long	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 can be changed between adds.
PictureName	String	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 PocketBuilder library. Not updated after initialization.
PictureWidth	Integer	Specifies the size, in pixels, for the width of the TreeView item picture. In a script, this value can only be set before a picture has been added to the picture index list. If the picture width is 0, PocketBuilder uses the width of the first picture added to the picture index list.
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
SingleExpand	Boolean	Specifies whether only the selected TreeView item is allowed to be expanded. A single mouse click selects an item. TRUE – When an item is selected, it is automatically expanded and the previously selected item is automatically collapsed. FALSE – More than one item can be expanded at a time. Note that if HasButtons = TRUE, more than one item can be expanded at a time by clicking on the item buttons.

TreeView property	Datatype	Description
SortType	grSortType	Selects the sort method. Values are: Ascending! – Alphabetic by label Descending! – Reverse-alphabetic by label UserDefinedSort! – According to the script in the Sort event Unsorted! – Not sorted When SortType specifies sorting, sorting happens automatically. For Unsorted!, you can call functions for alphabetic sorting.
StatePictureHeight	Integer	Specifies the size, in pixels, for the height of the state picture. In a script, this value can only be set before a state picture has been added to the state picture list. If the state picture height is 0, PocketBuilder uses the height of the first picture added to the state picture index list.
StatePictureMaskColor	Long	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 can be changed between adds.
StatePictureName[]	String	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. The picture can be an icon, cursor, or bitmap supplied by the user or a stock picture from the PocketBuilder library. Not updated after initialization.
StatePictureWidth	Integer	Specifies the size in pixels for the width of the state picture. In a script, this value can only be set before a state picture has been added to the state picture list. If the state picture width is 0, PocketBuilder uses the width of the first picture added to the state picture index list.
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.
TextColor	Long	Specifies the numeric value of the color used for text: -2 to 16,777,215.
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.
ToolTips	Boolean	Specifies whether the an item's label should be displayed in a tooltip if the label is cut off by the right edge of the control. Values are: TRUE – Tooltips displaying the label text are displayed when the label is cut off on the right side. This is the default. FALSE – Tooltips are not displayed.

TreeView property	Datatype	Description
TrackSelect	Boolean	Specifies whether items appear in a different color when the mouse moves over them (hot tracking). Values are: TRUE – An item changes color when the mouse moves over it FALSE – An item does not change color when the mouse moves over or pauses on it
Underline	Boolean	Specifies whether the text in the control is underlined. Values are: TRUE – Text is underlined FALSE – Text is not underlined
Visible	Boolean	Specifies whether the control is visible. Values are: 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

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.

TreeView event	Occurs
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.
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).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
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 PocketBuilder 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	Datatype 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.
GetItemAtPointer	Integer	Gets the handle or the index of the item under the cursor.
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.

TreeView function	Datatype returned	Description
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.
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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

TreeViewItem property	Datatype	Description
Bold	Boolean	Specifies whether the item is bold. Values are: TRUE – The item is bold FALSE – The item is not bold
Children	Boolean	Specifies whether the item has children. Values are: TRUE – The item has children FALSE – The item does not have children 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.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
CutHighLighted	Boolean	Specifies whether the item is the target of a cut operation. Values are: 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.
DropHighLighted	Boolean	Specifies whether the item is the target of a DragDrop operation. Values are: 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: TRUE – The item is expanded FALSE – The item is not expanded

TreeViewItem property	Datatype	Description
ExpandedOnce	Boolean	Specifies whether the item has been expanded at least once, also meaning the item has been populated with children. Values are: 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: 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.
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: 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	Datatype 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

TypeDefinition is used in the VariableDefinition class.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Windows CE platforms

This object can be used only in the development environment. It cannot be used in applications deployed to a Pocket PC or Smartphone device or emulator.

Information about the type of a variable. The variable can 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.

Properties

TypeDefinition property	Datatype	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 PowerObject containing information about the class definition of the object or control.

TypeDefinition property	Datatype	Description
DataTypeOf	String	The system class name or simple datatype of the variable. <code>DataTypeOf</code> is a string representation of a value of the <code>Object</code> enumerated datatype. Values are lowercase with no exclamation point. Sample values include: - window - string - any - dropdownlistbox For objects you have defined, the datatype 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 PocketBuilder as opposed to a type defined in a PKL by a user.
IsVariableLength	Boolean	Specifies whether the datatype has a fixed size. Values are: - TRUE – The datatype is variable length, meaning the datatype is a string, any, blob, or unbounded array. - FALSE – The datatype is a fixed length.
IsVisualType	Boolean	Indicates whether the type is a visual (displayable) or nonvisual type. Values are: - 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 datatype.
LibraryName	String	The fully qualified name of the library the type was loaded from. Note that the library might 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	Datatype 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 object

UserObjects are custom visual objects that you can build to supplement the standard PocketBuilder 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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

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.

Properties

UserObject property	Datatype	Description
BackColor	Long	Specifies the numeric value of the background color: -2 to 16,777,215. For more information about color, see the RGB function in the <i>PowerScript Reference</i> .
Border	Boolean	Specifies whether the object has a border. Values are: 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!
BringToTop	Boolean	Specifies whether PocketBuilder moves the object to the top of the front-to-back order of the window. Values are: TRUE – Object moved to top FALSE – Object not moved to top
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
ClassName	String	(External user objects only) Returns the name assigned to the object.

UserObject property	Datatype	Description
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” on page 569. PocketBuilder multiplies UnitsPerColumn by ColumnsPerPage to determine the number of PowerBuilder units to scroll the object horizontally when the user clicks in the scroll bar.
Control[]	WindowObject	Specifies the control’s objects. You cannot change the contents of this array in a script.
DragAuto	Boolean	Specifies whether PocketBuilder puts the object automatically into Drag mode. Values are: 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the object (the <i>ICO</i> file). The default icon is a box the size of the object. 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.
Enabled	Boolean	Specifies whether the object is enabled (can be selected). Values are: TRUE – Object can be selected FALSE – Object cannot be selected
Height	Integer	Specifies the height of the object, in PowerBuilder units.
HScrollBar	Boolean	Specifies whether a horizontal scroll bar displays. Values are: TRUE – Horizontal scroll bar displays FALSE – Horizontal scroll bar does not display
LibraryName	String	(External user objects only) The name of the dynamic-link library (DLL) that contains an external user object class.
LinesPerPage	Integer	Specifies the number of lines on a page. The default is 0 (10 lines per page). For information, see “Scrolling in windows and user objects” on page 569. PocketBuilder multiplies UnitsPerLine by LinesPerPage to determine the number of PowerBuilder units to scroll the object vertically when the user clicks in the scroll bar.

UserObject property	Datatype	Description
ObjectType	UserObjects (enumerated)	Specifies the type of user object. Valid values are: CustomVisual! ExternalVisual!
PictureMaskColor	Long	Specifies the numeric value of the color in the picture that is changed to the background color. Values can be: -2 to 16,777,215. For more information about color, see the RGB function in the <i>PowerScript Reference</i> . This property applies only when PictureName is a bitmap and only when the UserObject is a tab page.
PictureName	String	Specifies a value of the Pointer enumerated datatype 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. For more information about color, see the RGB function in the <i>PowerScript Reference</i> . This property applies only when the UserObject is a tab page.
TabTextColor	Long	Specifies the numeric value of the tab text color: -2 to 16,777,215. For more information about color, see the RGB function in the <i>PowerScript Reference</i> . This property applies only when the UserObject is a tab page.
TabOrder	Integer	Specifies tab value of the control within the user object (0 means the user cannot tab to the control).
Tag	String	Specifies the tag value assigned to the object.
Text	String	Specifies the text that displays in the object.

UserObject property	Datatype	Description
UnitsPerColumn	Integer	Specifies the number of PowerBuilder units to be scrolled right or left when a user clicks the left or right arrow in the horizontal scroll bar in a window or user object. The default is 0 (1/100 of the width of the window). To make the end of the scroll bar match the content, UnitsPerLine must be set according to the content width. For information, see “Scrolling in windows and user objects” on page 569. PocketBuilder multiplies UnitsPerColumn by ColumnsPerPage to determine the number of PowerBuilder units to scroll the window horizontally when the user clicks in the scroll bar.
UnitsPerLine	Integer	Specifies the number of PowerBuilder units to be scrolled up or down when a user clicks the up or down arrow in the vertical scroll bar in a window or user object. The default is 0 (1/100 of the window height). To make the end of the scroll bar match the content, UnitsPerLine must be set according to the content length. For information, see “Scrolling in windows and user objects” on page 569. PocketBuilder multiplies UnitsPerLine by LinesPerPage to determine the number of PowerBuilder units to scroll the window vertically when the user clicks in the scroll bar.
Visible	Boolean	Specifies whether the object is visible. Values are: TRUE – Object is visible FALSE – Object is not visible
VScrollBar	Boolean	Specifies whether a vertical scroll bar displays. Values are: TRUE – Vertical scroll bar displays FALSE – Vertical scroll bar 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.

UserObject event	Occurs
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.
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
Other	When a Windows message occurs that is not a PocketBuilder event.
RButtonDown	When the right mouse button is pressed on the object.

Functions

UserObject function	Datatype returned	Description
AddItem	Integer	Adds item to list.
ClassName	String	Returns the name assigned to the object.
CreatePage	Integer	Creates a tab page if it has not already been created.
DeleteItem	Integer	Deletes item from list.
Drag	Integer	Starts or ends the dragging 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.
Hide	Integer	Makes the object invisible.
InsertItem	Integer	Inserts item in list.
Move	Integer	Places the object in a new location specified by the X and Y arguments.
PageCreated	Boolean	Reports whether a tab page has been created.
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.

UserObject function	Datatype returned	Description
SetRedraw	Integer	Turns on or off automatic redrawing of the object after every change.
Show	Integer	Makes the object visible.
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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Windows CE platforms

This object can be used only in the development environment. It cannot be used in applications deployed to a Pocket PC or Smartphone device or emulator.

Properties

VariableCardinality Definition property	Datatype	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 PowerObject containing information about the class definition of the object or control.

Functions

VariableCardinality Definition function	Datatype 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.

PocketBuilder on Desktop	✓
PocketBuilder on Pocket PC	✗
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Windows CE platforms

This object can be used only in the development environment. It cannot be used in applications deployed to a Pocket PC or Smartphone device or emulator.

You cannot 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

VariableDefinition property	Datatype	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 PocketBuilder functions. They are shown as ellipses in the browser. For an example, see the ImportString function for DataWindow controls.
Cardinality	Variable Cardinality Definition	Cardinality information for the variable.
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
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 datatype. 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 PocketBuilder. 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!

VariableDefinition property	Datatype	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

VariableDefinition function	Datatype 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.

VProgressBar control

You can use a progress bar to indicate the progress of a lengthy operation, such as an installation program that copies a large number of files. The VProgressBar control is a vertical rectangle that fills with the system highlight color as the operation progresses.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Properties

VProgressBar property	Datatype	Description
BringToTop	Boolean	Specifies whether PocketBuilder moves the control to the top of the front-to-back order. Values are: TRUE – Control moved to top FALSE – Control not moved to top
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DragAuto	Boolean	Specifies whether PocketBuilder puts 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Height	Integer	Specifies the height of the control, in PowerBuilder units.
MaxPosition	Unsigned Integer	Specifies the value of the Position property when the progress bar is at the top of the control. This value can be different from the end of the control's range, set with the SetRange function. The default value is 100.
MinPosition	Unsigned Integer	Specifies the value of the Position property when the progress bar is at the bottom of the control. This value can be different from the start of the control's range, set with the SetRange function. The default value is 0.
Pointer	String	Specifies the name of the stock pointer or file containing the pointer used for the control.
Position	Integer	Specifies the value of the current position within the range of the control (set with the SetRange function). The control uses the range and the current position to determine the percentage of the progress bar to fill with the highlight color.
SetStep	Integer	Specifies a step increment for the progress bar. The default is 10.
SmoothScroll	Boolean	Specifies that the control displays as a smooth scrolling bar instead of the default segmented bar.

VProgressBar property	Datatype	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.
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.

Events

VProgressBar event	Occurs
Clicked	When the left mouse button is pressed on the control.
Constructor	Immediately before the Open event occurs in the window.
Destructor	Immediately after the Close event occurs in the window.
DoubleClicked	When the left mouse button is double-clicked on the 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.
GetFocus	Just before the control receives focus (before it is selected and becomes active).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
LoseFocus	When the control loses focus (becomes inactive).
Other	When a Windows message occurs that is not a PocketBuilder event.
RightClicked	When the right mouse button is pressed on the control.

Functions

VProgressBar function	Datatype 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.
OffsetPos	Integer	Moves the control's current position by the amount specified.
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.
SetRange	Integer	Sets the range of the control. The control uses the range and the current position to determine the percentage of the progress bar to fill with the highlight color.
SetRedraw	Integer	Controls automatic redrawing of the control after each change in its properties.
Show	Integer	Makes the control visible.
StepIt	Integer	Moves the control's current position by the amount specified by the value of the SetStep property.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.

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.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Usage note

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

Properties

VScrollBar property	Datatype	Description
BringToTop	Boolean	Specifies whether PocketBuilder 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
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DragAuto	Boolean	Specifies whether PocketBuilder puts 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.
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the ICO file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Height	Integer	Specifies the height of the control, in PowerBuilder units.

VScrollBar property	Datatype	Description
MaxPosition	Integer	Specifies the value of the Position property when the scroll box is at the bottom of the scroll bar.
MinPosition	Integer	Specifies the value of the Position property when the scroll box is at the top of the scroll bar.
Pointer	String	Specifies the name of the stock pointer or the file containing the pointer used for the control.
Position	Integer	Specifies the value between MinPosition and MaxPosition that indicates the position of the scroll box.
StdWidth	Boolean	Specifies whether the standard scroll bar width is used for the VScrollBar. Values are: 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.
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.

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).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
LineDown	When the down arrow of the control is clicked.

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

Functions

VScrollBar function	Datatype 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.
TypeOf	Object	Returns the type of the control.

VTrackBar control

Like a scroll bar, a trackbar is used as a scrolling control, but clicking on the trackbar slider moves it in discrete increments instead of continuously. The VTrackBar control has a series of tick marks to the right of the trackbar channel.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

To enable this control to be used properly from the keyboard, you must add code to the LineDown, LineUp, PageDown, and PageUp events. The code you add should change the slider Position property by the appropriate value and then pass the new slider position to the object or objects you associate with the trackbar control. You must code the Moved event if you want the trackbar control to pass on the slider position after the slider is dragged with a mouse.

Usage note

Use a trackbar when you want the user to select a discrete value. For example, you might use a trackbar to enable a user to select a timer interval or the size of a window.

Properties

VTrackBar property	Datatype	Description
BringToTop	Boolean	Specifies whether PocketBuilder moves the control to the top of the front-to-back order. Values are: TRUE – Control moved to top FALSE – Control not moved to top
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
DragAuto	Boolean	Specifies whether PocketBuilder puts 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.

VTrackBar property	Datatype	Description
DragIcon	String	Specifies the name of the stock icon or the file containing the icon you want to display when the user drags the control (the <i>ICO</i> file). The default icon is a box the size of the control. When the user drags the control, the icon displays when the control is over an area in which the control can be dropped (a valid drop area). When the control is over an area that is not a valid drop area, the No-Drop icon displays.
Height	Integer	Specifies the height of the control, in PowerBuilder units.
LineSize	Integer	Specifies how far the slider moves in response to keyboard input from the arrow keys.
MaxPosition	Integer	Specifies the value of the Position property when the slider is at the bottom of the control.
MinPosition	Integer	Specifies the value of the Position property when the slider is at the top of the control.
PageSize	Integer	Specifies how far the slider moves in response to keyboard or mouse input. Setting PageSize to 1 indicates moving 1 increment in the range of values.
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 slider.
Slider	Boolean	Specifies whether or not the trackbar contains a slider.
SliderSize	Integer	Specifies the size of the slider on the trackbar.
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.
TickFrequency	Integer	Specifies tick mark frequency. Setting TickFrequency to 1 indicates 1 tick mark for each increment in the trackbar range of values.
TickMarks	VTickMarks (enumerated)	Specifies where tickmarks should be displayed. Values are: VTicksOnTop! VTicksOnBottom! VTicksOnBoth! VTicksOnNeither!
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.

VTrackBar property	Datatype	Description
X	Integer	Specifies the X position (the distance from the left edge of the window), in PowerBuilder units.
Y	Integer	Specifies the Y position (the distance from the top of the window), in PowerBuilder units.

Events

VTrackBar 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).
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
LineDown	When the down arrow key is clicked.
LineUp	When the up arrow key is clicked.
LoseFocus	When the control loses focus (becomes inactive).
Moved	When the slider is moved (use the Position property to determine the new location).
Other	When a Windows message occurs that is not a PocketBuilder event.
PageDown	When the Page Down key is clicked or when mouse clicks are made below the slider in the trackbar channel.
PageUp	When the Page Up key is clicked or when mouse clicks are made above the slider in the trackbar channel.
RButtonDown	When the right mouse button is pressed on the control.

Functions

VTrackBar function	Datatype 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.

VTrackBar function	Datatype 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 control.
Print	Integer	Prints the control.
Resize	Integer	Changes the size of the control.
SelectionRange	Integer	Sets a selection range for the trackbar. When you select a range, a blue line is drawn in the channel of the trackbar and two arrows are drawn where the tickmarks are placed to indicate the beginning and end of the selection range.
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.
TriggerEvent	Integer	Triggers a specified event in the control and executes the script for the event.
TypeOf	Object	Returns the type of the control.

Window object

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

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

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.

Smartphone platforms

Because main windows on Smartphone platforms must be full screen size, PocketBuilder defaults all main windows to the 176 x 220 pixel size when you deploy an application to these platforms, ignoring any other size settings you made at design time.

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	Datatype	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 RGB function in 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 PocketBuilder 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
Center	Boolean	Causes the window to be centered when it is created or sized. Values are: TRUE – Window is centered FALSE – Window is not centered
ClassDefinition	PowerObject	An object of type PowerObject containing information about the class definition of the object or control.
ClientEdge	Boolean	Specifies whether the client area of the window appears sunken within the frame. TRUE – Client area appears sunken FALSE – Client area does not appear sunken

Window property	Datatype	Description
Close	Boolean	Adds an OK icon to the title bar of a window that you deploy to a Windows CE platform. When users click OK, the application is closed. Values are: TRUE – Window has an OK icon FALSE – Window has no OK icon Setting Close to TRUE sets SmartMinimize to FALSE.
ColumnsPerPage	Integer	Specifies the number of columns on a page. The default is 0 (10 columns per page). For information, see “Scrolling in windows and user objects” on page 569. PocketBuilder multiplies UnitsPerColumn by ColumnsPerPage to determine the number of PowerBuilder units to scroll the window horizontally when the user clicks in the scroll bar.
ContextHelp	Boolean	Not supported in PocketBuilder. When WindowType = Response!, this property specifies whether the small question mark button appears in the title bar. The question mark button can fire the Help event on the control that is clicked next. TRUE – Question mark button is displayed in title bar next to minimize button in Response windows FALSE – Question mark button is not displayed
Control[]	WindowObject	Contains the controls in the window. You should not change the contents of this array in a script.
ControlMenu	Boolean	Specifies whether the Control Menu box displays in the title bar. Values are: TRUE – Displays in title bar FALSE – Does not display in title bar
DefaultSize	Boolean	Changes the size of the current window to the default size specified on the Size page of the Design Options dialog box in the Window painter. Values are: TRUE – Window is default size FALSE – Window is not default size
Enabled	Boolean	Specifies whether the window is enabled (can send and receive messages). Values are: TRUE – Can send/receive messages FALSE – Cannot send/receive messages
Height	Integer	Specifies the height of the window, in PowerBuilder units. You cannot resize minimized or maximized windows at runtime.

Window property	Datatype	Description
HScrollBar	Boolean	Specifies whether a horizontal scroll bar displays in the window. Values are: TRUE – A scroll bar displays FALSE – A scroll bar does not display
Icon	String	Specifies a stock icon or an ICO file that will be displayed when the window is minimized. The default value is AppIcon!, which is the icon selected for the Application object. If no icon is selected for the Application object, the Windows logo is used.
LinesPerPage	Integer	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” on page 569. PocketBuilder multiplies UnitsPerLine by LinesPerPage to determine the number of PowerBuilder units to scroll the window vertically when the user clicks in the scroll bar.
MaxBox	Boolean	Specifies whether a Maximize Box displays in the title bar. Values are: TRUE – Maximize Box displays FALSE – Maximize Box does not display
MenuBar	Boolean	Allows room for a menu bar at the bottom of the window. Values are: TRUE – Window has room for a menu bar FALSE – Window does not have room for a menu bar
MenuID	Menu	Specifies the ID of a menu. PocketBuilder uses MenuID internally. To change the menu for a window from a script, use the ChangeMenu function; and to display a pop-up 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. PocketBuilder uses MenuName internally. To change the menu for a window from a script, use the ChangeMenu function; to display a pop-up 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: TRUE – Minimize Box displays FALSE – Minimize Box does not display

Window property	Datatype	Description
PaletteWindow	Boolean	Not supported in PocketBuilder. When WindowType = Popup!, this property specifies that the window will have an appearance appropriate for small palette windows that display over the application. TRUE – Pop-up window displays as the topmost window with a smaller Close button in the title bar and no Minimize or Maximize buttons. FALSE – No change in appearance
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: TRUE – Window is resizable FALSE – Window is not resizable
RightToLeft	Boolean	Not supported in PocketBuilder. Specifies that characters should be displayed in right-to-left order. The application must be running on 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
ShowSIPButton	Boolean	Specifies whether the soft input panel (SIP) button is displayed in a window. Values are: TRUE – Window displays the SIP button FALSE – Window does not display the SIP button
SmartMinimize	Boolean	Adds an X icon to the title bar of a main window that you deploy to a Windows CE platform. When users click the X, the application is removed from the current navigational stack, but remains in memory. Values are: TRUE – Window has an X icon FALSE – Window has no X icon Setting SmartMinimize to TRUE sets Close to FALSE.
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: TRUE – Title bar displays FALSE – No title bar displays The user can move a window only if it has a title bar.

Window property	Datatype	Description
ToolbarAlignment	Toolbar Alignment (enumerated)	In an MDI frame window, specifies where the toolbar displays. Values are: 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: 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.
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	Specifies the number of PowerBuilder units to be scrolled right or left when a user clicks the left or right arrow in the horizontal scroll bar in a window or user object. The default is 0 (1/100 of the width of the window). To make the end of the scroll bar match the content, UnitsPerLine must be set according to the content width. For information, see “Scrolling in windows and user objects” on page 569. PocketBuilder multiplies UnitsPerColumn by ColumnsPerPage to determine the number of PowerBuilder units to scroll the window horizontally when the user clicks in the scroll bar.

Window property	Datatype	Description
UnitsPerLine	Integer	Specifies the number of PowerBuilder units to be scrolled up or down when a user clicks the up or down arrow in the vertical scroll bar in a window or user object. The default is 0 (1/100 of the window height). To make the end of the scroll bar match the content, UnitsPerLine must be set according to the content length. For information, see “Scrolling in windows and user objects” on page 569. PocketBuilder multiplies UnitsPerLine by LinesPerPage to determine the number of PowerBuilder units to scroll the window vertically when the user clicks in the scroll bar.
Visible	Boolean	Specifies whether the window is visible. Values are: TRUE – Window is visible FALSE – Window is not visible
VScrollBar	Boolean	Specifies whether a vertical scroll bar displays. Values are: TRUE – Scroll bar displays FALSE – Scroll bar does not display
Width	Integer	Specifies the width of the window, in PowerBuilder units. You cannot resize minimized or maximized windows at runtime.
WindowState	WindowState (enumerated)	Specifies the state in which you want to run a window. Values are: Maximized! Minimized! Normal! Do not change the WindowState property in the Open event of a window opened as a sheet.
WindowType	WindowType (enumerated)	Specifies the type of window. Values are: Child! Main! MDI! MDIHelp! Popup! Response!

Window property	Datatype	Description
X	Integer	Specifies the X position (distance from left edge of screen) of the window, in PowerBuilder units. 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. The workspace is the area between the sides of the window (not including the thickness of the frame, toolbar, or scroll bar, if any) and the top and bottom of the window (not including the thickness of the border or the title bar, menu bar, toolbar, or scroll bar, if any). You cannot move a maximized window at runtime.
Y	Integer	Specifies the Y position (distance from the top of the screen) of the window, in PowerBuilder units. 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. The workspace is the area between the sides of the window (not including the thickness of the frame, toolbar, or scroll bar, if any) and the top and bottom of the window (not including the thickness of the border or the title bar, menu bar, toolbar, or scroll bar, if any). You cannot move a maximized window at runtime.

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.

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

Window event	Occurs
CloseQuery	When you remove a window from display (close it). When you close a window, PocketBuilder triggers the CloseQuery event and then inspects the value of Message.ReturnValue. If the Message.ReturnValue is 1, the window cannot be closed. Closing any window causes PocketBuilder to close all child and pop-up windows that it opened. In PowerBuilder, closing an MDI Frame window closes all sheet windows within it. Any window thus being closed can set Message.ReturnValue to cancel the close operation.
Deactivate	When the window becomes inactive.
DoubleClicked	When the user double-clicks in an unoccupied area of the window (any area with no visible, enabled object).
DragDrop	When a dragged control is dropped on the window.
DragEnter	When a dragged control enters the window.
DragLeave	When a dragged control leaves the window.
DragWithin	When a dragged control is within the window.
Help	When the user presses the F1 key or drags the context help button (question mark) from the title bar to a menu item or control.
Hide	Just before the window is hidden.
HotLinkAlarm	After a Dynamic Data Exchange (DDE) server application has sent new (changed) data and the client DDE application has received it.
Key	When the user presses a key and the insertion point is not in a 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 PocketBuilder event.
RButtonDown	When the right mouse button is pressed in an unoccupied area of the window (any area with no visible, enabled object).
RemoteExec	When a DDE client application has sent a command.
RemoteHotLinkStart	When a DDE client application wants to start a hot link.
RemoteHotLinkStop	When a DDE client application wants to end a hot link.
RemoteRequest	When a DDE client application requests data.
RemoteSend	When a DDE client application has sent data.
Resize	When the user or a script opens or resizes a window.

Window event	Occurs
Show	When a script executes the Show function for this window. The event occurs just before the window is displayed.
SystemKey	When the user presses Alt or Alt plus another key, except when the insertion point is in a DataWindow control or RichTextEdit control.
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	Datatype 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.
CloseChannel	Integer	Closes a DDE channel.
CloseUserObject	Integer	Removes the specified user object from view, closes it, and executes its Destructor event.
ExecRemote	Integer	Asks a DDE server application to execute the specified command.
GetActiveSheet	Window	Returns the currently active sheet in the specified MDI frame window.
GetCommandDDE	Integer	Obtains the command sent by the client application when you application is a DDE server.

Window function	Datatype returned	Description
GetCommandDDEOrigin	Integer	When called by the DDE server application, obtains the application name parameter used by the DDE client sending the command.
GetContextService	Integer	Creates a reference to a context-specific instance of the specified service.
GetDataDDE	Integer	Obtains data sent from another DDE application and stores it in the specified string variable.
GetDataDDEOrigin	Integer	Determines the origin of data from a hot-linked DDE server application or a DDE client application, and if successful, stores the application's DDE identifiers in the specified strings.
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.
GetRemote	Integer	Asks a DDE server application to provide data and stores that data in the specified variable.
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 or minimized window.
OpenChannel	Long	Opens a channel to a DDE server application.
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	Datatype 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.
RespondRemote	Integer	Not supported in PocketBuilder. Sends a DDE message indicating whether the command or data received from a remote DDE application was acceptable.
SetDataDDE	Integer	Not supported in PocketBuilder. Sends data to a DDE client application when PowerBuilder is acting as a DDE server.
SetFocus	Integer	Sets focus to the specified window.
SetMicroHelp	Integer	Not supported in PocketBuilder. 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.
SetRemote	Integer	Asks a DDE server application to accept data and store it in the specified location.
SetToolbar	Integer	Sets the values of the Visible, Alignment, and Title properties of the toolbar.
SetToolbarPos	Integer	Sets the position of a fixed toolbar.
Show	Integer	Makes the window visible.
StartHotLink	Integer	Not supported in PocketBuilder. Establishes a hot link with a DDE server application so that PowerBuilder will be notified immediately of any changes in the specified data.
StartServerDDE	Integer	Establishes your application as a DDE server.
StopHotLink	Integer	Terminates a hot link with a DDE server application.
StopServerDDE	Integer	Causes your application to stop acting as a DDE server application.
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.

Window function	Datatype returned	Description
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 scroll bar, if any) and the top and bottom of the window (not including the thickness of the border or the title bar, menu bar, toolbar, or scroll bar, if any).
WorkSpaceWidth	Integer	Returns the width of the workspace of the specified window.
WorkSpaceX	Integer	Returns the distance from the left edge of the screen to the left edge of the workspace of the specified window.
WorkSpaceY	Integer	Returns the distance from the top of the screen to the top of the workspace of the specified window.

Property Descriptions and Usage

About this chapter

This chapter lists the properties for PocketBuilder controls. For properties specific to controls in DataWindow objects, see the *DataWindow Reference*.

Accelerator

Applies to

Controls that accept user input, including list boxes, MultiLineEdit, SingleLineEdit, ListView, and TreeView

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

Accelerator keys allow users to select an item (that is, change focus) with a keystroke rather than with the mouse. An underlined character in the item's name or label tells the user what key to press. The user presses it in combination with the Alt key. If the currently selected control is not an editable control (such as a SingleLineEdit, MultiLineEdit, ListBox, or DropDownListBox), you need only press the accelerator key.

Accelerator keys are different from shortcut keys, which are defined key combinations that provide a quick way to accomplish certain tasks.

PocketBuilder term	Windows term	Motif term
accelerator key	mnemonic access character	mnemonic
shortcut key	shortcut key or accelerator key	accelerator key

Usage

In a painter

❖ To select a character as an accelerator key

- Type the character into the Accelerator box on the General page of the control's Properties view.

For example, to set m as the accelerator, type m in the box

Accelerators for unlabeled controls To show the user what accelerator key to use for an unlabeled control or box, define StaticText to act as a label. Include an ampersand (&) before the character you want underlined. For example, in the StaticText control's General page, set the Text property to a value like Edit &Maintenance Data for a drop-down list that has m as an accelerator key. If you want an ampersand to display in the text, type two ampersands, and if you want an ampersand to display and serve as the accelerator key, type three ampersands.

In scripts

The Accelerator property is an integer consisting of the ASCII value of the accelerator key. Both of the following lines set m as the accelerator character for a MultiLineEdit control:

```
mle_1.Accelerator=67
mle_1.Accelerator = ASC("M")
```

Activation

Applies to

OLE controls

PocketBuilder	✕
PowerBuilder	✓

Description

Specifies how the user activates the control.

Address

Applies to

SMSAddress objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✕

Description

The Address property identifies the destination address of an SMS message.

Usage

In scripts

The Address property takes a string value. For example:

```
SMSAddress mysmsaddress
mysmsaddress.Address = "+1.800.123.4567"
```

AddressType

Applies to

SMSAddress objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The AddressType property specifies the type of the destination address of an SMS message. Values are:

- **SMSAT_Abbreviated!** Abbreviated number
- **SMSAT_Alphanumeric!** Coded according to Global System for Mobile Communications (GSM) standard
- **SMSAT_International!** International number
- **SMSAT_National!** National number (no prefix or escape characters)
- **SMSAT_NetworkSpecific!** Number specific to the serving network, for example, used to access an operator
- **SMSAT_Subscriber!** A short number representation is stored in one or more service centers
- **SMSAT_Unknown!** The user or network has no information about the numbering plan

Usage

In scripts

The AddressType property takes a value of the SMSAddrType enumerated variable. This example sets the address type for an international message:

```
SMSAddress mysmsaddress
mysmsaddress.Address = "441304555444"
mysmsaddress.AddressType = smsat_international!
```

Alignment

Applies to

Controls that display text

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

For most controls that display text, the alignment property specifies the alignment of all the text in the control. Text can be centered, left aligned, or right aligned.

Usage

In a painter

- ❖ **To specify text alignment for controls other than RichTextEdit:**
 - On the General page of the control's Properties view, select an alignment from the Alignment drop-down list, or click the Left, Center, and Right alignment buttons in the StyleBar. Use the StyleBar to set the alignment for several selected objects at once.

In scripts

The datatype of the Alignment property is the Alignment enumerated datatype. It has four values: Center!, Left!, and Right! apply to all controls with text, Justify! applies only to RichTextEdit controls.

For example, the following line specifies center alignment for a MultiLineEdit control:

```
mle_1.Alignment = Center!
```

AllowEdit

Applies to

DropDownListBox controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

When AllowEdit is enabled, the user can edit the selection in the text box of a drop-down list.

If AllowEdit is not enabled, the user can only make a selection from the list and cannot edit the selection.

Usage

In a painter

❖ To allow editing:

- On the General page of the control's Properties view, select the AllowEdit check box.

In scripts

The AllowEdit property takes a boolean value.

This example sets AllowEdit for a DropDownListBox.

```
ddlb_1.AllowEdit = TRUE
```

Altitude

Applies to

GPSTFix objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Indicates the height above mean sea level in meters of the current position fix.

Usage

In scripts

The Altitude property is a real number indicating the height of a GPS fix above mean sea level. The following lines create a SerialGPS object, retrieve information about the current fix, and write the altitude to a single line edit box:

```
SerialGps myGPS
GPSTFix myFix
Integer rc

MyGPS = CREATE SerialGPS
rc = myGPS.Open()
...
rc = MyGPS.GetFix(myFix)
sle_1.txt = "Altitude: " + String(myFix.Altitude) + "M"
```

AutoArrange

Applies to

ListView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When AutoArrange is enabled, PocketBuilder arranges icons automatically in large and small icon views.

Usage

In a painter

❖ To enable automatic icon arrangement:

- Select the AutoArrange check box on the General page of the ListView control's Properties view.

In scripts

The AutoArrange property takes a boolean value. The following line specifies automatic arrangement of icons in a ListView.

```
lv_1.AutoArrange = TRUE
```

AutoHScroll

Applies to

DropDownListBox, DropDownPictureListBox, EditMask, MultiLineEdit, SingleLineEdit controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When automatic horizontal scrolling is enabled, text in the control will scroll left or right automatically when the user enters or deletes data. When automatic horizontal scrolling is not enabled, text does not scroll left or right as the user changes data, and data that exceeds the width of the line provided is ignored.

Usage

In a painter

❖ To enable automatic horizontal scrolling:

- Select the AutoHScroll check box on the General page of the object's Properties view.

In scripts

The AutoHScroll property takes a boolean value.

For example, this statement enables automatic right and left scrolling as the user enters or modifies data in the edit box of a DropDownListBox control.

```
ddl1.AutoHScroll = TRUE
```

Automatic

Applies to

CheckBox and RadioButton controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When the Automatic property is enabled, the state of the control changes automatically when the user selects it. Typically, the state toggles between *selected* and *not selected*. For check boxes, if the ThreeState property has been enabled, the state of the control also toggles to a *third state*.

When this property is enabled, a mark is displayed in the control when the control's state is *selected* and no mark is displayed when the control's state is *not selected*. For check boxes, if the ThreeState property is enabled, a grayed out mark is displayed for the *third state*.

Usage

In a painter❖ **To enable automatic state change:**

- Select the Automatic check box on the General page of the control's Properties view.

In scripts

The Automatic property takes a boolean value. This example sets a CheckBox so that its state changes each time it is selected.

```
cbx_1.Automatic = TRUE
```

AutoScale

Applies to grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description AutoScale is a property of the Category and Value grAxis objects that are part of graph controls. There are three grAxis objects: Category, Series, and Values. Autoscale specifies whether or not to scale the axis of the Graph automatically to the minimum and maximum values for the data.

Usage In a painter

- ❖ To enable autoscaling:
 - 1 Display the Axis tab page on the graph’s Properties view.
 - 2 Select the desired Axis from the Axis drop-down list.
 - 3 Select the AutoScale check box, if it is enabled.AutoScale is enabled only if it is applicable to the selected graph type and axis.

In scripts

The AutoScale property takes a boolean value. The following line turns off autoscaling for the Values axis in the Graph gr_emp.

```
gr_emp.Values.AutoScale = FALSE
```

AutoSize

Applies to grDispAttr objects within Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description The AutoSize property allows PocketBuilder to change the font size of the text object automatically according to the amount of text being displayed. If automatic sizing is not enabled, you must set the text size.

Usage**In a painter**❖ **To enable automatic sizing of text objects:**

- 1 Display the Text tab page of the control's Properties view.
- 2 Select a text object from the Text Object list.
- 3 Select the AutoSize check box.

If you uncheck the AutoSize check box for a specific text object, set the text size for that object by selecting a value from the TextSize list.

In scripts

The AutoSize property takes a boolean value and can be set via the grDispAttr object for each text component.

This example turns autosizing off for the graph control's title and then sets a specific text size.

```
gr_1.TitleDispAttr.AutoSize = FALSE
gr_1.TitleDispAttr.TextSize = 10
```

This example turns autosizing off for the label of the Category Axis of the graph control and then sets a specific text size.

```
gr_1.Category.LabelDispAttr.AutoSize = FALSE
gr_1.Category.LabelDispAttr.TextSize = 8
```

AutoSkip

Applies to**EditMask controls**

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When the AutoSkip property is enabled, the user's cursor will automatically skip to the next control in the tabbing order after entering all the characters allowed by the mask. If AutoSkip is not enabled, the cursor will not skip automatically to the next control.

Usage

In a painter

- ❖ **To enable Auto Skip:**
 - Select the AutoSkip check box on the Mask tab page of the EditMake control’s Properties view.

In scripts

The AutoSkip property takes a boolean value. This example enables automatic skipping to the next control.

```
em_1.AutoSkip = TRUE
```

AutoVScroll

Applies to

MultiLineEdit controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When automatic vertical scrolling is enabled, text in the control will scroll up or down automatically when the user enters or modifies data. When automatic vertical scrolling is not enabled, text will not scroll up or down automatically as the user changes data, and data that exceeds the height of the space provided is ignored.

Usage

In a painter

- ❖ **To enable automatic vertical scrolling:**
 - Select the AutoVScroll check box on the General page of the object’s Properties view.

In scripts

The AutoVScroll property takes a boolean value. The following example enables automatic vertical scrolling.

```
mle_1.AutoVscroll = TRUE
```

Azimuth

Applies to

GPSSatellitePosition objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The Azimuth property indicates the position of the satellite relative to true north in degrees. The range of values is 0 to 359.

Usage

In scripts

The Azimuth property takes an integer value. This example writes the azimuth, elevation, signal strength, and PRN properties to a multiline edit box:

```
integer li_aAzimuth, li_elevation, li_SNR, li_PRN
GPSSatellitePosition myGPS_SP

...
li_azimuth = myGPS_SP.Azimuth
li_elevation = myGPS_SP.Elevation
li_SNR = myGPS_SP.SNR
li_PRN = myGPS_SP.PRN
mle_1 = "Azimuth: " + String(li_azimuth) + "~r~n"
mle_1 += "Elevation: " + String(li_elevation) + "~r~n"
mle_1 += "Signal strength: " + String(li_SNR) + "~r~n"
mle_1 += "PRN: " + String(li_PRN)
```

BackColor

Applies to

Windows and most controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The BackColor property defines the color to be used for the background of an object. When you are defining the background color in a painter, some of the choices take their values from the current Windows color scheme or from custom colors. To add your own colors to the color drop-down list, select Design>Custom Colors before displaying the Properties view.

Usage

In a painter

❖ **To set the background color for a window or graph:**

- Select a color from the BackColor drop-down list on the General page in the window's or graph's Properties view.

❖ **To set the background color for controls:**

- Select a color from the BackColor drop-down list on the Font tab page in the control's Properties view.

❖ **To set the background color for text objects in graphs:**

- 1 Select the desired text object in the Text Object list box on the Text tab page of the graph control's Properties view.
- 2 Select a color from the BackColor drop-down list.

In scripts

The BackColor property takes a long (-2 to 16,777,215) that specifies the numerical value of the background color of windows and other objects. The BackColor value is a combination of values for the red, green, and blue components of the color.

If you do not know the long value for the color, choose Design>Custom Colors to determine the red, green, and blue values and then call the RGB function to specify the color in a script.

The following example sets yellow as the background color for a graph control:

```
gr_1.BackColor = RGB(255, 255, 0)
```

For text displayed in the graph, BackColor is a property of a text component, such as labels on an axis.

The following example sets background color to blue for text labels on the Category axis of the graph control:

```
gr_1.Category.LabelDispAttr.BackColor = RGB(0, 128, 255)
```

BeginX

Applies to

Line controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The BeginX property specifies the X position in PowerBuilder units of the beginning of the line.

The X coordinate is the distance from the left edge of the window or custom user object. If the object is a main window or custom user object, the distance is relative to the screen. If it is not a main window, the distance is relative to the parent window. In PowerBuilder, if a window is opened in an MDI frame window, the distance is relative to the MDI frame.

Usage

In a painter

❖ To specify the beginning X coordinate of the line:

- Insert the line. If you want to change the beginning location, change the value of the BeginX field on the Position tab page of the line's Properties view.

In scripts

The BeginX property takes an integer value. The following example sets the beginning of the line at 1000 Power Builder units from the left edge of the window or user object and 500 PowerBuilder units from the top edge:

```
ln_1.BeginX = 1000
ln_1.BeginY = 500
```

BeginY

Applies to

Line controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description	The BeginY property specifies the Y position in PowerBuilder units of the beginning of the line. The Y coordinate is the distance from the top edge of the window or custom user object. If the object is a main window or custom user object, the distance is relative to the screen. If it is not a main window, the distance is relative to the parent window. In PowerBuilder, if a window is opened in an MDI frame window, the distance is relative to the MDI frame.
Usage	In a painter ❖ To specify the beginning Y coordinate of the line: • Insert the line. If you want to change the beginning location, change the value of the BeginY field on the Position tab page of the line’s Properties view. In scripts The BeginY property takes an integer value. The following example sets the beginning of the line at 500 Power Builder units from the top edge of the window or user object and 1000 PowerBuilder units from the left edge.: <pre>ln_1.BeginY = 500 ln_1.BeginX = 1000</pre>

Body

Applies to	NotificationBubble objects <table><tr><td>PocketBuilder on Pocket PC</td><td>✓</td></tr><tr><td>PocketBuilder on Smartphone</td><td>✗</td></tr><tr><td>PowerBuilder</td><td>✗</td></tr></table>	PocketBuilder on Pocket PC	✓	PocketBuilder on Smartphone	✗	PowerBuilder	✗
PocketBuilder on Pocket PC	✓						
PocketBuilder on Smartphone	✗						
PowerBuilder	✗						
Description	Defines the main text for the notification bubble. Text must be in HTML format.						
Usage	In a painter ❖ To add text for the body of a notification bubble: • Enter the text you want in the Body text box on the General page of the NotificationBubble object’s Properties view.						

In scripts

The Body property takes a string in HTML format. The following example sets the text for a notification bubble:

```
STRING ls_body
NotificationBubble myBubble
Integer li_return

ls_body = "<html><body><form method=~\"POST~\" action=~\"&
+\"<p>This is <font color=~\"#0000FF~\"><b>HTML</b></p>\" &
+\"</font>notification coming from \"

ls_body += "<font color=~\"#FF0000~\"><i>Pocket</i>\"&
+\"PowerBuilder</font></p><p align=right><input \"&
+\"type=button name='cmd:10' value='My Ok'>&nbsp;&nbsp;&\"&
+\"<input type=button name='cmd:2' value='Cancel'>\"&
+\"</p></body></html>\"

myBubble = CREATE NotificationBubble
myBubble.caption = "My Title"
li_return = myBubble.icon ("bubble.ico")
myBubble.body = ls_body
li_return = myBubble.update()
```

BoldSelectedText

Applies to

Tab controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When the BoldSelectedText property is enabled, the label for a tab page becomes bold when the user selects the tab page. If this property is not enabled, the tab text of the selected tab page has the same appearance as the tab text of the other tab pages.

Usage

In a painter❖ **To enable the BoldSelectedText property:**

- Select the BoldSelectedText check box on the General page of the Tab control's Properties view.

In scripts

The BoldSelectedText property takes a boolean value. The following example specifies that labels on tab pages of the tab_1 control are bold when they are selected.

```
tab_1.BoldSelectedText = TRUE
```

Border

Applies to

Windows, other controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When the Border property is enabled, the window or control has a border.

Some controls and window types always have borders, whether or not the Border property is enabled.

Usage

In a painter

- ❖ **To specify a border:**
 - Select the Border check box on the General page of the window's or control's Properties view.

In scripts

The Border property takes a boolean value. The following example specifies that a static text control should display a border and sets the border style to 3D Lowered:

```
st_1.Border = TRUE  
st_1.BorderStyle = StyleLowered!
```

BorderColor

Applies to

StaticText and StaticHyperLink ccontrols

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

BorderColor defines the color to be used for the border around a StaticText or StaticHyperLink control. The color is only visible with the Box border style.

To add your own colors to the BorderColor list, select Design>Custom Color before displaying the Properties view.

Usage

In a painter

❖ To specify a border color:

- 1 Select the Border check box on the General page of the StaticText control's Properties view.
- 2 Select StyleBox! from the BorderStyle list.
- 3 Select the desired color from the BorderColor list on the General page.

In scripts

The BorderColor property takes a long value. If you do not know the long value for the color, choose Design>Custom Colors to determine the red, green, and blue values and then call the RGB function to specify the color in a script.

This example enables the display of a border, specifies the Box border style, and then specifies red as the border color for StaticText control st_1.

```
st_1.Border = TRUE
st_1.BorderStyle = StyleBox!
st_1.BorderColor = RGB(255,0,0)
```

BorderStyle

Applies to

Most controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

The BorderStyle property lets you define the border appearance of a control. Styles include 3D Lowered, 3D Raised, Box, and Shadow Box.

Usage

In a painter

❖ **To set the border style:**

- Select the desired style from the BorderStyle list on the General page of the control’s Properties view.

To turn the border off, uncheck the Border check box on the General page.

In scripts

To change the appearance of the border, set Border to TRUE and set the BorderStyle property to a value of the BorderStyle enumerated datatype.

The following example sets the border for a DropDownListBox.

```
ddlb_1.Border = TRUE
ddlb_1.BorderStyle = StyleLowered!
```

BottomMargin

Applies to

RichTextEdit controls

PocketBuilder	✗
PowerBuilder	✓

Description

The BottomMargin property specifies the size in inches of the bottom margin on the printed page.

BringToTop

Applies to

Windows and controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

For windows, this property specifies whether PocketBuilder will move the window to the top of the front-to-back order. For windows, this property can be set only in scripts.

For controls, this property specifies whether PocketBuilder moves the control to the top of the front-to-back order within the window.

Usage

In a painter

❖ To set BringToTop property for controls:

- 1 Select the control.
- 2 Right click on the control and select Bring to Front or Send to Back from the Pop-up menu, or select Format>Bring to Front or Format>Send to Back from the menu bar.

In scripts

BringToTop takes a boolean value.

This statement brings the window in front of other windows in the application.

```
w_1.BringToTop = TRUE
```

ButtonHeader

Applies to

ListView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When the ButtonHeader property is enabled, the column titles in a ListView's report view appear as pushable buttons instead of labels.

Usage

In a painter

- ❖ **To set the ButtonHeader property:**
 - Select the ButtonHeader check box on the General page of the ListView control's Properties view.

In scripts

The ButtonHeader property is only relevant to a report view in a ListView control. To enable report view, you must write a script that establishes columns with the AddColumn and SetColumn functions, and then populate the columns using the SetItem function.

See "Using Lists in a Window" in the *Resource Guide* for more information about using report view.

The ButtonHeader property takes a boolean value. The following example specifies a button header for a report in a ListView.

```
lv_1.ButtonHeader = TRUE
```

CallState

Applies to

CallLogEntry objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Indicates the state of the call log entry. Missed calls can indicate an occupied phone line or no answer. Values are:

- **1** Outgoing call that was answered
- **2** Outgoing call that was missed
- **11** Incoming call that was answered
- **12** Incoming call that was missed

Usage

In scripts

The CallState property is an integer indicating the state of an entry in the call log. The following lines get the state of an entry in the call log clog_1:

```
CallLogEntry l_entry
```

```
integer li_index
string ls_state

l_entry = clog_1.GetEntry(li_index)
ls_state = l_entry.CallState
```

CameraName

Applies to

Camera objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Read-only name for the camera that is specified in the CameraType property.

Usage

In scripts

The CameraName property is a string that corresponds to the type of camera specified before you call the Open function. For example, given these statements, the sle_name text box will contain the string “HP Photosmart”:

```
cam_1.CameraType=71 //specifier for HP Photosmart
cam_1.Port="SIO1:"
cam_1.Open(w_myphoto_main)
sle_name.text = cam_1.CameraName
```

CameraType

Applies to

Camera objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Specifier for the type of camera you want to use. Values are:

- 11 for a VEO 130S camera

Usage

- 71 for an HP Photosmart camera
- 81 for an HTC camera using the IA Camera Wizard

In a painter

❖ **To specify the CameraType**

- 1 Select the Camera object in the Non-Visual Object List and open the Properties view.
- 2 Select the CameraType you want from the drop-down list.
- 3 Enter the Port or Folder property as required for the camera type you selected.

In scripts

The CameraType property takes an integer value. The following example uses the specifier for the VEO 130S camera:

```
cam_1.CameraType=11
cam_1.Port="SIO1:"
cam_1.Open(w_myphoto_main)
```

Cancel

Applies to

CommandButton, PictureButton, OLECustomControl controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

Cancel defines whether the Esc key activates the button or control. If the Cancel property is enabled, the Esc key triggers the control's Clicked event. If Cancel is not enabled, the control does not respond to the Esc key.

If you enable Cancel for more than one control, the last one set responds to the Esc key.

Usage

In a painter

❖ **To enable the Cancel property:**

- Select the Cancel check box on the General page of the control's Properties view.

In scripts

The Cancel property takes a boolean value. The following line allows the CommandButton to respond to the Esc key.

```
cb_1.Cancel = TRUE
```

Caption

Applies to

NotificationBubble objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Description

Defines text for the notification bubble caption (title).

Usage

In a painter❖ **To add text for a notification bubble caption:**

- Enter the text you want in the Caption text box on the General page of the NotificationBubble object's Properties view.

In scripts

The Caption property takes a string. The following example sets the caption text for a notification bubble:

```
nb_myBubble.Caption = "Message in a bubble"
```

Category

Applies to

Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description The Category property of the Graph control allows you to specify the properties of the category axis of the graph. The category axis is an object (of the type grAxis) within the Graph control.

- Usage**
- In a painter**
- ❖ **To set the properties of the Category Axis of a graph control:**
 - 1 Display the Axis tab page of the graph control's Properties view.
 - 2 Select Category from the Axis drop-down list.
 - 3 Set the desired values on the Axis tab page.

In scripts

The datatype of the Category property is grAxis, which is a type of object that has its own properties for controlling the axis' appearance. Use the following syntax to specify values for the category axis:

GraphControlName.Category.grAxisProperty = value

The following example sets the label of the category axis of a graph control:

```
gr_1.Category.Label = "Types of Products"
```

CategorySort

Applies to	Graph controls	
	PocketBuilder on Pocket PC	✓
	PocketBuilder on Smartphone	✓
	PowerBuilder	✓

Description The CategorySort property specifies how the categories are sorted: ascending, descending, or unsorted.

- Usage**
- In a painter**
- ❖ **To specify how the categories are sorted:**
 - Select the desired sort type from the CategorySort drop-down list on the General page of the graph control's Properties view.

In scripts

The CategorySort property takes a value of the grSortType enumerated datatype, which has the values Ascending!, Descending!, Unsorted!, and UserDefinedSort!.

The following example specifies that the categories should be unsorted:

```
gr_1.CategorySort = Unsorted!
```

Center

Applies to

Windows

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

Causes the window to be centered on the screen when it is created or sized.

Usage

In a painter

❖ **To set the Center property for a window:**

- Select the Center check box on the General page of the window's Properties view.

In scripts

The Center property takes a boolean value. The following example sets the Center property for the window, w_mine.

```
w_mine.Center = TRUE
```

Checked

Applies to

CheckBox and RadioButton controls and Menu objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description	When the value of the Checked property is TRUE, the item or control is selected.								
	<table><tr><th>Control</th><th>Displayed when selected</th></tr><tr><td>CheckBox</td><td>X displays in the check box</td></tr><tr><td>Menu item</td><td>Check mark displays next to the item</td></tr><tr><td>RadioButton</td><td>Center of button becomes dark</td></tr></table>	Control	Displayed when selected	CheckBox	X displays in the check box	Menu item	Check mark displays next to the item	RadioButton	Center of button becomes dark
Control	Displayed when selected								
CheckBox	X displays in the check box								
Menu item	Check mark displays next to the item								
RadioButton	Center of button becomes dark								

If Checked is FALSE, the item or control is not selected.

Usage

In a painter

❖ To set the Checked property on a control:

- Select the Checked check box on the General page of the control’s Properties view

❖ To set the Checked property on a menu item:

1 Select the menu item in the Menu painter.

2 Select the Checked check box on the General page.

In scripts

The Checked property takes a boolean value. The following example sets the Checked property of a RadioButton to TRUE, which causes the button to be selected.

```
rb_1.Checked = TRUE
```

For menu items in drop-down or cascading menus, you can also use the Check and Uncheck functions. See the descriptions of those functions for examples of equivalent syntax.

Close

Applies to

Windows	
PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description The Close property adds an OK icon to the title bar of a window that you deploy to a Windows CE platform. When users click OK, user input is confirmed, the window object is destroyed, and the PocketBuilder application is closed

Usage

In a painter

❖ **To display an OK icon to a window:**

- Select the Close check box on the General page of the window's Properties view.

Selecting the Close check box clears the Smart Minimize check box.

In scripts

This property cannot be set in a script.

ColumnsPerPage

Applies to

Windows and user objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

The ColumnsPerPage property determines the number of columns on a page for scrolling purposes. The default is 0 (10 columns per page). PocketBuilder multiplies UnitsPerColumn by ColumnsPerPage to determine the number of PowerBuilder units to scroll the window horizontally when the user clicks in the scroll bar.

For information on calculating ColumnsPerPage and UnitsPerColumn, see "Scrolling in windows and user objects" on page 569.

Usage note

To control the vertical scroll bar in a window or user object, use the UnitsPerLine and LinesPerPage properties.

Usage

In a painter

❖ **To set the ColumnsPerPage property:**

- Enter the desired number (between 1 and 100) in the ColumnsPerPage option on the Scroll tab page of the window's Properties view.

In scripts

The ColumnsPerPage property takes an integer value between 1 and 100. The following line sets ColumnsPerPage for a window to 20.

```
This.ColumnsPerPage = 20
```

ConfigParams

Applies to

Camera and SerialGPS objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

For the Camera object, the ConfigParams property is intended to be used to customize the behavior of the object. It is currently undefined.

For the SerialGPS object, the ConfigParams property consists of four comma-separated keyword value pairs. The following table describes these keyword value pairs:

Keyword value pair	Description
BufferSize=nnnn	Number of bytes in the read buffer. A buffer size of 1500 to 2500 bytes is recommended.
Refresh=nnnn	Number of milliseconds before forcing a new read request. A 2000 byte buffer might contain 25 or more GPS sentences. If the user issues GetFix or GetHeading requests inside a loop, the parser works with the current data buffer until the refresh interval has expired, or until the end of the buffer is reached . When either of these conditions are met, a new data buffer is read from the serial port.
Timeout=nnnn	PocketBuilder terminates a read request when the timeout limit is exceeded, preventing your program from hanging indefinitely. Currently, this technique has only been tested on Windows XP and CE 2003 operating systems. As an additional precaution, the user can also set Multithread=1 to run the read request on a secondary thread. The timeout interval is specified in milliseconds and a -14 Read Timeout error is generated if it is exceeded.

Keyword value pair	Description
Multithread= <i>n</i> (where <i>n</i> = 0 or 1)	The default behavior (Multithread=0) is to use SetCommTimeouts to set a timeout interval for the serial port. A less efficient, but reliable alternate technique is to run each read request on a secondary thread and terminate the thread when the timeout interval is exceeded (Multithread=1). This should be considered only if the user's operating system does not reliably support SetCommTimeouts.

For the GPS base class, the ConfigParams property is used to specify a string literal containing NMEA raw data sentences. You can store this data in a database, in the Registry, or in an ANSI text file.

Usage

In a painter

Type the value in the ConfigParams box on the General page of the object's Properties view.

In scripts

The ConfigParams property takes a string value.

Examples

The following is an example for the serialGPS object:

```
Integer li_rc
GPSFix myFix
gps_1.SerialPort = "COM5:" //override default com8 port
gps_1.ConfigParams = "bufferSize=2000,refresh=2000," &
  + "timeout=5000,multithread=0"
li_rc = gps_1.Open()
IF li_rc = 1 THEN
  li_rc = gps_1.GetFix(myFix)
  //process data
ELSE //process error msg
END IF
```

The following is an example for the GPS base class object:

```
string lsBillaricay
GPS myGps
GPSFix myFix
Integer rc

lsBillaricay="$GPGGA,162443.559,5138.3437,\"+ &
"N,00025.3591,E,1,05,2.3,127.5,M,,,0000*00~r~n"
myGps = create GPS
myGps.ConfigParams = lsBillaricay
rc = myGps.Open()
```

```
rc = myGps.GetFix(myFix)
rc = myGps.Close()
```

ContentsAllowed

Applies to

OLE controls

PocketBuilder	✗
PowerBuilder	✓

Description

Specifies whether the OLE object in the control must be embedded or linked or whether either type of OLE object can be inserted at runtime.

ControlMenu

Applies to

Windows and DataWindow controls

PocketBuilder	✗
PowerBuilder	✓

PowerBuilder only
The ControlMenu property is not available in the Window painter in PocketBuilder. Use the Close and SmartMinimize properties instead.

Description

The Control Menu property specifies whether the Control Menu box appears in the title bar of the Window or DataWindow control.

CornerHeight

Applies to

RoundRectangle controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description The Corner Height property sets the radius of the vertical part of the corners of a RoundedRectangle control. The radius is in PowerBuilder units

Usage

In a painter

❖ **To set the CornerHeight property:**

- Enter an integer in the CornerHeight field on the General page of the control's Properties view.

In scripts

The CornerHeight property takes an integer. This example sets the corner height for a RoundedRectangle rr_1.

```
rr_1.CornerHeight = 10
```

CornerWidth

Applies to

RoundedRectangle controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description The Corner Width property sets the radius of the horizontal part of the corners of a RoundedRectangle control. The radius is in PowerBuilder units

Usage

In a painter

❖ **To set the CornerWidth property:**

- Enter an integer in the CornerWidth field on the General page of the control's Properties view.

In scripts

The CornerWidth property takes an integer. This example sets the corner width for a RoundedRectangle rr_1.

```
rr_1.CornerWidth = 10
```

CreateOnDemand

Applies to	Tab controls				
	<table><tr><td>PocketBuilder</td><td>✗</td></tr><tr><td>PowerBuilder</td><td>✓</td></tr></table>	PocketBuilder	✗	PowerBuilder	✓
PocketBuilder	✗				
PowerBuilder	✓				
Description	When CreateOnDemand is enabled, 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 either by clicking on the tab page or by calling the SelectTab function (the SelectTab function sets the SelectedTab property).				

DataObject

Applies to	DataWindow controls and DataStore objects						
	<table><tr><td>PocketBuilder on Pocket PC</td><td>✓</td></tr><tr><td>PocketBuilder on Smartphone</td><td>✓</td></tr><tr><td>PowerBuilder</td><td>✓</td></tr></table>	PocketBuilder on Pocket PC	✓	PocketBuilder on Smartphone	✓	PowerBuilder	✓
PocketBuilder on Pocket PC	✓						
PocketBuilder on Smartphone	✓						
PowerBuilder	✓						
Description	The DataObject property specifies the name of the DataWindow object or Report object associated with the DataWindow control or DataStore.						
Usage	In a painter ❖ To specify the name of the DataWindow object contained in a DataWindow control: Enter the name of an existing DataWindow object in the DataObject field on the General property page of the DataWindow control's Properties view, or use the Browse button to select an object. In scripts The DataObject property takes a string. This example specifies d_employ as the DataWindow object in the DataWindow control dw_1.						

```
dw_1.DataObject = "d_employ"
```

Reinitializing the DataWindow control or DataStore

Setting the DataObject to an empty string reinitializes the DataWindow control or DataStore and removes all storage associated with the previous DataObject.

DataSource

Applies to

DialingDirectoryEntry objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Indicates the phone number source. Values are:

- **0** Unspecified or unknown number
- **1** Pocket Outlook contact list
- **2** Pocket Outlook mobile number
- **3** Pocket Outlook business number
- **4** Pocket Outlook home number
- **10** General SIM list
- **11** Emergency dialing list
- **12** Fixed dialing list
- **13** Last dialing list
- **14** Private numbers list

Usage

In scripts

The DataSource property takes an integer value. The following example uses a CASE statement to obtain the appropriate string from the enumerated variable for a given entry and writes it to a list box:

```
// Instance variable:
// DialingDirectoryEntry iDialingDirEntries[]

integer iDataSource
String strDataSource
```

```
iDataSource = iDialingDirEntries[index].DataSource
choose case iDataSource
  case 0
    strDataSource = String("Unspecified or Unknown")
  case 1
    strDataSource = String("PO Contact List")
  case 2
    strDataSource = String("PO Mobile Number")
  case 3
    strDataSource = String("PO Business Number")
  case 4
    strDataSource = String("PO Home Number")
  case 10
    strDataSource = String("SIM - General List")
  case 11
    strDataSource = String("SIM - Emergency Dial List")
  case 12
    strDataSource = String("SIM - Fixed Dialing List")
  case 13
    strDataSource = String("SIM - Last Dialing List")
  case 14
    strDataSource = String("SIM - Own Numbers List")
  case else
    strDataSource = String("Unknown")
end choose
lb_1.AddItem("DataSource: " + strDataSource)
```

DataType

Applies to

grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

DataType is a property of the grAxis objects that can be part of graph controls. There are three grAxis objects: Category, Series, and Values.

DataType specifies the type of data that is assigned to the axis. Possible datatypes are Number, Date, DateTime, Default, Double, Text, and Time.

Usage**In a painter****❖ To specify datatype:**

- 1 Display the Axis tab page of the graph control's Properties view.
- 2 Select the desired Axis type from the Axis drop-down list.
- 3 Select the desired datatype from the DataType drop-down list.

In scripts

The DataType property takes a value of type `grAxisDataType`.

The following line sets the DataType of the Values axis of a graph.

```
gr_1.Values.DataType = AdtDate!
```

Default

Applies to

CommandButton, PictureBox, OLECustomControl controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The Default property specifies that the control is the default button. If Default is TRUE, the selected control has a thick border and receives a Clicked event when the user presses Enter (unless the user has tabbed to another control). If Default is FALSE, the control is not the default and pressing Enter does not affect it unless the user tabs to it.

Setting focus

If the window contains an editable field, such as a MultiLineEdit, then the default button behaves as expected (receives the Clicked event when the user presses Enter) when focus is on the editable field. When the user presses Tab to move focus to another button (not the default), pressing Enter fires the clicked event for the button that currently has focus.

If the window does not contain an editable field, use SetFocus or tab order to make sure the default button behaves as documented above.

You can make a `CommandButton`, `PictureButton`, or `OLECustomControl` control the default button so that it responds to the Enter key. If you check `Default` for more than one control, the last one set acts as the default.

Usage

In a painter

- ❖ **To enable the `Default` property**
 - Select the `Default` check box on the General page of the control's Properties view.

In scripts

The `Default` property takes a boolean value. To set a `PictureButton` as the default button, use a line like the following.

```
pb_1.Default = TRUE
```

DefaultSize

Applies to

Windows

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The `DefaultSize` property changes the size of the current window to the default size specified on the Size page of the Design Options dialog box in the Window painter.

Usage

In a painter

- ❖ **To change the size of the current window to the default:**
 - Select the `DefaultSize` check box on the General page of the window's Properties view.

You specify the default size for all main windows on the Size page of the Design Options dialog box. You can select the defaults for Portrait or Landscape orientation or specify a custom width and height.

All new windows are created as main windows with the width and height you specify in the Design Options dialog box and the `DefaultSize` check box selected. If you change the window type to Response, the `DefaultSize` check box is cleared and the size of the window changes to a size suitable for response windows.

If you change the size of a window on the Other property page, the DefaultSize check box is not cleared. To return to the default size, clear the check box then reselect it.

In scripts

This property cannot be set in a script.

Degree

Applies to

GPSCoordinate objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The Degree property specifies the position of a fix in degrees.

Usage

In scripts

The Degree property takes an integer value in the range 0 to 179:

```
integer li_degrees
GPSCoordinate myCoord
li_degrees = myCoord.Degree
```

See also

Latitude
Longitude

DeleteItems

Applies to

ListView, TreeView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description	When the DeleteItems property is TRUE, the user can delete items from the ListView or TreeView with the Delete key. When DeleteItems is FALSE, the user cannot delete items.
Usage	In a painter ❖ To allow users to delete items from the control: • Select the DeleteItems check box on the General page of the control's Properties view. In scripts The DeleteItems property takes a boolean value. The following example disables deletion of ListView items by the user: <pre>lv_1.DeleteItems = FALSE</pre>

Depth

Applies to	Graph controls <table><tr><td>PocketBuilder on Pocket PC</td><td>✓</td></tr><tr><td>PocketBuilder on Smartphone</td><td>✓</td></tr><tr><td>PowerBuilder</td><td>✓</td></tr></table>	PocketBuilder on Pocket PC	✓	PocketBuilder on Smartphone	✓	PowerBuilder	✓
PocketBuilder on Pocket PC	✓						
PocketBuilder on Smartphone	✓						
PowerBuilder	✓						
Description	For 3-D graphs, specifies the depth of the graph as a percentage of its width. The default is 100 percent.						
Usage	In a painter ❖ To set the Depth property: 1 Display the General page of the graph control's Properties view. 2 Select a 3-D type of graph from the GraphType drop-down list. 3 Use the Depth slider control to set the Depth to the desired percentage of width. In scripts The Depth property takes an integer. This example specifies that the depth of the graph is 50% of its width: <pre>gr_1.Depth = 50</pre>						

DeviceHandle

Applies to

Camera and SerialGPS objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

DeviceHandle is a read-only property that specifies the file handle of the camera or GPS device.

Usage

In scripts

The DeviceHandle property is a read-only property that takes an unsigned long value.

DisabledName

Applies to

PictureButton controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The DisabledName property specifies the name of a picture file to be displayed when the PictureButton is disabled. The picture can be in the following formats:

- bitmap (*.BMP*)
- GIF (*.GIF*)
- JPEG (*.JPG* or *.JPEG*)

Usage

In a painter

- ❖ **To specify a picture to be displayed when the button is disabled:**
 - Enter the name of the file in the DisabledName field on the General page of the control's Properties view, or use the Browse button next to the DisabledName field to select a file.

In scripts

The DisabledName property takes a string containing the name of a file. The string can include the path. This example specifies the picture *controls.bmp* for the disabled view of the PictureBox:

```
pb_1.DisabledName = "d:\pbhelp\controls.bmp"
```

DisableDragDrop

Applies to

TreeView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The DisableDragDrop property determines whether events for dragging, such as BeginDrag, are triggered when the user clicks on an item within the control and drags. DisableDragDrop only affects the dragging of items within the control.

When DisableDragDrop is TRUE, no drag events occur when the user tries to drag an item. To implement drag and drop, write scripts for the appropriate dragging events.

DisableDragDrop also affects when selection occurs. When it is TRUE, an item the user clicks is selected when the mouse button is pressed down. When it is FALSE, the item is selected when the mouse button is released.

Usage

In a painter

- ❖ **To disable drag and drop within the TreeView control:**
 - Select the DisableDragDrop check box on the General page of the control's Properties view.

In scripts

The DisableDragDrop property takes a boolean value. The following example prevents drag events from being triggered within a TreeView control.

```
tv_1.DisableDragDrop = TRUE
```

DisableNoScroll

Applies to

ListBox and PictureListBox controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

The DisableNoScroll property specifies the behavior of a scroll bar in a list box. If the property is enabled, the scroll bar is always visible, but is disabled when all the items can be accessed without it. If the property is disabled, the scroll bar is displayed only if it is necessary, based on the number of items and the height of the ListBox or PictureListBox.

Smartphone platform

On Smartphone platforms, ListBox controls are converted to spinner controls. The spinner control does its own scroll management, and a scroll bar displays only if there are more items than can be accommodated on a single screen.

Usage

In a painter

- ❖ **To make the scroll bar always visible but disabled when not needed:**
 - Select the DisableNoScroll check box on the General page of the control's Properties view

In scripts

The DisableNoScroll property takes a boolean value. This example for a ListBox displays the scroll bar only when needed:

```
lb_1.DisableNoScroll = FALSE
```

DisplayEveryNLabels

Applies to

grAxis objects of Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

DisplayEveryNLabels is a property of the grAxis objects that can be part of graph controls. There are three grAxis objects: Category, Series, and Values.

This property specifies which major divisions to label on the selected axis in the graph. For example, a value of 2 means to label every other tick mark. Use 0 to let the graph select the optimum number of labels to use.

Usage

In a painter

❖ **To specify the number of major divisions to label:**

1 Display the Axis tab page in the graph’s Properties view.

2 Select the desired Axis from the Axis drop-down list.

3 Use the spin control to select a number from 0 to 100 in the DisplayEveryNLabels field.

In scripts

The DisplayEveryNLabels property takes an integer. The following example sets labeling at every 10 tick marks for the Series Axis:


```
gr_1.Series.DisplayEveryNLabels = 10
```

DisplayExpression

Applies to

grDispAttr objects within Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The DisplayExpression property specifies an expression whose value is the label for a specified text object within the Graph control.

The default expression is the value of the property containing the text for the graph component.

Usage

In a painter

❖ **To specify a display expression for a text object:**

1 Display the Text tab page of the Graph control’s Properties view.

- 2 Select the text object for which you want to define a display expression from the Text Object list box.

The default value of the DisplayExpression property is displayed in the DisplayExpression field.

- 3 Specify the display expression in the Display Expression field, or click the More button to display the Modify Expression dialog box.

In scripts

The DisplayExpression property can be set via the grDispAttr object for each text component. DisplayExpression takes a string, which can contain an expression.

The following example appends today's date to the title of the graph:

```
gr_1.TitleDispAttr.DisplayExpression = 'title + " " +  
Today()'
```

DisplayName

Applies to

Application object, OLE controls

PocketBuilder	✗
PowerBuilder	✓

Description

DisplayName is a user-readable name for your application or OLE control. This name is displayed in OLE dialog boxes and windows that show the object's name. If you do not specify a value, the name of the control (such as ole_1) or application (value of the AppName property) is used for Display Name.

DisplayOnly

Applies to

MultiLineEdit, SingleLineEdit, RichTextEdit, EditMask controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description	When the Display Only property is enabled, users cannot change the text in an editable control. If the property is not enabled, users can change the text.
Usage	In a painter ❖ To specify that text is display only: • Select the Display Only check box on the General page of the control's Properties view. In scripts The DisplayOnly property takes a boolean value. The following example specifies that text in a MultiLineEdit control cannot be changed: <pre>mle_1.DisplayOnly = TRUE</pre>

DisplayType

Applies to	OLE controls				
	<table><tr><td>PocketBuilder</td><td>✗</td></tr><tr><td>PowerBuilder</td><td>✓</td></tr></table>	PocketBuilder	✗	PowerBuilder	✓
PocketBuilder	✗				
PowerBuilder	✓				
Description	The DisplayType property specifies how the OLE object is displayed in the control. The control can display the actual contents or an icon to represent the object.				

DocumentName

Applies to	RichTextEdit controls				
	<table><tr><td>PocketBuilder</td><td>✗</td></tr><tr><td>PowerBuilder</td><td>✓</td></tr></table>	PocketBuilder	✗	PowerBuilder	✓
PocketBuilder	✗				
PowerBuilder	✓				
Description	The Document Name property specifies the name that will appear in the print queue when the user prints the contents of the control.				

DragAuto

Applies to

Draggable controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The Drag Auto property determines whether PocketBuilder puts the control into drag mode automatically. If the property is enabled, when the user clicks the control and starts dragging it, PocketBuilder puts the control in drag mode. Clicking the control triggers a DragDrop event, not a Clicked event.

If Drag Auto is not enabled, then when the user clicks the control, PocketBuilder does not put the control in drag mode. You have to call the Drag function to put the control into drag mode.

Usage

In a painter

❖ To set DragAuto:

- Select the Drag Auto check box on the Other tab page of the control's Properties view.

In scripts

Most controls have a DragAuto property. It takes a boolean value. The following example sets drag mode for a CommandButton.

```
cb_1.DragAuto = TRUE
```

DragIcon

Applies to

Draggable controls

PocketBuilder	✕
PowerBuilder	✓

Description

The DragIcon property specifies the icon to display when the user drags the control. The default icon is a box the size of the control.

DriverName

Applies to

Camera objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

DriverName is a read-only value defined by the camera manufacturer. This property can be empty. If the required driver is not found, this value might provide the name of the driver.

Usage

In scripts

The DriverName property is a string that, if provided by the vendor, can be used in an attempt to find the driver. This example tests whether the string is empty and, if it is not, writes the driver name to a single line edit box:

```
cam_1.Open(w_myphoto_main)
if cam_1.drivename = "" then
    sle_dname.text = "No driver name specified."
else
    sle_dname.text = cam_1.drivename
end if
```

DriverVersion

Applies to

Camera objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

DriverVersion is a read-only value defined by the camera manufacturer. This property can be empty. If the required driver is not found, this value might provide the version of the driver.

Usage

In scripts

The `DriverVersion` property is a string that, if provided by the vendor, can be used in an attempt to find the driver. This example tests whether the string is empty and, if it is not, writes the driver version to a single line edit box:

```
cam_1.Open(w_myphoto_main)
if cam_1.driverversion = "" then
    sle_dversion.text = "No driver version specified."
else
    sle_dversion.text = cam_1.driverversion
end if
```

DropLines

Applies to

grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

A drop line is a line that extends from a data point to its axis. Drop lines are not available for all graph types.

Usage

In a painter

❖ **To set the drop line type:**

- 1 Display the Axis tab page of the graph control's Properties view.
- 2 Select the desired Axis from the Axis drop-down list.
- 3 Select the type of line desired from the DropLines drop-down list.

In scripts

The `DropLines` property takes a value of the `LineStyle` enumerated datatype. The following example sets dashed lines for the drop lines in the Series axis:

```
gr_1.Series.DropLines = Dash!
```

Dropped

Applies to

CallLogEntry objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Indicates whether a call was dropped by the service provider or ended normally. Values are:

- **true** The call was dropped by the provider
- **false** The call ended without service interruption

Usage

In scripts

The Dropped property is a boolean indicating whether an entry in the call log was dropped by the service provider. In this example, a list box is populated with the phone numbers in a call log. The following lines in the SelectionChanged event of the list box get the value of the Dropped property of the selected phone number and write it to a second list box:

```
// instance variable: CallLogEntry iLogEntries[]

if (iLoggedEntries[index].dropped = true
    lb_2.AddItem("Call Dropped")
else
    lb_2.AddItem("Call Completed")
end if
```

Duration

Applies to

NotificationBubble objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Description

Defines the length of time in seconds that the notification bubble displays.

Usage**In a painter**❖ **To select a duration for the display of a notification bubble:**

- Enter the duration you want in the Duration text box on the General page of the NotificationBubble object's Properties view.

In scripts

The Duration property takes an integer. The following example sets the duration for a notification bubble to 5 seconds:

```
nb_myBubble.Duration = 5
```

EditLabels

Applies to

ListView and TreeView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When EditLabels is enabled, the user can edit labels in the ListView or TreeView by selecting the item, clicking on the label, and then adding or deleting characters. When EditLabels is not enabled, the labels are not editable.

Usage**In a painter**❖ **To enable editing of labels:**

- Select the Edit Labels check box on the General page of the control's Properties view.

In scripts

The EditLabels property takes a boolean value. The following example enables editing of labels in a TreeView.

```
tv_1.EditLabels = TRUE
```

Elevation

Applies to

GPSSatellitePosition objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

For GPSSatellitePosition objects The Elevation property indicates the elevation of the satellite. The range of values is 0 to 90.

For Graph controls Elevation determines how much of the full perspective of a 3D graph is visible. It specifies the angle of front-to-back elevation.

Elevation is disabled for 2D graphs.

Usage

In scripts (GPSSatellitePosition objects)

The Elevation property takes an integer value:

```
integer li_elevation
GPSSatellitePosition myGPS_SP

...
li_elevation = myGPS_SP.Elevation
```

In a painter (graph controls)

- ❖ **To change the elevation of a 3-D graph:**
 - Move the Elevation slider on the General page of the graph's Properties view.

In scripts (graph controls)

The Elevation property takes an integer value. The following example specifies an elevation of 35.

```
gr_1.Elevation = 35
```

Enabled

Applies to

All graphic controls except drawing objects and progress, scroll, and track bars.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When the Enabled property is enabled, the control can have focus. Users can select the control by clicking on it. If the control is included in the tab order, users can tab to it.

If the Enabled property is not enabled, the control cannot have focus and the user cannot select it.

Enabled does not affect whether the control is visible (see Visible).

Usage

In a painter

❖ To set the Enabled property:

- Select the Enabled check box on the General page of the control's Properties view.

In scripts

The Enabled property takes a boolean value. Most controls have an Enabled property. This example sets Enabled for a CommandButton.

```
cb_1.Enabled = TRUE
```

Encoding

Applies to

SMSProtocol objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Specifies the data encoding used for outgoing text messages in SMSSession Send function calls. Values are:

- SMSDE_Optimal!** The data encoding that represents all characters in the least space (default and recommended value)
- SMSDE_GSM!** The default Global System for Mobile Communications (GSM) 7-bit encoding specified in GSM specification 03.38
- SMSDE_UCS2!** Unicode UCS-2 encoding

Usage

In scripts

The Encoding property takes a value of the SMSDataEncoding enumerated variable. This example sets the address type for an international message:

```
SMSProtocol mysmsprotocol
mysmsprotocol.Encoding = SMSDE_gsm!
```

EndTime

Applies to

CallLogEntry objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Indicates when the call ended.

Usage

In scripts

The EndTime property takes a DateTime value. The following example writes the start and end times of a call log entry to a multiline edit box:

```
// Integer idx passed into function
CallLog l_log
CallLogEntry l_logentry

l_log = CREATE CallLog
l_logentry = l_log.getEntry (idx)
mle_1.text = "Start time: " + &
    string(l_logentry.StartTime)
mle_1.text += "End time: " + string &
    (l_logentry.EndTime)
```

EndX

Applies to

Line controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The EndX property specifies the X coordinate of the end of the line in PowerBuilder units.

The X coordinate is the distance from the left edge of the window or custom user object. If the object is a main window or custom user object, the distance is relative to the screen. If it is not a main window, the distance is relative to the parent window.

Usage

In a painter

❖ **To set the X coordinate of the end of the line:**

- Insert the line. If you want to change the ending location, change the value of the EndX field on the Position tab page of the line's Property view.

In scripts

The EndX property takes an integer value. This example sets the X coordinate of the end of the line:

```
ln_1.EndX = 1200
```

EndY

Applies to

Line controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The EndY property specifies the Y coordinate of the end of the line in PowerBuilder units.

The Y coordinate is the distance from the top edge of the window or custom user object. If the object is a main window or custom user object, the distance is relative to the screen. If it is not a main window, the distance is relative to the parent window.

Usage

In a painter

- ❖ **To set the Y coordinate of the end of the line:**
 - Insert the line. If you want to change the ending location, change the value of the EndY field on the Position tab page of the line’s Property view.

In scripts

The EndY property takes an integer value. This example sets the Y coordinate of the end of the line:

```
ln_1.EndY = 1200
```

Escapement

Applies to

grDispAttr objects in a graph control

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

This property specifies the rotation for the baseline of the Axis text objects in a graph control.

Usage

In a painter

- ❖ **To set the rotation of text objects within a graph:**
 - 1 Display the Text tab of the graph’s Properties view.
 - 2 Select the desired text object from the Text Object list box.
 - 3 Use the Escapement spin control to set the desired value.

In scripts

Escapement takes an integer value that specifies the rotation in tenths of a degree. 0 is horizontal. A value of 900 rotates the text 90 degrees; 450 rotates the text 45 degrees. The following example sets the rotation of the Value Axis Label to 90 degrees:

```
gr_1.Value.LabelDispAttr.Escapement = 900
```

ExtendedSelect

Applies to

ListBox and ListView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

ExtendedSelect specifies whether users can select more than one item in a ListBox or ListView at one time. When ExtendedSelect is enabled, users can select multiple items by clicking on an item and dragging the mouse up or down to select items, using Click or Shift+Click to select a sequential group of items, or using Control+Click on multiple items. When ExtendedSelect is not enabled, users cannot select multiple items.

PocketBuilder applications

When ExtendedSelect is enabled, PocketBuilder applications users can select multiple items with the stylus while pressing the Shift or Control keys on the SIP.

Usage note

If both MultiSelect and ExtendedSelect are enabled, then the behavior of ExtendedSelect takes precedence.

Usage

In a painter

- ❖ **To enable extended select:**
 - Select the Extended Select check box on the General page of the control's property page

In scripts

The ExtendedSelect property takes a boolean value. The following example lets the user select multiple items using extended selection techniques for a ListBox lb_1:

```
lb_1.ExtendedSelect = TRUE
```

FaceName

Applies to

Controls that can display text

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The FaceName property specifies the typeface used for text in the control. For tab controls, the property specifies the typeface for text labels on tabs.

The typefaces available for your use are those installed on your system. Keep in mind that the fonts available to you may not be available where you deploy your application.

Usage

In a painter

- ❖ **To set the typeface of text in a control:**
 - Select a font type from the FaceName list box on the Font tab page of the control's Properties view.
- ❖ **To set the typeface of text objects in a graph control:**
 - 1 Display the Text tab page of the graph control's Properties view.
 - 2 Select the desired text object from the Text Object list box.
 - 3 Select a font from the FaceName list box.

In scripts

The FaceName property takes a string value. The following example sets the font for text labels on tab pages of a tab control to the Arial typeface:

```
tab_1.FaceName = "Arial"
```

FillColor

Applies to

Oval, Rectangle, RoundedRectangle controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The FillColor property defines the color used to fill the control. When you are defining the background color in a painter, some of the choices take their values from the current Windows color scheme or from custom colors.

To add your own colors to the color drop-down list, select Design>Custom Colors before displaying the Properties view.

Usage

In a painter

❖ To set the fill color:

- Select the desired color from the Fill Color drop-down list on the General page of the control's Properties view.

In scripts

The FillColor property takes a long value (-2 to 16,777,215) that specifies the numerical value of the background color of windows and other objects. The FillColor value is a combination of values for the red, green, and blue components of the color. If you do not know the long value for the color, choose Design>Custom Colors to determine the red, green, and blue values and then call the RGB function to specify the color in a script.

This example specifies yellow as the fill color for the RoundedRectangle rr_display.

```
rr_display.FillColor=RGB(255,255,0)
```

FillPattern

Applies to

Oval, Rectangle, RoundedRectangle, StaticText, and StaticHyperLink controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description	The FillPattern property specifies the hatch pattern used to fill the control. For drawing objects, the pattern uses the FillColor for the background and the LineColor for the foreground lines. For StaticText controls, the pattern uses the BackColor for the background and the TextColor for the foreground lines. The text and the pattern lines use the same color.
Usage	In a painter ❖ To set the fill pattern: • Select the desired hatch pattern from the Fill Pattern drop-down list on the General page of the control's Properties view. In scripts The FillPattern property takes a value of the FillPattern enumerated datatype. The following example sets a diamond fill for a StaticText control: <pre>st_1.FillPattern = Diamond!</pre>

FixedLocations

Applies to	ListView controls <table><tr><td>PocketBuilder on Pocket PC</td><td>✓</td></tr><tr><td>PocketBuilder on Smartphone</td><td>✓</td></tr><tr><td>PowerBuilder</td><td>✓</td></tr></table>	PocketBuilder on Pocket PC	✓	PocketBuilder on Smartphone	✓	PowerBuilder	✓
PocketBuilder on Pocket PC	✓						
PocketBuilder on Smartphone	✓						
PowerBuilder	✓						
Description	When the FixedLocations property is enabled, the user cannot drag items to new positions in the control. When Fixed Locations is not enabled and DragAuto is enabled, the user can drag items to new positions.						
Usage	In a painter ❖ To set the FixedLocation property: • Select the Fixed Locations check box on the General page of the control's Properties view.						

In scripts

The `FixedLocations` property takes a boolean value. The following example enables dragging of items within a `ListView`.

```
lv_1.DragAuto = TRUE
lv_1.FixedLocations = FALSE
```

FixedWidth

Applies to

Tab controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

When the `FixedWidth` property is enabled, tabs have a fixed width. The width is determined by the longest tab label. When `FixedWidth` is not enabled, tabs shrink to the length of their text labels.

Usage

In a painter

❖ **To set the `FixedWidth` property:**

- Select the `Fixed Width` check box on the General page of the tab control's Properties view.

In scripts

The `FixedWidth` property takes a boolean value. The following example specifies that tabs in the control `tab_1` have a fixed width.

```
tab_1.FixedWidth = TRUE
```

FixQuality

Applies to

GPSTFix objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Indicates whether the position fix uses the standard Global Positioning System (GPS) or Differential GPS (DGPS). DGPS has an additional correction signal that improves the accuracy of the GPS. Values are:

- **0** Invalid
- **1** GPS fix
- **2** DGPS fix

Usage

In scripts

The FixQuality property is an integer that indicates the quality of the fix. In this example, the SerialGPS object retrieves information about the current fix and, based on that information, writes an appropriate string value to a single line edit box:

```
integer li_quality

li_quality = myFix.FixQuality
CHOOSE CASE li_quality
CASE 0
    sle_quality.text = "Invalid fix"
CASE 1
    sle_quality.text = "GPS fix"
CASE 2
    sle_quality.text = "DGPS fix"
END CHOOSE
```

FixTime

Applies to

GPSTime objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Indicates the time at which the current position fix was read using Coordinated Universal Time (UTC). UTC is approximately the same as Greenwich Mean Time (GMT).

Usage

In scripts

The FixTime property takes a value of the Time datatype. The following lines create a SerialGPS object, retrieve information about the current fix, and write the fix time to a single line edit box:

```
SerialGps myGPS
GPSTime myFix
Integer rc

MyGPS = CREATE SerialGPS
rc = myGPS.Open()
...
rc = MyGPS.GetFix(myFix)
sle_time.txt = "Recorded at: " + String(myFix.FixTime)
```

Flags

Applies to

NotificationBubble objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Description

Sets behavior changes for the notification bubble. Values are:

Value	Windows CE value	Description
0	—	(Default) No changes
1	Straight to tray	Hides initial display of the notification bubble, displaying it only as an icon
2	Critical	Highlights notification bubble borders and title
16	Display on	Forces the display to turn on for the notification (Pocket PC 2003 only)
32	Silent	Forces the notification to be silent and not vibrate regardless of system settings (Pocket PC 2003 only)

The value you set for the Flags property is additive. A value of 1, 3, 17, 19, 33, 35, 49, or 51 overrides the InitiallyIconic property, because the Flags value of 1 is set, meaning that the notification bubble will not display initially when a notification update occurs. A value of 16 or greater is valid only for Pocket PC 2003 operating systems.

Usage

In a painter

❖ To set flags for a notification bubble:

- Enter the number you want in the Flags text box on the General page of the control's Properties view.

In scripts

The Flags property takes a long. The following example sets the Flags property for a notification bubble to 3:

```
nb_myBubble.Flags = 3
```

In this example, the notification bubble is not displayed initially and when it is displayed, the bubble borders and title are highlighted to indicate a critical message.

FocusOnButtonDown

Applies to

Tab controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

When the FocusOnButtonDown property is enabled, each tab page gets focus when the user clicks on it. A dotted rectangle marks the tab page. If FocusOnButtonDown is not enabled, the clicked tab page does not display the focus rectangle. In either case, the selected tab page comes to the front.

Usage

In a painter

❖ To set the FocusOnButtonDown property:

- Select the Focus On Button Down check box on the General page of the tab control's Properties view.

In scripts

The FocusOnButtonDown property takes a boolean value. The following example specifies that tab pages within the control tab_1 display the focus rectangle when clicked.

```
tab_1.FocusOnButtonDown = TRUE
```

FocusRectangle

Applies to

Graph, Picture, PictureHyperLink, StaticText, and StaticHyperLink controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When the FocusRectangle property is enabled, a dotted rectangle (the focus rectangle) displays when the control has focus. If this property is not enabled, the focus rectangle does not appear.

Usage

In a painter

- ❖ **To set the FocusRectangle property:**
 - Select the Focus Rectangle check box on the General page of the control's Properties view.

In scripts

The FocusRectangle property takes a boolean value. The following example specifies that a focus rectangle will appear when the StaticText control has focus.

```
st_1.FocusRectangle = TRUE
```

Folder

Applies to

Camera objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The Folder property must be set for connections to HTC camera devices before you call the Open function. This property is ignored for connections to HP or VEO camera devices.

You set the Folder property value to the directory on the Windows CE device where you want to store your photos for the current Camera object.

Usage

In a painter

- ❖ **To set the Folder property**
 - 1 Select the Camera object in the Non-Visual Object List and open the Properties view.
 - 2 Select the CameraType you want from the drop-down list.
 - 3 Type the path to the directory on a Windows CE device that you want to use to store camera images.

In scripts

The Folder property takes a string. This example saves images snapped by an HTC camera to the \Program Files\Photos directory:

```
cam_1.CameraType = 81 //specifier for HTC camera
cam_1.Folder="\Program Files\My Photos"
cam_1.Open(w_myphoto_main)
```

FontCharSet

Applies to

Controls that can display text

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

This property specifies the font character set to be used for the text in the control. Character sets and font typefaces are related, so choosing the wrong character set can cause a different font to be used from what is expected.

When working in a painter, setting the font face name property causes the correct character set to be selected.

Usage

In a painter❖ **To set the font character set:**

- Select a font character set from the FontCharSet list box on the Font tab page of the control's Properties view.

❖ **To set the font character set of text objects in a graph control:**

- 1 Display the Text tab page of the graph control's Properties view.
- 2 Select the desired text object from the Text Object list box.
- 3 Select a font character set from the FontCharSet list box.

In scripts

The FontCharSet takes a value of the FontCharSet enumerated datatype. The following example sets the character set for a static text control to ANSI:

```
st_1.FontCharSet = ANSI!
```

FontFamily

Applies to

Controls that can display text

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The FontFamily property sets the type style used for the text in the control.

When working in a painter, setting the font face name causes the correct font family to be selected.

Usage

In a painter

❖ **To set the font type style:**

- Select a font style from the FontFamily list box on the Font tab page of the control's Properties view.

❖ **To set the font type style of text objects in a graph control:**

- 1 Display the Text tab page of the graph control's Properties view.
- 2 Select the desired text object from the Text Object list box.
- 3 Select a font style from the FontFamily list box.

In scripts

The FontFamily property takes a value of the FontFamily enumerated datatype. The following example sets the FontFamily for a static text control to Roman:

```
st_1.FontFamily = Roman!
```

FontPitch

Applies to

Controls that can display text

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The FontPitch property specifies the spacing of the font used for the text in the control.

Usage

When working in a painter, setting the font face name causes the correct font pitch to be selected.

In a painter❖ **To set the font spacing:**

- Select a font spacing from the FontPitch list box on the Font tab page of the control's Properties view.

❖ **To set the font spacing of text objects in a graph control:**

- 1 Display the Text tab page of the graph control's Properties view.
- 2 Select the desired text object from the Text Object list box.
- 3 Select a font spacing from the FontPitch list box.

In scripts

The FontPitch property takes a value of the FontPitch enumerated datatype. The following example sets the font pitch for a static text control:

```
st_1.FontPitch = Fixed!
```

Format

Applies to

grDispAttr objects in a graph control

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The Format property allows you to define display formats for text objects in graphs. Display formats are masks in which certain characters have special significance.

The characters you use for formatting depend on the datatype of the data. PocketBuilder supports four kinds of display formats:

- Numbers
- Strings
- Dates
- Times

You can specify colors in any display format by specifying a color keyword before the format.

For more information about using colors and each kind of display format, see “Using colors with display formats for PocketBuilder controls” on page 563 and the sections that follow it. For more information about defining display formats, see the *User’s Guide*.

Usage

In a painter

- ❖ **To set the display format for a text object:**
 - 1 Display the Text tab page of the graph control’s Properties view.
 - 2 Select a text object in the Text Object list box.
 - 3 Enter an expression in the DisplayExpression field or select a format from the Format drop-down list.

In scripts

Each type of display format uses special characters that have special meaning for that format. The Format property takes a string value composed of these special characters.

The following example specifies a format for numeric data that always displays three digits, with two decimal places:

```
gr__1.Values.DispAttr.Format = "0.00"
```

The following example specifies a string format for alphanumeric data.

```
gr_1.Category.dispAttr.Format = "@@@/AAA"
```

Frame

Applies to

grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The Frame property specifies the line style used in the frame for an axis of a 3D Graph. The frame is the side of the 3D box associated with the selected axis.

Usage

In a painter❖ **To set the Frame line style of an Axis:**

- 1 Select a 3D graph style on the General page of the graph control's Properties view. Not all 3D graph styles support the Frame property.
- 2 Display the Axis tab page of the Properties view and select the desired axis from the Axis drop-down list.
- 3 Select the desired line style from the Frame drop-down list in the Line Style group.

In scripts

The Frame property takes a value of the LineStyle enumerated datatype.

This example specifies a dashed line for the Series axis frame of Graph gr_1.

```
gr_1.Series.Frame = Dash!
```

FreeDBLibraries

Applies to

Application object

PocketBuilder	✗
PowerBuilder	✓

Description

Determines whether PowerBuilder libraries are held in memory after PowerBuilder disconnects from a database.

GeoidalHeight

Applies to

GPSFix objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description	Indicates the height of the geoid (approximate mean sea level) relative to the World Geodetic System (WGS 84) ellipsoid in meters. The WGS 84 ellipsoid is an earth-fixed global reference frame used as the theoretical sea level by GPS. A negative number indicates that the geoid is below the WGS 84 ellipsoid.
Usage	In scripts The GeoidalHeight property takes a real value that indicates the difference in height in meters between mean sea level and the WGS 84 ellipsoid. The following example retrieves information about the current fix and writes the difference in height between the geoid and the WSGS 84 ellipsoid to a single line edit box: <pre>li_rc = MyGPS.GetFix(myFix) sle_1.txt = "Geoidal separation: " + & String(myFix.GeoidalHeight) + "M"</pre>

GraphType

Applies to	Graph controls <table><tr><td>PocketBuilder on Pocket PC</td><td>✓</td></tr><tr><td>PocketBuilder on Smartphone</td><td>✓</td></tr><tr><td>PowerBuilder</td><td>✓</td></tr></table>	PocketBuilder on Pocket PC	✓	PocketBuilder on Smartphone	✓	PowerBuilder	✓
PocketBuilder on Pocket PC	✓						
PocketBuilder on Smartphone	✓						
PowerBuilder	✓						
Description	The GraphType property specifies the kind of graph: Area, Bar, Column, Line, Pie, or Scatter.						
Usage	In a painter ❖ To select the graph type: Select the type of graph desired from the Graph Type drop-down list on the General page of the graph's Properties view. The type of graph displayed in the control changes to show an example of the selected type. The type of graph you select affects what properties are available on other tabs.						

In scripts

The `GraphType` property takes a value of the `grGraphType` enumerated datatype. The following example defines the Graph `gr_1` as a 3D pie chart.

```
gr_1.GraphType=Pie3D!
```

GroundSpeed

Applies to

GPSHeading objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The `GroundSpeed` property specifies the speed over the ground in knots.

Usage

In scripts

The `GroundSpeed` property takes a real value:

```
real mygroundspeed
GPSHeading myGPSHeading

mygroundspeed = myGPSHeading.GroundSpeed
```

HasButtons

Applies to

TreeView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When `HasButtons` is enabled, PocketBuilder displays + and - buttons next to parent items. The buttons indicate if an item is expanded (-) or collapsed (+).

Usage

In a painter

- ❖ **To enable the display of buttons:**
 - Select the HasButtons check box on the General page of the control's Properties view.

In scripts

The HasButtons property takes a boolean value. The following line specifies that PocketBuilder will display + and - buttons in a TreeView.

```
tv_1.HasButtons = TRUE
```

HasLines

Applies to

TreeView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When the HasLines property is enabled, PocketBuilder connects tree items by lines.

Usage

In a painter

- ❖ **To enable connecting TreeView items:**
 - Select the HasLines check box on the General page of the control's Properties view.

In scripts

The HasLines property takes a boolean value. The following line specifies that PocketBuilder will display lines connecting tree items.

```
tv_1.HasLines = TRUE
```

HDOP

Applies to

GPSTFix objects, GPSSatellitesInView objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The Horizontal Dilution of Precision (HDOP) property indicates the level of confidence in the accuracy of measurements related to latitude and longitude based on current satellite geometry. A lower value indicates greater confidence.

Usage

In scripts

The HDOP property takes a real value. The following example assumes that an HDOP value of 2 or less is required:

```
li_return = MyGPS.GetFix(myFix)
if (myFix.HDOP <= 2) then
    // continue
else
    // take another reading
end if
```

HeaderFooter

Applies to

RichTextEdit controls

PocketBuilder	✗
PowerBuilder	✓

Description

The HeaderFooter property specifies whether the control has a header/footer section. You must write a menu or button script to allow users to display the header and footer editing panels.

Heading

Applies to

GPSTHeading objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The Heading property specifies the true heading, measured in degrees in relation to geographic north (a magnetic heading is measured in relation to magnetic north). The range of values is 0 to 359.9.

Usage

In scripts

The Heading property takes a real value:

```
real myheading
GPSTHeading myGPSTHeading

myheading = myGPSTHeading.Heading
```

Hemisphere

Applies to

GPSCoordinate objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The Hemisphere property specifies the hemisphere of a position fix. The values are from raw satellite feed and are not localized. For latitude, the values are:

- **N** North
- **S** South

For longitude, the values are:

- **E** East
- **W** West

Usage**In scripts**

The Hemisphere property takes a Char value:

```
char hemisphere  
GPSCoordinate myCoord  
hemisphere = myCoord.Hemisphere
```

See also

Latitude
Longitude

Height

Applies to

Visible controls, windows

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The Height property specifies the height of a control or window in PowerBuilder units.

Usage**In a painter****❖ To set the height of a control or window**

- Enter the desired height in the Height edit box on the Other tab page of the object's Properties view, or select the control or window and resize it with your cursor.

In scripts

The Height property takes an integer value specifying the height of an object in PowerBuilder units. The following example sets the height of a DataWindow control dw_1.

```
dw_1.Height = 750
```

It is illegal to resize a minimized or maximized sheet or frame. Changing the Width or Height property for a minimized or maximized window is not supported.

HideSelection

Applies to

SingleLineEdit, MultiLineEdit, ListView, TreeView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

If the HideSelection property is enabled, selected text does not stay selected (highlighted) when the control does not have focus. If this property is not enabled, selected text stays highlighted when the control loses focus.

Usage

In a painter

❖ To enable HideSelection:

- Select the Hide Selection check box on the General page of the control's Properties view.

In scripts

The HideSelection property takes a boolean value. The following example specifies that selected text in a SingleLineEdit is always highlighted.

```
sle_1.HideSelection = FALSE
```

HScrollBar

Applies to

DataWindow, DropDownListBox, ListBox, MultiLineEdit and RichTextEdit controls, user objects, and windows

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

When the HScrollBar property is enabled, a horizontal scroll bar appears when all of the data cannot be displayed at one time. If this property is not enabled, no horizontal scroll bar appears.

Usage

In a painter❖ **To set a horizontal scroll bar for controls:**

- Select the HScroll Bar check box on the General page of the control's Properties view.

❖ **To set a horizontal scroll bar for windows or user objects:**

- Select the HScroll Bar check box on the Scroll tab page of the window's or object's Properties view.

In scripts

The HScrollBar property takes a boolean value. The following example allows a horizontal scroll bar to appear when needed in a ListBox.

```
lb_1.HScrollBar = TRUE
```

HSplitScroll

Applies to

DataWindow controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

If the HSplit Scroll property is enabled, the user can split the DataWindow control into two panes with separate scroll bars. The user moves the split bar to divide the DataWindow control into two panes.

If this property is not enabled, the user cannot split the DataWindow control.

Usage

In a painter❖ **To allow splitting the control into two panes:**

- Select the HSplit Scrolling check box on the General page of the control's Properties view.

In scripts

The HSplitScroll property takes a boolean value. The following example allows splitting of a DataWindow control dw_1.

```
dw_1.HSplitScroll = TRUE
```

HTextAlign

Applies to

PictureButton controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The HTextAlign property specifies whether text in the PictureButton control is right aligned, left aligned, or centered horizontally.

Usage

In a painter

- ❖ **To set the horizontal alignment of text:**
 - Select the desired alignment from the Horizontal Alignment drop-down list on the General tab of the control’s Properties view, or use the Left, Right, and Center alignment buttons on the StyleBar.

In scripts

The HTextAlign property takes a value of the Alignment enumerated datatype.

The following example specifies right alignment for text in a PictureButton.

```
pb_1.HTextAlign = Right!
```

Icon

Applies to

DataWindow controls and windows

PocketBuilder	✕
PowerBuilder	✓

Description

The Icon property specifies the icon to display when the DataWindow control or window is minimized. You can specify a stock icon or any icon filename.

ID

Applies to

SMSSession objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

If the SMSSession Send function is successful, the ID property is populated with a value that identifies a specific message.

Usage

In scripts

The ID property takes a long value. This example checks that the Send function was successful, then, if the ID property is not null, writes it to a single line edit box:

```
li_ret = g_smsSess.Send(g_smsMsg, g_smsAddr)
if li_ret <> 1 then
    sle_1.text = "Send failed. Return value: " + &
        String(li_ret)
else
    if not IsNull(g_smsMsg.ID) then
        sle_1.text = string(g_smsMsg.ID)
    else
        sle_1.text = "Message ID is null"
    end if
end if
```

IgnoreDefaultButton

Applies to

EditMask, MultiLineEdit controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The IgnoreDefaultButton property specifies whether the Clicked event for the window's default button is triggered when user presses Enter.

When this property is enabled, pressing Enter does not trigger the clicked event, but instead adds a new line in the control.

When this property is not enabled, pressing Enter does trigger the clicked event and a new line is *not* added in the control.

Usage

In a painter

- ❖ **To set the IgnoreDefaultButton property:**
 - Select the Ignore Default Button check box on the General page of the control’s Properties view.

In scripts

The IgnoreDefaultButton property takes a boolean value. The default is FALSE.

The following example specifies that pressing Enter does not trigger the clicked event for the window’s default button and adds a new line in the MultiLineEdit control instead.

```
mle_1.IgnoreDefaultButton = TRUE
```

Increment

Applies to

EditMask controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When an EditMask control has been defined as a spin control (that is, a control with up and down arrows the user clicks to cycle through predefined values), the Increment property specifies the increment of the spin arrows.

Increment is only valid for numeric and date datatypes. In a date datatype, the increment applies only to the year.

Usage

In a painter

- ❖ **To set the increment of a spin control:**
 - 1 Select the Spin Control check box on the Mask tab page of the control’s Properties view.

The Increment field becomes active.
 - 2 Select the mask datatype from the Type drop-down list.
 - 3 Enter an increment value in the Spin Increment field.

In scripts

The Increment property takes a double. The following line specifies an increment of 10 for an EditMask.

```
em_1.Increment = 10.0
```

Indent

Applies to

TreeView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The Indent property specifies how far each level of the TreeView is indented. The numeric value you type is the indentation amount in PowerBuilder units.

Usage

In a painter❖ **To set the indentation of items in a tree view control:**

- Enter the amount of the indentation, in PowerBuilder units, in the Indentation field of the General page of the control's Properties view.

In scripts

The Indent property takes an integer value. The following script sets an indentation of 100 PowerBuilder units.

```
tv_1.Indent = 100
```

InitiallyIconic

Applies to

NotificationBubble objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Description

Determines whether a notification bubble displays initially as an icon, or whether the bubble itself displays when the notification event is triggered. Values are:

- TRUE – Displays initially as an icon
- FALSE – Displays initially as a notification bubble

In all cases, users can click the notification icon to display the bubble. The value set for this property can be overridden by a Flags property setting.

Usage

In a painter

❖ To select the notification bubble for initial display as an icon:

- Click the Initially Iconic check box on the General page of the NotificationBubble object’s Properties view.

In scripts

The InitiallyIconic property takes a boolean. The following example hides the notification bubble when a notification is sent and sets the initial display to an icon only:

```
nb_myBubble.InitiallyIconic = true
```

InputEditMode

Applies to

EditMask, ListView, ListBox, DropDownListBox, MultiLineEdit, SingleLineEdit, and TreeView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

For applications that you deploy to a Pocket PC device or emulator, you can use the InputEditMode property instead of the system SetSIPType function to set the SIP type for an edit control. On Smartphone platforms, you can use the InputEditMode property to set the Multipress or T9 edit modes for an edit control.

The default value of the `InputEditMode` property is 0. The default value does not change the current SIP type on a Pocket PC or the current edit mode on a Smartphone. The table that follows lists `InputEditMode` values that are particular to the Pocket PC or Smartphone platforms:

Values for the Smartphone keypad	Values for the Pocket PC SIP
1 = Numeric mode	11 = Keyboard mode
2 = T9 mode	12 = Jot mode
3 = Multipress lowercase mode	13 = Block mode
4 = T9 uppercase mode	14 = WordLogic mode
5 = T9 first letter uppercase	15 = Transcriber mode
6 = Multipress uppercase mode	16 = Fitaly SIP
7 = Multipress first letter uppercase	

Usage

In a painter

- ❖ **To change the input method edit mode for an edit control:**
 - Select the input method you want in the Input Edit Mode drop-down list on the Other page of the control's Properties view.

Do not set in script

Do not set this property in script. `InputEditMode` property values are cached in internal structures for performance reasons.

InputFieldBackColor

Applies to

RichTextEdit controls

PocketBuilder	✗
PowerBuilder	✓

Description

The `InputFieldBackColor` property sets the color for the background of input fields in the RichTextEdit control. This item can also be selected by the user at runtime from the Properties item of the pop-up menu.

InputFieldNamesVisible

Applies to	RichTextEdit controls				
	<table><tr><td>PocketBuilder</td><td>✗</td></tr><tr><td>PowerBuilder</td><td>✓</td></tr></table>	PocketBuilder	✗	PowerBuilder	✓
PocketBuilder	✗				
PowerBuilder	✓				
Description	When the InputFieldNamesVisible property is enabled, the control will display input field names rather than input field values. When this property is not enabled, the RichTextEdit control will display the input field values.				

Invert

Applies to	Picture and PictureHyperLink controls						
	<table><tr><td>PocketBuilder on Pocket PC</td><td>✓</td></tr><tr><td>PocketBuilder on Smartphone</td><td>✓</td></tr><tr><td>PowerBuilder</td><td>✓</td></tr></table>	PocketBuilder on Pocket PC	✓	PocketBuilder on Smartphone	✓	PowerBuilder	✓
PocketBuilder on Pocket PC	✓						
PocketBuilder on Smartphone	✓						
PowerBuilder	✓						
Description	If the Invert property is enabled, PocketBuilder will display the picture with its colors inverted. If this property is not enabled, the picture will appear in its normal color.						
Usage	In a painter ❖ To invert colors in a picture control: • Select the Invert Image check box on the General page of the control’s property page. In scripts The Invert property takes a boolean value. The following example specifies that a Picture control, p_1, will appear in its normal colors. <pre>p_1.Invert = FALSE</pre>						

IsFixValid

Applies to

GPSTFix objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The IsFixValid property determines whether the position fix data returned from the satellite is valid.

Usage

In scripts

The IsFixValid property takes a boolean value. The following example stops the execution of the script or function if the data returned from the satellite is invalid:

```
rc = MyGPS.GetFix(myFix)
IF NOT myFix.IsFixValid THEN RETURN -1
```

IsHeadingValid

Applies to

GPSHeading objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The IsHeadingValid property determines whether the heading data returned from the satellite is valid.

Usage

In scripts

The IsHeadingValid property takes a boolean value. The following example stops the execution of the script or function if the data returned from the satellite is invalid:

```
myHeading = MyGPS.GetHeading()
IF NOT myHeading.IsHeadingValid THEN RETURN -1
```

Italic

Applies to

Controls that display text

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

Italic is a property of text in a control.

Usage

In a painter

- ❖ **To italicize all text items in a control:**
 - Select the Italic check box on the Font tab page of the control’s property page, or select the control and then click the I button on the StyleBar
- ❖ **To italicize a text objects in a graph control:**
 - 1 Display the Text tab page of the graph control’s Properties view.
 - 2 Select the desired text object from the Text Object list box.
 - 3 Select the Italic check box.

In scripts

The Italic property takes a boolean value. The following example italicizes the text in a StaticText control:

```
st_1.Italic = TRUE
```

This example italicizes the label of the Value axis of a graph control.

```
gr_1.Values.LabelDispAttr.Italic = TRUE
```

InputFieldsVisible

Applies to

RichTextEdit controls

PocketBuilder	✗
PowerBuilder	✓

Description

When the InputFieldVisible property is enabled, input fields appear on the RichTextEdit control

Item[]

Applies to

ListView, ListBox, DropDownListBox, and Toolbar controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The Item property array specifies the items in the control. This array is not updated after initialization.

Usage

In a painter

❖ To add items to a control:

- Enter the items on the Items tab page of the control's Properties view.

In scripts

The Item[] property is an array of strings, but it is not updated after initialization. Use the AddItem or appropriate InsertItem function instead.

ItemGroup

Applies to

ToolBarItem objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Determines whether a toolbar item is part of a toolbar item group. Values are:

Value	Description
0	Specifies that the toolbar button does not belong to a toolbar item group
1	Specifies that the toolbar button belongs to a toolbar item group

A toolbar item group must have a minimum of two toolbar buttons. The buttons in a group act in a dependent manner. Only one button in a group can be depressed at the same time. You can identify different toolbar button groups only by their position in the toolbar. The limit of a toolbar group is defined by a succeeding toolbar item that is not part of a group. You cannot include two groups of items in the same toolbar without separating them by a toolbar item that is not part of either group.

Usage

In a painter

❖ To add for a toolbar button to a group:

- 1 Select the toolbar item on the Items page of the Properties view for a Toolbar control.
 - 2 Select StyleCheckGroup! from the ButtonStyle drop-down list.
- Selecting this style sets the ItemGroup value to 1. Selecting any other style sets the ItemGroup value to 0.

In scripts

The ItemGroup property has an integer datatype. You can use integer values greater than 1, but this has the same effect as setting the ItemGroup property to 1. The following example adds two toolbar buttons to a toolbar item group:

```
myItem.ItemPictureIndex = 1
myItem.ItemGroup = 1
li_rtn = tlbr_mytoolbar.AddItem(myItem)
myItem.ItemPictureIndex = 2
myItem.ItemGroup = 1
li_rtn = tlbr_mytoolbar.AddItem(myItem)
```

ItemPictureIndex

Applies to

ToolbarItem objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Determines the picture that displays on a toolbar button. The value you select for the picture index must correspond to an entry in the toolbar picture array.

Usage**In a painter****❖ To select a picture for a toolbar item:**

- 1 Select the Pictures tab page of the Properties view for a Toolbar control.
- 2 To select pictures to add to the toolbar picture array, do one of the following :
 - In the rows provided in the PictureName field, type the complete path and name of the files containing the desired pictures.
 - Use the Browse button.
 - Select pictures from the stock icon drop-down list.
- 3 Select the Items tab page of the Properties view for the Toolbar control.
- 4 Specify a picture index number for each item on the Items tab page that you want to associate with a picture.

The picture index number must correspond to the row number of a picture you selected in the PictureNames list.

In scripts

The ItemPictureIndex property has an integer datatype. The following example sets the picture associated with a toolbar button to the second picture in the toolbar picture array:

```
tbi_myToolbarItem.ItemPictureIndex = 2
```

ItemPictureIndex[]

Applies to

PictureListBox, DropDownPictureListBox, ListView

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

PowerBuilder only

PictureListBox and DropDownPictureListBox are not supported in PocketBuilder.

Description The ItemPictureIndex property array identifies the pictures associated with items in the control. This array is not updated after initialization.

Usage

In a painter

❖ **To associate pictures with list items:**

- 1 Display the Pictures tab page in the control’s Properties view and add the pictures to be used in the control to the PictureName list.

For ListView controls, add pictures to the PictureName lists on the LargePicture, SmallPicture, and State tab pages.
- 2 Display the Items tab page in the Properties view and add text to the Item list.
- 3 In the ItemPictureIndex list, add the index number for a picture (from the PictureName lists) on the appropriate lines for the items with which you want to associate pictures.

In scripts

You add pictures to controls with the AddPicture function and add items to these controls with the AddItem or InsertItem functions. You use picture indexes in the AddItem and InsertItem functions to associate pictures with the items.

See “Using Lists in a Window” in the *Resource Guide* for more information about using scripts for these controls.

ItemState

Applies to	ToolbarItem objects	
	PocketBuilder on Pocket PC	✓
	PocketBuilder on Smartphone	✓
	PowerBuilder	✗

Description Determines the state of a toolbar item. Values are:

Value	Description
0	Disables a toolbar button
1	Specifies a StyleCheck! or StyleCheckGroup! toolbar button in the depressed state
2	Specifies a StyleButton! toolbar button in a transitional depressed state

Value	Description
4	Enables a toolbar button for selection
32	Specifies that the next item in the toolbar will wrap to a separate line if it is not part of the same toolbar item group

Values are additive. For example, suppose you want to set a toolbar button with the checked state (1) and enable it (4), with the next set of buttons wrapped to a different line (32). You would enter 37 for the value of the ItemState property.

When you add a toolbar item in the Properties view for a Toolbar control, the default value for the ItemState property is 4. If you clear the ItemEnabled check box, the ItemState property changes to 0. In the PocketBuilder UI, there is currently no other way to change the ItemState value directly or to assign it a value other than 0 or 4.

The default value for the ItemState property of a toolbar item that you add in script is 0. You can change the property directly in script by assigning a valid ItemState value or by calling SetItemState on the Toolbar control.

Usage

In a painter

❖ To select a style for a toolbar item:

- 1 Select the toolbar item on the Items page of the Properties view for a Toolbar control.
- 2 Select or clear the Enabled check box.

Selecting the Enabled check box sets the ItemState value to 4.

In scripts

The ItemState property has an integer datatype. The following example enables the toolbar button and sets the toolbar item with a checked state:

```
tbi_myToolbarItem.ItemState = 5
```

ItemStyle

Applies to

ToolbarItem objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

- Description
- Determines the style of a toolbar item. Values are:
- **StyleButton!** Specifies a standard toolbar button that automatically returns to the up position after a user taps it.
 - **StyleCheck!** Specifies a toolbar button that remains depressed when a user taps it. The user must tap the button a second time to raise it.
 - **StyleCheckGroup!** Specifies a toolbar button that is grouped with other buttons having this same style. A button of this style remains depressed after a user taps it, but returns to the up position when the user taps it again or taps another toolbar button of the same style.
 - **StyleSeparator!** Specifies a separator bar.

- Usage
- In a painter**
- ❖ **To select a style for a toolbar item:**

 - 1 Select the toolbar item on the Items page of the Properties view for a Toolbar control.
 - 2 Select the style for the toolbar item in the Style drop-down list.

In scripts

The Style property takes an enumerated value of the type `ToolbarItemStyle`. The following example changes a toolbar item style, adding it to a group of buttons. The button remains depressed after a user taps it, but returns to the up position when the user taps the button again or taps another button of the same style:

```
tbi_myToolbarItem.Style = StyleCheckGroup!
```

Label

Applies to

grAxis objects in Graph controls, ListViewItem objects, TreeViewItem objects	
PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

Within graphs The Label property specifies the label of an axis of the graph.

Usage

Within ListView and TreeView controls The Label property specifies the label associated with a ListViewItem or TreeViewItem object. You cannot manipulate items in TreeView controls in a painter. You must write scripts to add items to a TreeView.

In a painter❖ **To specify an Axis label in a graph control:**

- 1 Display the Axis tab page of the graph control's Properties view.
- 2 Select the desired axis from the Axis drop-down list.
- 3 Enter the label text in the Label text field.

❖ **To specify labels for items in a ListView control:**

- 1 Display the Items tab page of the ListView control's Properties view.
- 2 For each item, enter label text in the appropriate Text field.

In scripts

The Label property takes a string value. The following example sets text for the label on the Values axis of graph gr_1.

```
gr_1.Values.Label = 'Lawsuits per 1000'
```

To add or insert an item with a label into a ListView control, use the AddItem or InsertItem functions. For example, this line adds an item to ListView control lv_1, specifying the label and picture index for the item:

```
lv_1.AddItem ( "Oranges", 1)
```

To change the label, get the item from the ListView and set the item's Label property:

```
ListViewItem lvi
lv_1.GetItem(4, lvi)
lvi.Label = "Apples"
lv_1.SetItem(4, lvi)
```

To add or insert items in a TreeView control, use the InsertItem, InsertItemFirst, InsertItemLast, or InsertItemSort functions.

For more information, see “Using TreeView Controls” and “Using ListView Controls” in the *Resource Guide*.

LabelWrap

Applies to

ListView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When the LabelWrap property is enabled, long ListView item labels wrap in small icon view. If LabelWrap is not enabled, labels are displayed on a single line. LabelWrap does not apply to report and list views.

Usage

In a painter

- ❖ **To enable label wrap for a ListView control:**
 - Select the Label Wrap check box on the General page of the control's Properties view.

In scripts

The LabelWrap property takes a boolean value. The following line enables word wrapping of labels in a ListView.

```
lv_1.LabelWrap = TRUE
```

LargePictureName[]

Applies to

ListView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

PocketBuilder stores ListView images in several indexed arrays of images. You can associate an image with a specific ListView item when you create a ListView in the painter or use the AddItem and InsertItem functions at execution time.

You identify a specific image by its index number. Because the same index number refers to both the large picture and the small picture for the item (depending on which view is selected), you will want to make sure the images for each position in the array are compatible. The type of image used is determined by the value of the View property of the control.

Usage

In a painter

❖ To specify images for the Large Icon view

- 1 Select the Large Picture tab page from the ListView control's Properties view.
- 2 Do one of the following:
 - In the rows provided in the Picture Name field, type the complete path and name of the files containing the desired pictures.
 - Use the Browse button.
 - Select one or more pictures from the Stock Pictures list.

The order of the picture names specified here should match the picture name order used for the Small Icon view.

- 3 Use the row numbers from this Picture Name list to specify the Picture Index for each List View Item on the Items tab page.

In scripts

The LargePictureName property takes a string value. You cannot use the LargePictureName property to update the image list during execution. Use the AddLargePicture function to add large pictures to a ListView control. For example:

```
lv_1.AddLargePicture("c:\ArtGal\bmps\celtic.bmp")
```

When you add a large picture to a ListView control, it is given the next available picture index in the ListView.

For more information about scripting ListView controls, see "Using ListView controls" in the *Resource Guide*.

LargePictureWidth

Applies to

ListView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The LargePictureWidth property specifies the display width of all the pictures in the Large Icon view of the ListView control. The size is specified in pixels.

If you choose the value (Default) in the painter, or set the value to 0, PocketBuilder uses the width of the first picture in the array as the width for all the pictures. The other choices in the painter, 16 and 32, are standard pixel widths for icons.

Usage

In a painter

❖ **To set the large picture width:**

- Select a value from the Width drop-down list on the Large Picture tab page of the control's Properties view.

In scripts

The LargePictureWidth property takes an integer value. This value can only be set before the first call to the AddLargePicture function or after calling DeleteLargePictures. If this value is set to 0, then the size of the first picture is used to set the size of large pictures.

The following line sets the width for large pictures in a ListView to 32 pixels.

```
lv_1.LargePictureWidth = 32
```

For more information about scripting ListView controls, see “Using ListView controls” in the *Resource Guide*.

LargePictureHeight

Applies to

ListView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The LargePictureHeight property specifies the display height of all the pictures in the Large Icon view of the ListView control. The size is specified in pixels.

If you choose the value (Default) in the painter, or set the value to 0, PocketBuilder uses the height of the first picture in the array as the height for all the pictures. The other choices in the painter, 16 and 32, are standard pixel heights for icons.

Usage

In a painter

❖ To set the large picture height:

- Select a value from the Height drop-down list on the Large Picture tab page of the control's Properties view.

In scripts

The LargePictureHeight property takes an integer value. This value can only be set before the first call to the AddLargePicture function or after calling DeleteLargePictures. If this value is set to 0, then the size of the first picture is used to set the size of large pictures.

The following line sets the height for large pictures in a ListView to 32 pixels.

```
lv_1.LargePictureHeight = 32
```

For more information about scripting ListView controls, see "Using ListView controls" in the *Resource Guide*.

LargePictureMaskColor

Applies to

ListView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The mask color is the color in the picture that is transparent when the picture is displayed.

Select the color to mask newly added user-defined bitmaps. In scripts, you can change the mask color before adding each picture. Each image uses the mask color that was in effect when it was added.

Usage

In a painter

❖ To specify a picture mask color:

- Select a color from the Picture Mask Color drop-down list on the Large Picture tab page of the control's Properties view.

To add your own colors to the color drop-down list, select Design>Custom Colors before displaying the Properties view.

In scripts

The LargePictureMaskColor property takes a long (-2 to 16,777,215) that specifies the numerical value of the background color. This property is used when each bitmap is added and, therefore, can be changed between AddLargePicture calls.

The LargePictureMaskColor value is a combination of values for the red, green, and blue components of the color. If you do not know the long value for a particular color, choose Design>Custom Colors to determine the red, green, and blue values and then call the RGB function to specify the color in a script.

The following example sets yellow as the mask color for user-defined bitmaps in a ListView.

```
lv_1.LargePictureMaskColor = RGB(255, 255, 0)
```

Latitude

Applies to

GPSTFix objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The Latitude property holds information about the latitude of a position fix.

Usage

In scripts

The Latitude property takes a value of the GPSCoordinate structure, which holds the latitude of a position fix in degrees and minutes in the Northern or Southern Hemisphere. The Minute property is a real value. The following example takes the Minute property, separates it into whole minutes and a partial minute, and converts the partial minute into a number of seconds:

```

SerialGps myGPS
GPSTFix myFix
GPSCoordinate myCoord
Integer fixMinutes, rc
Real fixSeconds

MyGPS = CREATE SerialGPS
rc = myGPS.Open()
...
rc = MyGPS.GetFix(myFix)
myCoord = myFix.Latitude
fixMinutes = Integer(myCoord.Minute)
fixSeconds = (myCoord.Minute - fixMinutes) * 60
mle_1.text = "Latitude: " &
    + String(myCoord.degree) + " degrees " &
    + String(fixMinutes) + " minutes " &
    + String(fixSeconds) + " seconds " &
    + String(myCoord.Hemisphere)

```

LeftMargin

Applies to

RichTextEdit controls

PocketBuilder	✗
PowerBuilder	✓

Description

The LeftMargin property specifies the size in inches of the left margin on the printed page.

LeftText

Applies to

CheckBox, RadioButton controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

If the LeftText property is enabled, the text for a CheckBox or RadioButton appears to the left of the button. If LeftText is not enabled, the text appears to the right of the control. You can also specify left or right alignment with the left and right StyleBar buttons.

Usage

When the LeftText property is enabled and you align multiple CheckBoxes or RadioButtons to the left, PocketBuilder may align the text but not the boxes or buttons. This is because PocketBuilder aligns the complete control.

In a painter

- ❖ **To place text to the left:**
 - Select the Left Text check box on the General page of the control's Properties view.

In scripts

The LeftText property takes a boolean value. The following line puts the text for a CheckBox to the left of the box.

```
cbx_1.LeftText = TRUE
```

Legend

Applies to

Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The value of the Legend property specifies the placement of the graph's legend, or that there should be no legend.

Usage

In a painter

❖ **To set the location of the legend:**

- Select the desired location from the Legend drop-down list on the General page of the control's Properties view.

In scripts

The Legend property takes a value of the `grLegendType` enumerated datatype. The following example sets the location of the legend to below the graph:

```
gr_1.Legend = AtBottom!
```

Limit

Applies to

DropDownListBox, MultiLineEdit, SingleLineEdit controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The Limit property controls the number of characters the user can enter in the control.

Usage

In a painter

❖ **To set the maximum number of characters allowed:**

- Type the number of characters that the user can enter in the control in the Limit field of the General page of the control's Properties view.

0 indicates an unlimited number of characters.

In scripts

The Limit property takes an integer value.

The following example sets 256 as the maximum number of characters for the MultiLineEdit mle_1.

```
mle_1.Limit = 256
```

LineCallParamFlags

Applies to

PhoneCall objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

A value that specifies a set of boolean parameters used to set options when making a phone call using the MakeCall function and for data transmission. Values are:

- **1** Secure – set up the call as secure
- **2** Idle – do not break into a call in progress
- **4** BlockID – block the identity of the caller
- **8** OrigOffHook – take the originator’s phone off the hook automatically
- **16** DestOffHook – take the called party’s phone off the hook automatically
- **32** NoHoldConference – do not put other parties on hold when joining a conference call
- **64** PredictiveDial – enable enhanced features when placing a call on an address with predictive dialing capability
- **128** OneStepTransfer – enable one-step transfer when transferring a call

Usage

In a painter

Specify a value for the LineCallParamFlags property in the LineCallParamFlags box on the General page of the PhoneCall object’s Properties view. Values are additive. For example, to specify that both the originating and destination phones should be taken off the hook, specify 24 (8 + 16).

In scripts

The `LineCallParamFlags` property is an unsigned long value indicating the set of parameters to apply when making a phone call. The following example specifies that the call should be secure and the caller ID should be blocked (1 + 4):

```
pcall_1.LineCallParamFlags = 5
```

LineColor

Applies to

Line, Oval, Rectangle, RoundedRectangle controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The line color is the color for the border and the lines in the control's fill pattern.

Usage

In a painter❖ **To change the line color:**

- Select a color from the Line Color drop-down list on the General page of the control's property sheet, or select a color from the Background Color option on the Style Bar.

Using the StyleBar lets you change the line color for several selected objects at the same time. To add your own colors to the color drop-down list, select Design>Custom Colors before opening the Properties view.

In scripts

The `LineColor` property takes a long value. If you do not know the long value for the color, choose Design>Custom Colors to determine the red, green, and blue values and then call the `RGB` function to specify the color in a script.

The following statement specifies red as the line color for a Rectangle.

```
r_1.LineColor = RGB(255,0,0)
```

LineMediaMode

Applies to

PhoneCall objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

A value that specifies the media type of a phone call. Values are:

- **2** Unknown – a media stream exists but its mode is currently unknown
- **4** InteractiveVoice – an interactive voice call with human parties at both ends
- **8** AutomatedVoice – an automated call handled by an answering machine or similar application
- **16** DataModem – a data modem session
- **32** G3Fax – a group 3 fax is being sent or received
- **64** TDD – a Telephony Device for the Deaf session
- **128** G4Fax – a group 4 fax is being sent or received
- **256** DigitalData – a digital data stream of unspecified format
- **512** Teletex – a teletex session
- **1024** Videotex – a videotex session
- **2048** Telex – a telex session
- **4096** Mixed – a mixed (ISDN telematic) session
- **8192** ADSI – an Analog Display Services Interface session
- **16384** VoiceView – VoiceView media type

Usage

In a painter

Specify a value for the LineMediaMode property in the LineMediaMode box on the General page of the PhoneCall object’s Properties view.

In scripts

The LineMediaMode property is an unsigned long value indicating the type of information stream exchanged over a call. The following line writes the LineMediaMode value to a single line edit box:

```
sle_media.text = string(pcall_1.LineMediaMode)
```

LinesAtRoot

Applies to

TreeView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When LinesAtRoot is enabled, PocketBuilder will connect all root items in a TreeView with lines.

Usage

In a painter

❖ To specify that root items in a TreeView are connected:

- Select the Lines At Root check box on the General page of the control's property page.

In scripts

The LinesAtRoot property takes a boolean value.

The following line specifies that all root items in a TreeView are connected.

```
tv_1.LinesAtRoot = TRUE
```

LinesPerPage

Applies to

User objects and windows

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

The LinesPerPage property determines the number of lines on a page for scrolling purposes. PocketBuilder multiplies Units Per Line by Lines Per Page to determine the number of PowerBuilder units to scroll the window vertically when the user clicks in the scroll bar.

For information on calculating LinesPerPage and UnitsPerLine, see “Scrolling in windows and user objects” on page 569.

To control the horizontal scroll bar in a window or user object, use the UnitsPerColumn and ColumnsPerPage properties.

Usage

In a painter

- ❖ **To set the LinesPerPage property:**
 - Enter the desired number (between 1 and 100) in the Lines Per Page option on the Scroll tab page of the window’s Properties view.

In scripts

The LinesPerPage property takes an integer value between 1 and 100. The following line sets LinesPerPage for a window to 20.

```
This.LinesPerPage = 20
```

LineStyle

Applies to

Line, Oval, Rectangle, RoundedRectangle controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The LineStyle property specifies the appearance of a line object or the border around other drawing objects.

Usage

In a painter

- ❖ **To set the line style:**
 - Select a line style from the Line Style drop-down list on the General page of the control’s Properties view.

In scripts

The LineStyle property takes a value of the LineStyle enumerated datatype. The following example sets a dashed line for a Rectangle.

```
r_1.LineStyle = Dash!
```

LinkUpdateOptions

Applies to

OLE controls

PocketBuilder	✗
PowerBuilder	✓

Description

The LinkUpdateOptions property specifies how a linked object in an OLE control is updated. There are two options: automatic and manual. If automatic is chosen, the link is updated when the object is opened and whenever the object changes in the server application. If manual is chosen, the link is not updated.

LiveScroll

Applies to

DataWindow controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When the LiveScroll property is enabled, the rows in the DataWindow control will scroll while the user is moving the scroll box. If this property is not enabled, the rows do not scroll until the user releases the scroll box.

Usage

In a painter

❖ To enable live scrolling in a DataWindow control:

- Select the Live Scrolling check box on the General page of the control's Properties view.

In scripts

The LiveScroll property takes a boolean value.

The following example allows scrolling while the user moves the scroll box in the DataWindow control:

```
This.LiveScroll = TRUE
```

Longitude

Applies to

GPSTFix objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The Longitude property holds information about the longitude of a position fix.

Usage

In scripts

The Longitude property takes a value of the GPSCoordinate structure, which holds the longitude of a position fix in degrees and minutes in the Eastern or Western hemisphere. The Minute property is a real value. The following example takes the Minute property, separates it into whole minutes and a partial minute, and converts the partial minute into a number of seconds:

```
SerialGps myGPS
GPSTFix myFix
GPSCoordinate myCoord
Integer fixMinutes, rc
Real fixSeconds

MyGPS = CREATE SerialGPS
rc = myGPS.Open()
...
rc = MyGPS.GetFix(myFix)
myCoord = myFix.Longitude
fixMinutes = Integer(myCoord.Minute)
fixSeconds = (myCoord.Minute - fixMinutes) * 60
mle_1.text = "Longitude: " &
    + String(myCoord.degree) + " degrees " &
    + String(fixMinutes) + " minutes " &
    + String(fixSeconds) + " seconds " &
    + String(myCoord.Hemisphere)
```

MagneticVariation

Applies to

GPSHeading objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The MagneticVariation property specifies the magnetic variation in degrees. Magnetic variation is the difference between the magnetic heading and the true heading. If the GPS device does not support measurement of the magnetic variation, this value is 0.

Usage

In scripts

The MagneticVariation property takes a real value in the range 0.0 to 180.0 degrees:

```
real mVariation
GPSHeading myGPSHeading

mVariation = myGPSHeading.MagneticVariation
```

MagneticVariationDirection

Applies to

GPSHeading objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The MagneticVariationDirection property specifies the direction, east or west, of the magnetic variation. If the GPS device does not support measurement of the magnetic variation, this value is empty. Values are E for East or W for West.

Usage

In scripts

The MagneticVariationDirection property takes a Char value. This example writes the true heading and magnetic variation to a multiline edit box:

```
real TrueHeading
real MV
char MVD
```

```
GPSHeading myGPSHeading

TrueHeading = myGPSHeading.Heading
MV = myGPSHeading.MagneticVariation
MVD = myGPSHeading.MagneticVariationDirection
mle_1 = "True heading: " + String(TrueHeading) + "~r~n"
mle_1 += "Variation: " + String(MV) + MVD
```

MajorGridLine

Applies to

grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The major grid for an axis extends from the tick marks on the axis across the data area of the graph. The minor grid lines fall between the tick marks and display when the number of divisions is 2 or greater.

The MajorGridLine property specifies the line style for the major grid.

Usage

In a painter

- ❖ **To set the line style for a major grid line:**
 - 1 Display the Axis tab page of the graph control’s Properties view and select the desired axis from the Axis drop-down list.
 - 2 Select the desired line style from the MajorGridLine drop-down list in the Major Divisions group

In scripts

The MajorGridLine property takes a value of the LineStyle enumerated datatype.

This example specifies a dashed line for the major grid line on the Value axis of Graph gr_1.

```
gr_1.Value.MajorGridLine = Dash!
```

MajorDivisions

Applies to

grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The MajorDivisions property specifies the number of divisions or ticks on the axis, not counting the origin point.

The default value of 0 means the graph uses a MajorDivision value optimized for the data and will suppress MinorDivision ticks.

Usage

In a painter

❖ **To specify the number of major divisions on an axis:**

- 1 Display the Axis tab page from the graph's Properties view and select the desired axis from the Axis drop-down list.
- 2 Use the spin control in the MajorDivisions field of the Major Divisions group to specify the desired number of divisions.

In scripts

The MajorDivisions property takes an integer specifying the number of major divisions on an axis.

The following example sets 10 ticks on the major grid of the Values axis of a graph.

```
gr_1.Values.MajorDivisions = 10
```

MajorTic

Applies to

grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description The MajorTic property specifies how ticks overlap the axis for the major grid. Ticks can be placed on the inside of the axis line, on the outside, or straddling it; or there can be no ticks visible.

Usage **In a painter**

❖ **To specify the type of major tick marks:**

- 1 Display the Axis tab page of the control’s Properties view and select the desired axis from the Axis drop-down list.
- 2 Select the desired type of tick mark from the MajorTick drop-down list box in the Major Division group.

In scripts

The MajorTic property takes a value of the grTicType enumerated datatype. The following line sets ticks on the major grid to straddle the grid.

```
gr_1.Values.MajorTic = Straddle!
```

Map3DColors

Applies to Picture, PictureHyperlink, and PictureButton controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description Maps the silver and other gray colors in the bitmap associated with the control to the button highlight, button face, or button shadow colors set in the Windows control panel. When this property is FALSE (the default), the control uses the standard PocketBuilder button colors defined in the bitmap.

Use this feature if you want to place a control containing a picture on a window and have the picture blend in with the background color of the window when the window's background is using Button Face for a 3D effect. The control's picture then takes on the 3D colors the user has selected on the Appearance page of the Display Properties dialog box in the Windows control panel.

The window's background must be set to Button Face. To make the image blend in with the window, give it a background color in the range between RGB(160,160,160) and RGB(223,223,223), such as silver. Lighter shades of gray map to the button highlight color and darker shades to the button shadow color.

This option can affect other colors used in the bitmap. It does not affect the control's border settings, and it has no effect if there is no image associated with the control.

Usage

In a painter

❖ **To set 3DColor mapping:**

- Select the 3D Color check box on the General page of the control's Properties view.

In scripts

The Map3DColors property takes a boolean value. The following example sets 3D color mapping for a PictureBox.

```
pb_1.Map3DColors = TRUE
```

Mask

Applies to

EditMask controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The Mask property controls the characters the user can enter in the control and also the formatting of the characters. You must use special characters to define the mask, depending on the mask type defined with the MaskDataType property.

PocketBuilder supports four kinds of display formats:

- Numbers
- Strings
- Dates
- Times

Characters that do not have special meaning for the format appear as is in the EditMask control.

To display additional characters as part of the mask for a decimal value, you must precede each character with a backslash. For example, to display a decimal number with two digits of precision preceded by four asterisks, you must type a backslash before each asterisk: `\*\*\*0.00`

For more information about using each kind of display format, see Chapter 4, “About Display Formats and Scrolling.” For more information about defining display formats, see the *User’s Guide*.

Usage

In a painter

❖ To specify an edit mask:

- 1 Display the Mask tab page of the control’s Properties view.
- 2 Select the mask datatype from the MaskDataType drop-down list.
- 3 Type the mask characters in the Mask field, or click the right arrow at the end of the Mask field and select one or more of the mask character examples displayed in the pop-up menu.

The pop-up menu examples change based on the mask datatype you selected in the MaskDataType list.

In scripts

The Mask property takes a string value and can be used to obtain the value of a mask. It cannot be used to set the value.

The following example uses the SetMask function to set the datatype and string format for a mask, and then uses the Mask property to obtain the value of the string format:

```
string ls_mask

em_1.SetMask(StringMask!, '@@@,@@')
ls_mask = em_1.Mask
```

MaskDataType

Applies to

EditMask controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

This property specifies the datatype of the control. The special characters used to define the mask differ depending on the datatype of the mask control.

Usage

In a painter

❖ To select the mask datatype:

- Select the desired type from the Type drop-down list box in the Options group on the Mask tab.

The examples of special characters displayed in the Masks field change to show the characters relevant to the selected mask datatype.

In scripts

The MaskDataType property takes a value of the MaskDataType enumerated datatype and can be used to obtain the datatype of a mask. It cannot be used to set the datatype.

The following example uses the SetMask function to set the datatype and date format for a mask, and then uses the Mask property to obtain the value of the date format:

```
MaskDataType l_mdt

em_1.SetMask(DateMask!, 'mm/dd/yy')
l_mdt = em_1.MaskDataType
```

MaxBox

Applies to DataWindow controls, Windows

PocketBuilder	✕
PowerBuilder	✓

Description The MaxBox property specifies whether a Maximize box is displayed on the control’s title bar.

MaximumValue

Applies to grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description The MaximumValue property specifies the maximum value for an axis when the axis datatype is numeric. This property is not used if the Autoscale property is enabled.

Usage **In a painter**

❖ To set the maximum value of an axis with a numeric datatype:

- 1 Display the Axis tab page of the graph’s Properties view and select the desired axis from the Axis drop-down list.
- 2 Make sure that the Autoscale check box is not checked.
- 3 Select adtDouble! from the DataType drop-down list.
- 4 Specify the desired maximum numeric value in the MaximumValue field.
This value should be larger than the maximum data value being graphed.

In scripts

The MaximumValue property takes a double value.

The following line sets a maximum value for an Axis with a datatype of double.

```
gr_1.Values.DataType = AdtDouble!  
gr_1.Values.MaximumValue = 500000.00
```

MaxPosition

Applies to

HScrollBar, VScrollBar controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

The MaxPosition property specifies what the value of the Position property is when the scroll box is at the bottom of the VScrollBar control or the right edge of the HScrollBar scroll bar.

Usage

In a painter

❖ To specify the maximum position of the scroll box:

- Enter an integer value into the Max Position field of the General tab of the control's Properties view.

In scripts

The MaxPosition property takes an integer value.

The following example specifies that the value of the Position property is 120 when the scroll box is in the maximum position.

```
vsb_1.MaxPosition = 120
```

MaxValDateTime

Applies to

grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The MaxValDateTime property specifies the maximum value for an axis when the axis datatype is date or time. This property is not used if the Autoscale property is enabled.

Usage

In a painter

- ❖ **To set the maximum value of an axis with a date or time datatype:**
 - 1 Display the Axis tab page of the graph's Properties view and select the desired axis from the Axis drop-down list.
 - 2 Make sure that the Autoscale check box is not checked.
 - 3 Select adtDate!, adtTime!, or adtDateTime! from the DataType drop-down list.
 - 4 Specify the desired maximum date or time value in the MaximumValue field.

This value should be larger than the maximum data value being graphed.

In scripts

The MaxValDateTime property takes a value of the DateTime datatype.
The following example sets the MaxValDateTime property for an Axis with a datatype of date.

```
gr_1.Values.DataType = AdtDate!  
gr_1.Values.MaxValDateTime = 12/31/1999
```

MenuBar

Applies to

Windows

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The MenuBar property makes room at the bottom of the current window to allow a menu to be inserted.

Usage

In a painter

- ❖ **To make room for a menu at the bottom of a window:**
 - Select the MenuBar check box on the General page of the window's Properties view.

When you set a value for the MenuName property of a window, the MenuBar property is automatically selected and disabled.

In scripts

This property cannot be set in a script.

MenuName

Applies to

Windows

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The MenuName property specifies the menu object that is the menu for the window.

Usage

In a painter**❖ To specify a menu:**

- Enter a menu name in the Menu Name field on the General page of the window's Properties view, or use the Browse button to choose a menu object from the current or another PKL.

In scripts

The MenuName property takes a string containing the name of a menu object. PocketBuilder uses it internally to identify the menu. Do not change this property in a script. Instead, use the ChangeMenu or PopMenu functions to display a menu.

MessageClass

Applies to

SMSPProviderSpecificData objects

PocketBuilder	✗
PowerBuilder	✗

Property not implemented
The SMSProviderSpecificData object and its properties are not implemented in the current release of PocketBuilder. They are reserved for future use.

MessageOptions

Applies to SMSProviderSpecificData objects

PocketBuilder	✗
PowerBuilder	✗

Property not implemented
The SMSProviderSpecificData object and its properties are not implemented in the current release of PocketBuilder. They are reserved for future use.

MinBox

Applies to DataWindow controls, Windows

PocketBuilder	✗
PowerBuilder	✓

Description The MinBox property specifies whether a Minimize box is displayed on the control's title bar.

MinimumValue

Applies to

grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The MinimumValue property specifies the minimum value for an axis when the axis datatype is numeric. This property is not used if the Autoscale property is enabled.

Usage

In a painter

❖ **To set the minimum value of an axis with a numeric datatype:**

- 1 Display the Axis tab page of the graph's Properties view and select the desired axis from the Axis drop-down list.
- 2 Make sure that the Autoscale check box is not checked.
- 3 Select adtDouble! from the DataType drop-down list.
- 4 Specify the desired minimum numeric value in the MinimumValue field.
This value should be smaller than the minimum data value being graphed.

In scripts

The MinimumValue property takes a double value.

The following line sets a minimum value for an Axis with a datatype of double.

```
gr_1.Values.DataType = AdtDouble!  
gr_1.Values.MinimumValue = 0.00
```

MinMax

Applies to

EditMask controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The value of the MinMax property specifies the minimum and maximum values allowed when the EditMask functions as a spin control.

You can specify minimum and maximum values only for date and numeric datatypes. For dates, enter a full date (for example, 1/1/2003), although the minimum and maximum values only affect the year. The user can scroll freely through the days and months.

- Usage**
- In a painter**
- ❖ **To set the minimum and maximum spin values:**
 - 1 Select the Spin Control check box on the Mask tab page of the control's Properties view.
 - 2 Enter minimum and maximum values in the Min and Max fields in the Spin Range group.

In scripts

The MinMax property takes a string value. The values are separated with two tildes (~~)

The following example sets the minimum and maximum spin for an Edit Mask em_1.

```
em_1.MinMax = ("100 ~~ 10000")
```

MinorGridLine

Applies to	grAxis objects in Graph controls	
	PocketBuilder on Pocket PC	✓
	PocketBuilder on Smartphone	✓
	PowerBuilder	✓

Description

The minor grid for an axis extends from the tick marks on the axis across the data area of the graph. The minor grid lines fall between the tick marks and display when the number of divisions is two or greater.

The MinorGridLine property specifies the line style for the minor grid.

Usage**In a painter**❖ **To set the line style for a minor grid line:**

- 1 Display the Axis tab page of the graph control's Properties view and select the desired axis from the Axis drop-down list.
- 2 Select the desired line style from the MinorGridLine drop-down list in the Minor Divisions group

In scripts

The MinorGridLine property takes a value of the LineStyle enumerated datatype.

This example specifies a dotted line for the minor grid line on the Value axis of Graph gr_1.

```
gr_1.Value.MinorGridLine = Dot!
```

MinorDivisions

Applies to

grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The MinorDivisions property specifies the number of spaces between major ticks. To see minor ticks, specify a value of two or greater because the last minor tick is overlaid by the next major tick.

The default value of 0 in the MajorDivisions field means the graph uses a MajorDivision value optimized for the data and will suppress MinorDivision ticks.

Usage**In a painter**❖ **To specify the number of minor divisions on an axis:**

- 1 Display the Axis tab page from the graph's Properties view and select the desired axis from the Axis drop-down list.
- 2 Use the spin control in the MinorDivisions field of the Minor Divisions group to specify the desired number of divisions.

In scripts

The MinorDivisions property takes an integer specifying the number of minor divisions on an axis.

The following example sets 10 ticks on the minor grid of the Values axis of a graph.

```
gr_1.Values.MinorDivisions = 10
```

MinorTic

Applies to

grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The MinorTic property specifies how ticks overlap the axis for the minor grid. Ticks can be placed on the inside of the axis line, on the outside, or straddling it; or there can be no ticks visible.

Usage

In a painter

❖ **To specify the type of minor tick marks:**

- 1 Display the Axis tab page of the control’s Properties view and select the desired axis from the Axis drop-down list.
- 2 Select the desired type of tick mark from the MinorTicks drop-down list box in the Minor Division group.

In scripts

The MinorTic property takes a value of the grTicType enumerated datatype.

The following line sets ticks on the minor grid to outside the grid.

```
gr_1.Values.MinorTic = Outside!
```

MinPosition

Applies to

HScrollBar, VScrollBar controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

The MinPosition property specifies what the value of the Position property is when the scroll box is at the top of the VScrollBar control or the left edge of the HScrollBar scroll bar.

Usage

In a painter

❖ **To specify the minimum position of the scroll box:**

- Enter an integer value into the Min Position field of the General tab of the control's Properties view.

In scripts

The MinPosition property takes an integer value.

The following example specifies that the value of the Position property is 0 when the scroll box is in the minimum position.

```
vsb_1.MinPosition = 0
```

Minute

Applies to

GPSCoordinate objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The Minute property specifies the minute portion of the position of a fix in degrees.

Usage **In scripts**

The Minute property takes a real value in the range 0 to 59.9999:

```
real minutes
GPSCoordinate myCoord
lr_minutes = myCoord.Minute
```

See also Latitude
Longitude

MinValDateTime

Applies to grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description The MinValDateTime property specifies the minimum value for an axis when the axis datatype is date or time. This property is not used if the Autoscale property is enabled.

Usage **In a painter**

- ❖ **To set the minimum value of an axis with a date or time datatype:**
 - 1 Display the Axis tab page of the graph's Properties view and select the desired axis from the Axis list.
 - 2 Make sure that the Autoscale check box is not checked.
 - 3 Select adtDate!, adtTime!, or adtDateTime! from the DataType drop-down list.
 - 4 Specify the desired minimum date or time value in the MinimumValue field.

This value should be smaller than the minimum data value being graphed.

In scripts

The MinValDateTime property takes a value of the DateTime datatype.

The following example sets the MinValDateTime property for an Axis with a datatype of date.

```
gr_1.Values.DataType = AdtDate!
gr_1.Values.MinValDateTime = 01/31/1900
```

Mode

Applies to

SMSProtocol objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Specifies whether an SMSSession object is opened in send or receive mode. Values are:

- **SMSModeReceive!** Receive mode (not supported in this release)
- **SMSModeSend!** Send mode

Usage

In scripts

The Mode property takes a value of the SMSMsgModes enumerated variable. This example sets the mode of an SMSProtocol object to send:

```
SMSProtocol mysmsprotocol
mysmsprotocol.Mode = SMSModeSend!
```

MultiSelect

Applies to

ListBox, PictureListBox controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description	The MultiSelect property specifies whether users can select multiple items in the list box at one time. When it is enabled, users can select multiple items by clicking them, and you can call SetState in script to set the state of each list box item. When MultiSelect is not enabled, users cannot select multiple items at once, and you must use SelectItem instead of SetState to select an item in script. If MultiSelect and ExtendedSelect are both enabled, then the behavior of ExtendedSelect takes precedence. For ExtendedSelect, the user must press Shift or Ctrl when clicking additional items.
Usage	In a painter ❖ To enable multiple selections from the list: • Select the Multi Select check box on the General page of the control's Properties view. In scripts The MultiSelect property takes a boolean value. The following example allows multiple selections in the ListBox lb_1. <pre>lb_1.MultiSelect = TRUE</pre>
See also	SetState

Multiline

Applies to	Tab controls
------------	--------------

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description	When Multiline is enabled, the tabs can appear in more than one row if there is not room for all the tabs in a single row. When Multiline is not enabled, a dual arrow control will appear to allow the user to scroll to tabs that do not fit.
Usage	In a painter ❖ To enable multiline display of tabs: • Select the Multiline check box on the General page of the control's Properties view.

In scripts

The Multiline property takes a boolean value. The following line allows tabs to be arranged in multiple rows when necessary.

```
tab_1.Multiline = TRUE
```

Name

Applies to

CallLogEntry and DialingDirectoryEntry objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The Name property holds a string associated with the phone number, typically the caller's name.

Usage

In scripts

The Name property takes a String value. The following example for a CallLogEntry writes the name associated with the entry to a single line edit box:

```
// Integer index passed into function
CallLog l_log
CallLogEntry l_logentry

l_log = CREATE CallLog
l_logentry = l_log.getEntry (index)
sle_1.text = l_entry.Name
```

NotificationID

Applies to

NotificationBubble objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Description

Defines a unique ID for a notification bubble. If you do not set the NotificationID property for a NotificationBubble object, the PocketBuilder runtime generates a unique notification ID. This ensures that the value for the notification ID is unique. When you set this property yourself, you run the risk of assigning an ID that is already used by the system.

Usage

In a painter

❖ To add an ID for a notification bubble:

- Enter the number you want in the Notification ID text box on the General page of the NotificationBubble object's Properties view.

The default value is 0. If you do not change the default value, the PocketBuilder runtime generates a unique notification ID.

In scripts

The NotificationID property takes an unsignedlong. The following example sets the ID for a notification bubble to 555:

```
nb_myBubble.NotificationID = 555
```

The following example captures the NotificationID of a NotificationBubble object and assigns it to a local variable.

```
unsignedlong ll_ID  
ll_ID = nb_myBubble.NotificationID
```

You can also save the ID for later use by assigning it to a global variable.

NumberOfSatellites

Applies to

GPSTFix objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The NumberOfSatellites property specifies how many satellites are in use.

Usage

In scripts

The NumberOfSatellites property takes an integer value. The following example writes the number of satellites to a single line edit box:

```
li_returncode = MyGPS.GetFix(myFix)
sle_1.text = "Satellites in use: " &
    + String(myFix.NumberOfSatellites)
```

Options

Applies to

SMSMessage objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Indicates the retry options if an SMS message is not delivered by the message router. Values are:

- **SMS_Option_Delivery_None!** No special handling specified.
- **SMS_Option_Delivery_No_Retry!** No attempt to resend the message will be made. When this option is not specified, the message is resent according to a short-term retry schedule.

Usage

In scripts

The Options property takes a value of the SMSMsgOptions enumerated variable. This example sets the value of the property to SMS_Option_Delivery_No_Retry!:

```
g_smsMsg.Options = SMS_OPTION_DELIVERY_NO_RETRY!
```

OriginalSize

Applies to

Picture, PictureBox, PictureHyperLink controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The OriginalSize property specifies whether the width and height of the picture are set to their original values.

In the painter, if you use the mouse to resize the control or if you set the Width or Height properties on the Position tab, the OriginalSize property becomes disabled and the check box becomes unchecked.

Usage

In a painter

❖ To set the picture to its original size:

- Select the Original Size check box on the General page of the control's Properties view.

In scripts

The OriginalSize property takes a boolean value. The following line sets the OriginalSize property to FALSE.

```
p_1.OriginalSize = FALSE
```

You should not try to change the width or height of a picture control when OriginalSize is set to TRUE because it can lead to unexpected behavior. In this example, the OriginalSize property is checked and set to FALSE before the control is doubled in size:

```
integer li_width, li_height
li_width = p_1.width * 2
li_height = p_1.height * 2
parent.setredraw(false)
p_1.setredraw(false)

if p_1.originalsize then p_1.originalsize = FALSE
p_1.width = li_width
p_1.height = li_height
p_1.setredraw(TRUE)
parent.setredraw(TRUE)
```

The `SetRedraw` function must be called only when the image is very large. Before performing multiple resize operations of large JPEG images, set the name of the picture to an empty string temporarily to avoid unnecessary lengthy recompilations.

OriginLine

Applies to

grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The `OriginLine` property specifies the style of the line that represents the value zero for that axis in the graph. In the painter, the line style settings for an axis are disabled if the axis is not appropriate for the graph type.

Usage

In a painter

❖ To select an origin line style:

- 1 Display the Axis tab page from the graph's Properties view and select the desired axis from the Axis drop-down list.
- 2 Select the desired line style from the `OriginLine` list in the Line Style group.

In scripts

The `OriginLine` property takes a value of the `LineStyle` enumerated datatype. The following statement makes the Values axis origin line a dashed line.

```
gr_1.Values.OriginLine = Dash!
```

OutlookCompatible

Applies to

POOM objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description This is a read-only property that indicates whether the version of Microsoft Pocket Outlook on the Pocket PC is compatible with Outlook on the desktop.

Usage **In scripts**
The OutlookCompatible property is a boolean datatype. The following example determines whether a POOM object is compatible with Microsoft Outlook:

```
boolean bool_compatible
bool_compatible = po_1.OutlookCompatible
if bool_compatible = false
    MessageBox("Incompatible", "This Pocket Outlook " &
        + "is not compatible with the desktop Outlook")
end if
```

OverlapPercent

Applies to Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description Overlap specifies how much 2D bar and column data markers in different series in a graph overlap. The number you specify is a percentage of the width of the data marker. This property is not applicable to all graph types.

Usage **In a painter**
❖ **To set overlap for bar or column graphs:**
1 Select the desired 2D graph type from the GraphType list on the General page of the control's Properties view.
2 Use the OverlapPercent slide control to select the desired percentage of overlap.

In scripts
The OverlapPercent property takes an integer value. The following line sets the overlap to 10%.

```
gr_1.OverlapPercent = 10
```

PageStyle

Applies to

Signature controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Specifies the style of the control, which can be one of the following:

- SigPageStyleNone! (Default style)
- SigPageStyleDottedLines!
- SigPageStyleGridLines!
- SigPageStyleLeftMargin!
- SigPageStyleRuledLines!
- SigPageStyleTopMargin!
- SigPageStyleTopLeft Margin!
- SigPageStyleYellowBackground!

Usage

In a painter

❖ To select a page style for the control:

- Select the value you want from the Page Style drop-down list on the General page of the control's Properties view.

In scripts

The PageStyle property takes an enumerated datatype. The following example sets the page style for the control to a yellow background:

```
sig_mine.PageStyle = SIGPageStyleRuledLines!
```

Password

Applies to

SingleLineEdit controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The Password property specifies whether the control is a password field, in which characters the user types appear as asterisks (*). If Password is not enabled, the characters appear as the user types them.

Usage

In a painter

❖ To make the control a password field:

- Select the Password check box on the General page of the control's Properties view.

In scripts

The Password property takes a boolean value.

The following example sets the SingleLineEdit to a password field so that characters typed in appear as asterisks.

```
sle_1.Password = TRUE
```

PenMode

Applies to

Signature controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Specifies the drawing mode. Values are:

- SIGPenModePen! – User can write or draw (default)
- SIGPenModeSelect! – User can select text
- SIGPenModeSpace! – User can collapse and expand white space

Usage**In a painter**❖ **To select a pen mode for the control:**

- Select the value you want from the Pen Mode drop-down list on the General page of the control's Properties view.

In scripts

The PenMode property takes an enumerated datatype. The following example sets the pen mode for the control to enable text selection:

```
sig_mine.PenMode = SigPenModeSelect!
```

PerpendicularText

Applies to

Tab controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When PerpendicularText is enabled, the tab labels are drawn perpendicular to the tab page, resulting in narrower tabs. When PerpendicularText is not enabled, text runs parallel to the edge of the tab page, resulting in wider tabs.

Usage**In a painter**❖ **To select a perpendicular orientation for tab text:**

- Select the Perpendicular Text check box on the General page of the control's Properties view.

In scripts

The PerpendicularText property takes a boolean value.

The following line specifies that tab labels are perpendicular to the tab page.

```
tab_1.PerpendicularText = TRUE
```

Perspective

Applies to Graph controls Properties view

PocketBuilder	✗
PowerBuilder	✓

Description Perspective controls the distance of a 3D graph from the front of the window. Perspective is not available for 2D graphs.

PhoneNumber

Applies to CallLogEntry, DialingDirectoryEntry, and PhoneCall objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description A string containing a valid phone number.

A phone number consists of an optional country or region code followed by a sequence of numbers that makes up the phone number. International numbers must be preceded by a plus (+) sign. The PhoneNumber string can contain parentheses, periods, dashes, and spaces, all of which are ignored.

Usage

- In a painter**
- ❖ **To set the PhoneNumber property of a PhoneCall object:**
 - Enter the phone number in the PhoneNumber text box on the General page of the Properties view for the PhoneCall object.
- The PhoneNumber property of CallLogEntry and DialingDirectoryEntry objects cannot be set in a painter.

In scripts

The PhoneNumber property is a string that contains a valid phone number.

For CallLogEntry and PhoneCall objects, the PhoneNumber property contains the number of the other party in an ingoing or outgoing call. If the caller ID of an incoming call is blocked, the PhoneNumber property is empty.

For `DialingDirectoryEntry` structures, the phone number is the number to be dialed.

The following statements create an instance of a call log and add all the phone numbers in the log to a list box:

```
// instance variable: CallLogEntry iLogEntries[]

CallLog callLog1

callLog1 = CREATE CallLog
callLog1.getEntries(iLoggedEntries)

int count
for count = 1 to UpperBound(iLoggedEntries)
    lb_1.AddItem(iLoggedEntries[count].PhoneNumber)
end for
```

The following statement displays a phone number in a single line edit box, where *newEntry* is a `DialingDirectoryEntry` structure:

```
sle_call.text = newEntry.PhoneNumber
```

The following statement sets the `PhoneNumber` property of a `PhoneCall` object to the string entered in a single line edit box::

```
pcall_1.PhoneNumber = sle_newcall.text
```

PhoneNumberPlan

Applies to

`DialingDirectoryEntry` objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Indicates the calling (numbering) plan to use when dialing. Values are:

- **0** Unspecified or unknown
- **1** ISDN/Telephone numbering plan
- **2** Data numbering plan (X.121)

- **3** Telex numbering plan
- **4** Private numbering plan
- **5** ERMES numbering plan

Usage

In scripts

The following example tests whether the `PhoneNumberType` property is set for a given call log entry, and writes its value to a multiline edit box if it is set:

```
CallLogEntry l_entry
integer index
string ls_type

l_entry = clog_1.GetEntry(index)
if (l_entry.PhoneNumberType <> "") then
    mle_1.text += "Type: " + l_entry.PhoneNumberType
else
    mle_1.text += "Type: Unknown"
end if
```

PhoneNumberType

Applies to

CallLogEntry and DialingDirectoryEntry objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Indicates the type of phone number.

For a `CallLogEntry`, this is a string value, typically *w* for a work phone number, *h* for a home number, and *m* for a mobile number.

For a `DialingDirectoryEntry`, this is an integer indicating the protocol to use when dialing. Values are:

- **0** Unspecified or unknown
- **1** International
- **2** National
- **3** Network specific

- **4** Subscriber number (protocol specific)
- **5** Alphanumeric address
- **6** Abbreviated (speed dial) number

Usage**In scripts**

The following example tests whether the `PhoneNumberType` property is set for a given call log entry, and writes its value to a multiline edit box if it is set:

```
CallLogEntry l_entry
integer index
string ls_type

l_entry = clog_1.GetEntry(index)
if (l_entry.PhoneNumberType <> "") then
    mle_1.text += "Type: " + l_entry.PhoneNumberType
else
    mle_1.text += "Type: Unknown"
end if
```

PicturesAsFrame

Applies to

RichTextEdit controls

PocketBuilder	✗
PowerBuilder	✓

Description

When `PicturesAsFrame` is enabled, any bitmaps used in the control will appear as empty frames. If this property is not enabled, graphics will appear normally.

PictureHeight

Applies to

TreeView and Toolbar controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description The `PictureHeight` property specifies in pixels the display height of all the pictures in the control. The `Toolbar` control shrinks or grows to fit the height you set. Pictures you select for the toolbar buttons are automatically scaled to the `PictureWidth` and `Picture Height` settings.

For the `Toolbar` control, you can set the `PictureWidth` and `PictureHeight` properties in script whether or not there are images in the `PictureName` property array. For the `TreeView` control, the `PictureName` array must be empty before you can set the `PictureWidth` and `PictureHeight` properties in script, although you can change these values in the painter whether or not there are images in the `Picture` list.

Usage **In a painter**

- ❖ **To set the picture height:**
 - Select the desired value from the `Height` drop-down list on the `Pictures` tab page of the control's `Properties` view.

The choices of 16 and 32 are standard pixel heights for icons. If you select `Default`, `PocketBuilder` uses the height of the first picture in the `PictureName` array as the height for all the pictures.

In scripts

The `PictureHeight` property takes an integer value. This value can only be set before the first call to the `AddPicture` function or after calling `DeletePictures`. If this value is set to 0, then the size of the first picture in the `PictureName` property array is used as the height for all the pictures.

The following line sets the height for a `TreeView`'s pictures to 16 pixels.

```
tv_1.PictureHeight = 16
```

PictureIndex

Applies to

`ListViewItem`, `TreeViewItem`

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description The `PictureIndex` property identifies pictures in the control's Picture list. For `ListViewItems`, the index identifies the large, small, and state picture associated with the item.

For `TreeViewItems`, the index identifies the picture displayed to the left of the item label. If the index is 0, no picture is displayed. You can only set the `PictureIndex` property for `TreeViewItems` with scripts, but you can add pictures to the control's Picture list in the painter.

Usage**In a painter****❖ To associate pictures with a ListView item:**

- 1 Select the Large Picture tab page, Small Picture tab page, or State tab page from the `ListView` control's Properties view.
- 2 Do one of the following:
 - In the rows provided in the Picture Name field, type the complete path and name of the files containing the desired pictures.
 - Use the Browse button.
 - Select one or more pictures from the Stock Pictures list.
- 3 Select the Items tab page from the `ListView` control's Properties view
- 4 Use the row numbers from the Picture Name list to specify the `PictureIndex` for each List View Item on the Items tab page.

❖ To add pictures to a TreeView control's picture list:

- 1 Select the Pictures tab page from the `TreeView` control's Properties view.
- 2 Do one of the following:
 - In the rows provided in the Picture Name field, type the complete path and name of the files containing the desired pictures.
 - Use the Browse button.
 - Select one or more pictures from the Stock Pictures list.

You associate pictures in the `TreeView` control's picture list with `TreeViewItems` using scripts.

In scripts

This example illustrates how to get a `ListViewItem` object and change the value of `PictureIndex`.

```
listviewitem lvi  
lv_1.GetItem(4, lvi)
```

```
lvi.PictureIndex = 2
lv_1.SetItem(4, lvi)
```

For more information about scripting ListView and TreeView controls, see “Using ListView Controls” and “Using TreeView Controls” in the *Resource Guide*.

PictureMaskColor

Applies to TreeView controls, TabPage user objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description The PictureMaskColor property specifies the color in the picture that is transparent when the picture is displayed. You can change the mask color before adding each picture. Each image uses the mask color that was in effect when it was added.

Usage **In a painter**

To add your own colors to the color drop-down list, select Design>Custom Colors before displaying the Properties view.

- ❖ **To set the picture mask color for ListBox and TreeView controls:**
 - 1 Select the Pictures tab page of the control’s Properties view.
 - 2 Select the desired color from the Picture Mask Color drop-down list box.
- ❖ **To set the picture mask color for TabPage objects in a tab control:**
 - 1 Select the desired TabPage object of the tab control.
 - 2 Select the Picture tab page of the TabPage object’s Properties view.
 - 3 Select the desired color from the Picture Mask Color drop-down list box.
- ❖ **To set the picture mask color for a TabPage user object:**
 - 1 Select the TabPage tab page of the user object’s Properties view.
 - 2 Select the desired color from the Picture Mask Color drop-down list box.

The mask color selected for the user object can be changed after it has been inserted into a tab control.

In scripts

The `PictureMaskColor` property takes a long value (-2 to 16,777,215) that specifies the numerical value of the mask color. The `PictureMaskColor` value is a combination of values for the red, green, and blue components of the color. If you do not know the long value for a particular color, choose **Design>Custom Colors** to determine the red, green, and blue values and then call the `RGB` function to specify the color in a script. In scripts, this property is used when each picture is added and, therefore, can be changed between `AddPicture` calls.

The following example sets yellow as the mask color for pictures in a `DropDownPictureListBox`.

```
ddplb_1.PictureMaskColor = RGB(255, 255, 0)
```

PictureName

Applies to

Picture, `PictureButton`, and `PictureHyperLink` controls, and `UserObject` used as tab page

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The `PictureName` property specifies the name of the file that contains the picture displayed in the control. For `PictureButton` controls, the picture specified by the `PictureName` property is the picture that is displayed when the button is enabled.

The picture can be in the following formats:

- bitmap (*.BMP*)
- GIF (*.GIF*)
- JPEG (*.JPG* or *.JPEG*)

Usage

In a painter

- ❖ **To specify the picture for a Picture control and for the enabled state of a PictureButton:**
 - Enter the name of the file in the PictureName field on the General page of the control's Properties view, or use the Browse button next to the PictureName field to select a file.

In scripts

The PictureName property takes a string value.

The following line selects a picture file for a PictureButton pb_1.

```
pb_1.PictureName = "c:\pictures\pb1.bmp"
```

PictureName[]

Applies to

TreeView, Toolbar controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The PictureName[] property specifies an indexed array of files containing the pictures used in the control. You can add pictures to the array in the painter, or use the AddPicture function at execution. However, adding or deleting pictures during execution does not update the PictureName property array.

The pictures can be in the following formats:

- bitmap (.BMP)
- GIF (.GIF)
- JPEG (.JPG or .JPEG)

Usage

In a painter

- ❖ **To add pictures to the PictureName array:**
 - 1 Select the Pictures tab page from the control's Properties view.
 - 2 Do one of the following:
 - In the rows provided in the Picture Name field, type the complete path and name of the files containing the desired pictures.

- Use the Browse button.
- Select one or more pictures from the Stock Pictures list.

In scripts

The `PictureName` property array is populated at initialization and cannot be updated during execution.

The following example adds a picture to a `TreeView` control and associates it with a new `TreeView` item:

```
long ll_tvi
integer li_picture
li_picture = tv_1.AddPicture("c:\images\new.gif"
ll_tvi = tv_1.FindItem(RootTreeItem!, 0)
tv_1.InsertItemFirst(ll_tvi, "New", li_picture)
```

PictureOnRight

Applies to

Tab controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

When the `PictureOnRight` property is enabled, the picture, if any, that is part of the tab label is to the right of the text. When `PictureOnRight` is not enabled, the picture is to the left of the tab label text.

Usage

In a painter

❖ To specify pictures to the right of text on tab labels:

- Select The Pictures on Right check box on the General page of the tab control's Properties view.

In scripts

The `PictureOnRight` property takes a boolean value.

The following line puts pictures to the right of the tab labels.

```
tab_1.PictureOnRight = TRUE
```

PictureWidth

Applies to

TreeView, Toolbar controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The PictureWidth property specifies in pixels the display width of all the pictures in the control. Pictures you select for buttons on the Toolbar control are automatically scaled to the PictureWidth and Picture Height settings.

For the Toolbar control, you can set the PictureWidth and PictureHeight properties in script whether or not there are images in the PictureName property array. For the TreeView control, the PictureName array must be empty before you can set the PictureWidth and PictureHeight properties in script, although you can change these values in the painter whether or not there are images in the Picture list.

Usage

In a painter

❖ To set the picture width:

- Select the desired value from the Width drop-down list on the Pictures tab page of the control's Properties view.

The choices of 16 and 32 are standard pixel widths for icons. If you select Default, PocketBuilder uses the width of the first picture in the PictureName array as the width for all the pictures.

In scripts

The PictureWidth property takes an integer value. This value can only be set before the first call to the AddPicture function or after calling DeletePictures. If this value is set to 0, then the size of the first picture in the PictureName property array is used as the width for all the pictures.

The following line sets the width for a TreeView's pictures to 16 pixels.

```
tv_1.PictureWidth = 16
```

Pointer

Applies to

All controls

PocketBuilder	✗
PowerBuilder	✓

Description

The Pointer property specifies the pointer image displayed when the pointer is over a control.

PopupMenu

Applies to

RichTextEdit controls

PocketBuilder	✗
PowerBuilder	✓

Description

When PopMenu is enabled, the user has access to a pop-up menu by clicking the right mouse button on the control. The pop-up menu allows the user to cut and paste, insert a file, and select formatting properties.

Port

Applies to

Camera objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The Port property must be set for connections to HP and VEO camera devices before you call the Open function. This property is ignored for connections to HTC camera devices.

Usage

In a painter

❖ To set the Port property

- 1 Select the Camera object in the Non-Visual Object List and open the Properties view.

- 2 Select the CameraType you want from the drop-down list.
- 3 Type a value for the Port property—typically SIO1:

In scripts

The Port property takes a string. This example uses the SDIO port for an HP Photosmart camera device:

```
cam_1.CameraType = 71
cam_1.Port="SIO1:"
cam_1.Open(w_myphoto_main)
...
```

Position

The position porperty has different meanings and values for the scroll bar and toolbar controls.

Applies to

HScrollBar, VScrollBar controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

Position specifies where the scroll box or thumb will appear when the scroll bar is first displayed during execution.

Usage

In a painter

- ❖ **To set the initial position of the scroll box or thumb:**
 - Type a number that is between the values you have specified in MinPosition and MaxPosition.

In scripts

The Position property for scroll bar controls takes an integer value. It should be used in conjunction with MaxPosition and Min Position.

For example, if the scroll bar's minimum is 0 and maximum is 100, this statement positions the scroll box 80 percent of the way toward the bottom.

```
vsb_1.Position = 80
```

Applies to

Toolbar controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Description

Determines where on the window the toolbar is docked. Values are:

- **AlignAtBottom!** Docks toolbar at bottom of window or user object
- **AlignAtTop!** Docks toolbar at top of window or user object
- **Floating!** Allows you to position the toolbar using x and y coordinates

Usage

In a painter❖ **To select an alignment for a toolbar:**

- Enter an enumerated value in the Position list box on the General page of the control's Properties view.

In scripts

The Position property takes an enumerated value of type `ToolbarAlignment`. The following example displays the toolbar at the top of a parent window:

```
tlbr_myToolbar.Position = AlignAtTop!
```

PowerTipText

Applies to

Picture, PictureBox, and PictureHyperlink controls, and UserObjects with tab pages

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

Displays a PowerTip when the user moves a cursor over the control or over the tab area of the tab page.

Usage

In a painter❖ **To set the PowerTip:**

- 1 In the Window painter, display the General page of the control's Properties view, or in the User Object or Window painter, display the TabPage tab of the UserObject's Properties view.

2 Type a PowerTip in the box for the PowerTipText field.

In scripts

The PowerTipText property takes a string value.

The following line adds a PowerTip for tabpage_2 on tab control tab_1:

```
tab_1.tabpage_2.PowerTipText = "Cancel the operation"
```

This adds a PowerTip for a PictureBox control:

```
pb_1.PowerTipText = "This button opens a new form"
```

PowerTips

Applies to

Tab controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

When the PowerTips property is enabled, any PowerTip text defined for a tab page is displayed as pop-up text when the mouse pointer pauses over the tab. PowerTips are useful if the tabs are labeled with pictures.

Usage

In a painter

- ❖ **To enable the display of PowerTip text:**
 - Select the Power Tips check box on the General page of the Tab control's Properties view.

In scripts

The PowerTips property takes a boolean value. The following line allows display of PowerTips for each tab page.

```
tab_1.PowerTips = TRUE
```

PrimaryLine

Applies to

grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The PrimaryLine property specifies the line style for the primary line used for the axis itself. The line style settings for an axis are disabled if the axis is not appropriate for the graph type. Primary lines are not visible if the line style is set to transparent!.

Usage

In a painter

❖ To set the primary line style for an axis:

- 1 Display the Axis tab page from the graph's Properties view and select the desired axis from the Axis drop-down list.
- 2 Select the desired line style from the PrimaryLine drop-down list in the Line Style group.

In scripts

The PrimaryLine property takes a value of the LineStyle enumerated datatype. The following line sets the PrimaryLine property for the Values axis of a Graph to a dash.

```
gr_1.Values.PrimaryLine = Dash!
```

PRN

Applies to

GPSSatellitePosition objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The PRN property holds the satellite's PRN number, which is a unique number associated with a satellite.

Usage

In scripts

The PRN property takes an integer value:

```
integer li_PRN
GPSSatellitePosition myGPS_SP

...
li_PRN = myGPS_SP.PRN
```

ProtocolType

Applies to

SMSProtocol objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Specifies which SMS protocol type an SMSSession object uses. Values are:

- **SMS_MsgType_Text!** Text SMS Protocol
- **SMS_MsgType_Notification!** Notification SMS Protocol (receive only)
- **SMS_MsgType_WDP!** Wireless Datagram Protocol (WDP)
- **SMS_MsgType_WCMP!** Wireless Control Message Protocol (WCMP)
- **SMS_MsgType_Status!** Status Message SMS Protocol (receive only)
- **SMS_MsgType_Broadcast!** Broadcast Message SMS Protocol (receive only)
- **SMS_MsgType_Raw!** Raw SMS Protocol (receive only)

Usage

In scripts

The ProtocolType property takes a value of the SMSProtocolType enumerated variable. This example sets the protocol type of an SMSProtocol object to text:

```
SMSProtocol mysmsprotocol
mysmsprotocol.ProtocolType = SMS_MsgType_Text!
```

RaggedRight

Applies to

Tab controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

When `RaggedRight` is enabled, tab size is determined by the label text and the Fixed Width setting. If `RaggedRight` is not enabled, tabs are stretched so that they fill space along the edge of the control.

Usage

In a painter

❖ To set the `RaggedRight` property:

- Select the Ragged Right check box on the General page of the tab control's Properties view.

When this check box is selected, the tabs are sized based on their label text and whether the Fixed Width check box is selected.

In scripts

The `RaggedRight` property takes a boolean value. The following line specifies that tabs are stretched to fill the edge of the control.

```
tab_1.RaggedRight = FALSE
```

RawData

Applies to

GPS and SerialGPS objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

RawData is a read-only property that contains raw data formatted according to the National Marine Electronics Association (NMEA) 0183 standard. NMEA-0183 data is transmitted in “sentences”, each of which typically begins with an identifier that consists of a dollar sign, two-letter talker ID, and three-letter sentence ID. This identifier determines the meaning of each of the data fields in the rest of the sentence. The data fields follow the identifier and are separated by commas.

The RawData property returns the NMEA sentences that are in the input buffer. The string is not parsed for validity and there is no guarantee that the data is valid or correlated with the recent fix, satellite, or heading reading.

For GPS objects, RawData is first populated after the Open call. It contains the entire contents of the string literal or data file used to open the GPS object. After the first successful GetFix, GetHeading, or GetSatellitesInView request, RawData contains the NMEA sentences that were parsed up to and including the \$GPGGA sentence that was actually processed. If a second request is issued, parsing begins with the sentence following the last \$GPGGA sentence and completes with the next sentence found. After successfully obtaining the second \$GPGGA sentence, the RawData property is updated accordingly.

RawData is not updated if an error occurs or the end of a buffer is reached.

For SerialGPS objects, RawData is populated in a similar manner, except that the property is not populated during an Open call.

Usage

In a painter

Type the value in the RawData box on the General page of the Object's Properties view.

In scripts

The RawData property takes a string value. This example writes the RawData property to a single line edit box:

```
string myRawData
GPS myGPS
Integer li_rc

myGPS = CREATE GPS
li_rc = myGPS.Open()
...
myRawData = myGPS.RawData
sle_1.txt = "Raw data: " + myRawData
```

Resizable

Applies to

DataWindow, OLE, and RichTextEdit controls and windows

PocketBuilder	✗
PowerBuilder	✓

Description

A resizable window or control has a thick border, and the user can use the mouse or the keyboard to resize it.

ReturnsVisible

Applies to

RichTextEdit controls

PocketBuilder	✗
PowerBuilder	✓

Description

When the ReturnsVisible property is enabled, characters for carriage returns in the text will display.

RightMargin

Applies to

RichTextEdit controls

PocketBuilder	✗
PowerBuilder	✓

Description

The RightMargin property specifies the size in inches of the right margin on the printed page.

RightToLeft

Applies to Application and Window objects, and CheckBox, DropDownListbox, DropDownPictureListBox, EditMask, GroupBox, ListBox, PictureListBox, MultiLineEdit, RadioButton, SingleLineEdit, StaticHyperLink, and StaticText controls

PocketBuilder	✗
PowerBuilder	✓

Description The RightToLeft property specifies that characters should be displayed in right-to-left order. The application must be running on an operating system that supports right-to-left display.

Rotation

Applies to Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description The Rotation property specifies the rotation from left to right of 3-D graphs. Rotation is disabled for 2D graphs.

Usage **In a painter**

- ❖ **To set the rotation of the graph:**
 - 1 Select a 3D graph type from the GraphType list on the General page of the graph control's Properties view.
 - 2 Move the Rotation slider to change the graph's rotation.

In scripts

The Rotation property takes an integer value.
The following example rotates the graph 45 degrees to the left:

```
gr_1.Rotation = -45
```

RulerBar

Applies to

RichTextEdit controls

PocketBuilder	✗
PowerBuilder	✓

Description

When the RulerBar property is enabled, a ruler bar appears above the editing area of the control. The user can use it to set tabs and margins on the tab bar.

Roaming

Applies to

CallLogEntry objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Indicates whether the call was placed or received in the local calling area or while roaming. Values are:

- **true** Call was placed or received while roaming
- **false** Call was placed or received locally

Usage

In scripts

The Roaming property is a boolean indicating whether an entry in the call log was placed or received while roaming. In this example, a listbox is populated with the phone numbers in a call log. The following lines in the SelectionChanged event of the listbox get the value of the Roaming property of the selected phone number and write it to a second listbox:

```
// instance variable: CallLogEntry iLogEntries[]

if (iLoggedEntries[index].roaming = true
    lb_2.AddItem("Roaming")
else
    lb_2.AddItem("Local")
end if
```

RoundTo

Applies to

grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When the AutoScale property is enabled, the RoundTo and RoundToUnit properties specify how to round the end points and tick marks of an axis. Rounding affects axis labels, not graph data.

The RoundTo property specifies the value to which you want to round the axis values, in the units specified by the RoundToUnit property.

Usage

In a painter

❖ To set the value to which to round axis values:

- 1 Display the Axis tab page from the graph control's Properties view and select the desired axis from the Axis list.
- 2 Turn on autoscaling by checking the AutoScale check box.
- 3 Choose the datatype of the axis by selecting an option from the DataType drop-down list.
- 4 Enter a value in the RoundTo edit field.

In scripts

The RoundTo property takes a double value indicating the multiple to which you want to round axis tick marks.

The following example sets the datatype of the Values axis to date, sets the unit for rounding to months, and then sets the rounding value to six months:

```
gr_1.Values.DataType = AdtDate!  
gr_1.Values.RoundToUnit = RndMonths!  
gr_1.Values.RoundTo = 6
```

RoundToUnit

Applies to

grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When the AutoScale property is enabled, the RoundTo and RoundToUnit properties specify how to round the end points and tick marks of an axis. Rounding affects axis labels, not graph data.

The RoundToUnit property specifies the type of units that should be used for the rounding. The type of units that can be specified are based on the datatype of the axis. For example, for a date axis, you might round tick marks to the nearest five years or to every third month.

Usage

In a painter

❖ To specify the type of unit to be used for rounding:

- 1 Display the Axis tab page of the graph control's Properties view and select the desired axis from the Axis list.
- 2 Turn on autoscaling by checking the AutoScale check box.
- 3 Choose the datatype of the axis by selecting an option from the DataType drop-down list.
- 4 Choose the desired unit from the RoundToUnit drop-down list.

In scripts

The RoundToUnit property takes a value of the enumerated datatype grRoundToType. When you set this property in scripts, make sure the value is compatible with the datatype of the axis.

The following example sets the datatype of the Values axis to date and then sets the unit for rounding to months and the number of months to which to round:

```
gr_1.Values.DataType = AdtDate!
gr_1.Values.RoundToUnit = RndMonths!
gr_1.Values.RoundTo = 6
```

Satellite[]

Applies to

GPSSatellitesInView objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The Satellite[] property array returns information about the azimuth, elevation, signal strength, and PRN number of each of the satellites currently in view.

Usage

In scripts

The Satellite property holds an array of GPSSatellitePosition structures. The following statements create an instance of a GPSSatellitesInView object and write information about each satellite to a multiline edit box:

```
// instance variable: GPSSatellitePosition iGPS_SP[]

SerialGPS myGPS
GPSSatellitesInView mySIV
Integer rc

myGPS = CREATE SerialGPS
rc = myGPS.open()
mySIV = myGPS.getSatellitesInView()

int count
for count = 1 to UpperBound(iGPS_SP)
    mle_1.text += String(iGPS_SP[i].PRN + " "
    mle_1.text += String(iGPS_SP[i].Azimuth + " "
    mle_1.text += String(iGPS_SP[i].Elevation + " "
    mle_1.text += String(iGPS_SP[i].SNR + "~r~n "
end for
```

ScaleType

Applies to

grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description The `ScaleType` property specifies the scale used for an axis. An axis can have linear or logarithmic scaling. The default is `Linear`. Other values are `Log10` and `LogE`.

Usage

In a painter

❖ To select the scale type for an axis:

- 1 Display the Axis tab page of the graph control's Properties view and select the desired axis from the Axis list.
- 2 Select the desired type from the `ScaleType` drop-down list in the Scale group.

In scripts

The `ScaleType` property takes a value of the `grScaleType` enumerated datatype.

To set the scale type of the Values axis of `gr_1` to log 10, use a line like the following.

```
gr_1.Values.ScaleType=Log10!
```

ScaleValue

Applies to

`grAxis` objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The `ScaleValue` property specifies the scale of values on the axis. You cannot set this property in the painter.

Usage

In scripts

The `ScaleValue` property takes a value of the `grScaleValue` enumerated datatype.

The following line sets the `ScaleValue` of the Values axis of a graph:

```
gr_1.Values.ScaleValue = Actual!
```

ScannedData

Applies to

BarcodeScanner objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Description

Specifies the data from the most recent scan after a RetrieveData call.

Usage

The ScannedData property is a read-only string.

Examples

The following example retrieves data from a single scan:

```
Integer l_iret
l_iret = l_scanner.Open()
l_iret = l_scanner.ScanWait( 30 )
l_iret = l_scanner.RetrieveData()
sl_data.text = l_scanner.ScannedData
```

ScannedSymbology

Applies to

BarcodeScanner objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Description

Specifies the bar code symbology type from the most recent scan after a RetrieveData call.

Usage

The ScannedSymbology property is a read-only integer specifying the decoder ID.

Examples

The following example retrieves the type of bar code from a bar code scan:

```
Integer l_iret, l_id
l_iret = l_scanner.Open()
l_iret = l_scanner.ScanWait( 30 )
l_iret = l_scanner.RetrieveData()
l_id = l_scanner.ScannedSymbology
sl_symbology.text = l_scanner.DecoderName(l_id)
```

ScannedTimeStamp

Applies to

BarcodeScanner objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Description

Specifies the time of the most recent scan after a RetrieveData call.

Usage

The ScannedTimeStamp property is a datetime datatype.

Examples

The following example retrieves the timestamp from a single bar code scan:

```
Integer l_iret
l_iret = l_scanner.Open()
l_iret = l_scanner.ScanWait( 30 )
l_iret = l_scanner.RetrieveData()
sl_data.text = l_scanner.ScannedData
sl_time.text = String (l_scanner.ScannedTimeStamp)
```

ScannerType

Applies to

BiometricScanner objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Description

Specifies the bitset of authentication techniques used by the biometric scanner, including voice, fingerprint, and iris recognition.

Usage

The ScannerType property takes a string. For the HPBiometricScanner object, it defaults to the only supported value (fingerprint).

Examples

The following example retrieves the authentication type from the current scanner:

```
Integer l_iret
l_iret = l_scanner.Open()
sl_type.text = string (l_scanner.ScannerType)
```

Scrolling

Applies to

ListView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When Scrolling is enabled, the user can scroll vertically when some of the items in a ListView control are not visible. When Scrolling is not enabled, the user cannot scroll.

Usage

In a painter

❖ To enable scrolling:

- Select the Scrolling check box on the General page of the control's Properties view.

In scripts

The Scrolling property takes a boolean value. The following line enables scrolling when necessary in a ListView.

```
lv_1.Scrolling = TRUE
```

SecondaryLine

Applies to

grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The SecondaryLine property specifies the style of the lines used in the axis parallel to and opposite the primary axis in the graph.

Usage

In a painter

The line style settings for an axis are disabled in the painter if the axis isn't appropriate for the graph type.

❖ **To set the secondary line style:**

- 1 Display the Axis tab from the graph control's Properties view and select the desired axis from the Axis list.
- 2 Select the desired line style from the SecondaryLine drop-down list in the Line Style group.

In scripts

The SecondaryLine property takes a value of the LineStyle enumerated datatype. The following example sets the SecondaryLine property of the Values axis of a graph to a dash.

```
gr_1.Values.SecondaryLine = Dash!
```

SelectedTab

Applies to

Tab controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

The SelectedTab property specifies the index number of the selected tab page in the tab control. As the user selects tabs in the Tab control, the value of SelectedTab changes to reflect the currently selected tab.

Usage

In a painter❖ **To specify the selected tab:**

- Enter a number in the Selected Tab field on the General page of the control's Properties view.

The number should be in the range 1 to N where N is the number of tab pages in the tab control.

In scripts

The SelectedTab property takes an integer value. The default value is 1, and the integer must be in the range 1 to *N*, where *N* is the number of tab pages.

The following line sets the index number of the selected tab page in the Tab control tab_1 to 3.

```
tab_1.SelectedTab = 3
```

SerialPort

Applies to

SerialGPS objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The SerialPort property specifies the serial port used by the GPS device. For Bluetooth devices, the value of this property is typically COM8.

Usage

In a painter

Type the value in the SerialPort box on the General page of the Object's Properties view.

In scripts

The SerialPort property takes a string value. This example overrides the default serial port before opening a GPS connection and getting a location fix:

```
gps_1.SerialPort = "COM5:" //override default com8 port
gps_1.ConfigParams = "buffersize=2000,refresh=2000," &
+ "timeout=5000,multithread=0"
rc = gps_1.Open()
IF rc = 1 THEN
    rc = gps_1.GetFix(myFix)
    ... //process data
ELSE //process error msg
END IF
rc = gps_1.Close()
```

Series

Applies to

Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The Series property is used to define the properties of the Series axis in a graph. The Series axis is valid for 3D graphs only.

Usage

In a painter

❖ To define the properties of the Series axis of a 3D graph:

- 1 Select a 3D graph type from the GraphType list on the General page of the graph control's Properties view.
- 2 Display the Axis tab page of the graph control's Properties view and select Series in the Axis list.

All the properties of the Series axis can be set from the Axis tab page.

In scripts

The Series axis is an object of type `grAxis` within the Graph control. The Series object has its own properties for controlling the axis' appearance.

The following line sets the scale type of the Series axis of `gr_1` to log 10.

```
gr_1.Series.Scaletype = Log10!
```

SeriesSort

Applies to

Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The SeriesSort property specifies how the series are sorted: ascending, descending, or unsorted.

Usage

In a painter

- ❖ **To specify how the series are sorted:**
 - Select the desired sort type from the SeriesSort drop-down list on the General page of the graph control’s Properties view.

In scripts

The datatype of the CategorySort property is the grSortType enumerated datatype which has the values Ascending!, Descending!, Unsorted!, and UserDefinedSort!.

The following example specifies that the series should be unsorted:

```
gr_1.SeriesSort = Unsorted!
```

SetStep

Applies to

HProgressBar and VProgressBar controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

A progress bar has a range and a current position. The SetStep property allows you to set the size of the increments by which the current position advances as progress is shown. The default value is 10.

Usage

In a painter

- ❖ **To set the increment size:**
 - Use the spin control or enter an integer in the SetStep text box on the General page of the control’s Properties view.

In a script

SetStep takes an integer value. In the following example, the range of the progress bar is set to 0 to 500, and the step value is set to 50:

```
hpb_1.setrange(0,500)
hpb_1.setstep = 50
```

ShadeBackEdge

Applies to

grAxis objects in Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

Specifies whether the back edge of an axis is shaded. Applies only to 3D graphs. The shade color is a property of the graph, not the axis.

Usage

In a painter

❖ To shade the back edge of an axis in a 3D graph:

- 1 Select a 3D graph type from the GraphType list on the General page of the graph control's Properties view.
- 2 Select a shade color from the ShadeColor list on the General page.
- 3 Display the Axis tab page of the graph control's Properties view and select the desired axis from the Axis list.
- 4 Select the ShadeBackEdge check box on the Axis tab page.

In scripts

The ShadeBackEdge property takes a boolean value.

The following example selects the shade color for the graph and then specifies that the back edge of the Category axis in a 3D graph is shaded.

```
gr_1.ShadeColor = RGB(240,250,150)
gr_1.Category.ShadeBackEdge = TRUE
```

ShowList

Applies to

DropDownListBox, DropDownPictureListBox controls

PocketBuilder	✗
PowerBuilder	✓

Description

If the ShowList property is enabled, the option list is always displayed. If this property is not enabled, the list is displayed only when the user clicks on the control's down arrow.

ShowHeader

Applies to

ListView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

When the ShowHeader property is enabled, column titles appear in the report view of a ListView control. When ShowHeader is not enabled, column titles do not appear in the report view.

To enable report view in a ListView control, you must write a script that establishes and populates columns. See “Using ListView controls” in the *Resource Guide* for more information about enabling report view.

Usage

In a painter

- ❖ **To specify a header for report view:**
 - Select the Show Header check box on the General page of the control’s Properties view.

In scripts

The ShowHeader property takes a boolean value. The following line enables display of a header in report view.

```
lv_1.ShowHeader = TRUE
```

ShowPicture

Applies to

Tab controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

When the ShowPicture property is enabled, the picture specified for each tab, if any, is displayed. When this property is not enabled, no pictures appear.

You can use ShowPicture with ShowText to display a picture and a text label, picture only, text label only, or no label at all.

Usage**In a painter**

- ❖ **To show the pictures on the tab pages in the tab control:**
 - Select the Show Pictures check box on the General page of the tab control's Properties view.

In scripts

The ShowPicture property takes a boolean value. The following line allows the picture, if any, to appear for each tab.

```
tab_1.ShowPicture = TRUE
```

ShowSIPButton

Applies to**Windows**

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Description

The ShowSIPButton property specifies whether the soft input panel (SIP) button is displayed in a window.

Usage**In a painter**

- ❖ **To display the SIP button in a window:**
 - Select the Show SIP Button check box on the General page of the window's Properties view.

If the MenuBar property is also set, the SIP button displays in the menu bar region. If you turn off the MenuBar property, the SIP button displays in the window, and you should make sure that you do not place any controls in the region in which it displays.

In scripts

This property cannot be set in a script.

ShowText

Applies to

Tab controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

When the ShowText property is enabled, the text specified for each tab, if any, is displayed. When this property is not enabled, no text appear.

You can use ShowText with ShowPicture to display a picture and a text label, picture only, text label only, or no label at all.

Usage

In a painter

❖ **To show the text on the tab pages in the tab control:**

- Select the Show Text check box on the General page of the tab control's Properties view.

In scripts

The ShowText property takes a boolean value. The following line allows the text, if any, to appear for each tab.

```
tab_1.ShowText = TRUE
```

SIPOnFocus

Applies to

EditMask, MultilineEdit, and SingleLineEdit controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Description

Determines whether the SIP displays when a user changes focus to an editable control. Values are:

- TRUE – SIP displays when user changes focus to the control
- FALSE – (Default) SIP does not automatically display

This property has no effect on an application running on the desktop.

Usage

In a painter❖ **To select the SIP for automatic display:**

- Select the Show SIP On Focus check box on the General page of the control's Properties view.

Do not set in script

Do not set this property in script. SIPOnFocus property values are cached in internal structures for performance reasons.

SmallPictureHeight

Applies to

ListView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The SmallPictureHeight property specifies the display height of all the pictures in the Small Icon view of the ListView control. The size is specified in pixels.

If you choose the value (Default) in the painter, or set the value to 0, PocketBuilder uses the height of the first picture in the array as the height for all the pictures. The other choices in the painter, 16 and 32, are standard pixel heights for icons.

The type of picture used is determined by the value of the View property of the control.

Usage

In a painter❖ **To set the small picture height:**

- Select a value from the Height drop-down list on the Small Picture tab page of the control's Properties view.

In scripts

The SmallPictureHeight property takes an integer value. This value can only be set before the first call to the AddSmallPicture function or after calling DeleteSmallPictures. If this value is set to 0, then the size of the first picture is used to set the size of small pictures.

The following line sets the height for small pictures in a ListView to 16 pixels.

```
lv_1.SmallPictureHeight = 16
```

For more information about scripting ListView controls, see “Using ListView controls” in the *Resource Guide*.

SmallPictureMaskColor

Applies to

ListView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The mask color is the color in the picture that is transparent when the picture is displayed.

Select the color to mask newly added user-defined bitmaps. In scripts, you can change the mask color before adding each picture. Each image uses the mask color that was in effect when it was added.

Usage

In a painter

- ❖ **To specify a picture mask color:**
 - Select a color from the Picture Mask Color drop-down list on the Small Picture tab page of the control’s Properties view.

To add your own colors to the color drop-down list, select Design>Custom Colors before displaying the Properties view.

In scripts

The `SmallPictureMaskColor` property takes a long (-2 to 16,777,215) that specifies the numerical value of the background color. This property is used when each bitmap is added and, therefore, can be changed between `AddSmallPicture` calls.

The `SmallPictureMaskColor` value is a combination of values for the red, green, and blue components of the color. If you do not know the long value for a particular color, choose `Design>Custom Colors` to determine the red, green, and blue values and then call the `RGB` function to specify the color in a script.

The following example sets yellow as the mask color for user-defined bitmaps in a `ListView`.

```
lv_1.SmallPictureMaskColor = RGB(255, 255, 0)
```

SmallPictureName[]

Applies to

ListView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

PocketBuilder stores `ListView` images in several indexed arrays of images. You can associate an image with a specific `ListView` item when you create a `ListView` in the painter or use the `AddItem` and `InsertItem` functions at execution time.

You identify a specific image by its index number. Because the same index number refers to both the large picture and the small picture for the item (depending on which view is selected), you will want to make sure the images for each position in the array are compatible. The type of picture used in the control is determined by the value of the control's `View` property.

Usage

In a painter

- ❖ To specify images for the Small Icon view
- 1 Select the Small Picture tab page from the ListView control’s Properties view.
 - 2 Do one of the following:
 - In the rows provided in the Picture Name field, type the complete path and name of the files containing the desired pictures.
 - Use the Browse button.
 - Select one or more pictures from the Stock Pictures list.

The order of the picture names specified here should match the picture name order used for the Large Icon view.
 - 3 Use the row numbers from this Picture Name list to specify the Picture Index for each List View Item on the Items tab page.

In scripts

The SmallPictureName property takes a string value. You cannot use the SmallPictureName property to update the image list during execution. Use the AddSmallPicture function to add small pictures to a ListView control. For example:

```
lv_1.AddSmallPicture("c:\ArtGal\bmps\celtic.bmp")
```

When you add a small picture to a ListView control, it is given the next available picture index in the ListView.

For more information about scripting ListView controls, see “Using ListView controls” in the *Resource Guide*.

SmallPictureWidth

Applies to

ListView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description	The <code>SmallPictureWidth</code> property specifies the display width of all the pictures in the Small Icon view of the <code>ListView</code> control. The size is specified in pixels. If you choose the value (Default) in the painter, or set the value to 0, PocketBuilder uses the width of the first picture in the array as the width for all the pictures. The other choices in the painter, 16 and 32, are standard pixel widths for icons.
Usage	In a painter ❖ To set the small picture width: • Select a value from the Width drop-down list on the Small Picture tab page of the control's Properties view. In scripts The <code>SmallPictureWidth</code> property takes an integer value. This value can only be set before the first call to the <code>AddSmallPicture</code> function or after calling <code>DeleteSmallPictures</code>. If this value is set to 0, then the size of the first picture is used to set the size of small pictures. The following line sets the width for small pictures in a <code>ListView</code> to 16 pixels. <pre>lv_1.SmallPictureWidth = 16</pre> For more information about scripting <code>ListView</code> controls, see "Using <code>ListView</code> controls" in the <i>Resource Guide</i>.

SmartMinimize

Applies to	Windows <table border="1"> <tbody> <tr> <td>PocketBuilder on Pocket PC</td><td>✓</td></tr> <tr> <td>PocketBuilder on Smartphone</td><td>✓</td></tr> <tr> <td>PowerBuilder</td><td>✗</td></tr> </tbody> </table>	PocketBuilder on Pocket PC	✓	PocketBuilder on Smartphone	✓	PowerBuilder	✗
PocketBuilder on Pocket PC	✓						
PocketBuilder on Smartphone	✓						
PowerBuilder	✗						
Description	The <code>SmartMinimize</code> property adds an X icon to the title bar of a window that you deploy to a Windows CE platform. When users click the X, the application is removed from the current navigational stack, but remains in memory for quicker availability and enhanced performance.						

Usage

In a painter

- ❖ **To display an X icon to a window:**
 - Select the Smart Minimize check box on the General page of the window's Properties view.

Selecting the Smart Minimize check box clears the Close check box.

In scripts

This property cannot be set in a script.

SNR

Applies to

GPSSatellitePosition objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The SNR property hold the satellite's signal to noise ratio (SNR), which is an indicator of signal strength. The range of values is 0 to 99.

Usage

In scripts

The SNR property takes an integer value:

```
integer li_SNR
GPSSatellitePosition myGPS_SP

...
li_SNR = myGPS_SP.SNR
```

Sorted

Applies to

DropDownListBox, ListBox controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description Items in a list box can be sorted alphabetically. If the Sorted property is enabled, the items in the list box are sorted in ascending order. If this property is not enabled, the items in the list box are not sorted and are displayed in the order they were added.

Usage

In a painter

❖ **To enable automatic sorting:**

- Select the Sorted check box on the General page of the control's Properties view.

In scripts

The Sorted property takes a boolean value. The following line specifies that items in the ListBox lb_1 are sorted.

```
lb_1.Sorted = TRUE
```

SortType

Applies to

ListView, TreeView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The SortType property specifies how items should be sorted. Items can be sorted alphabetically based on the item names or according to user-defined rules. If you specify a user defined or unsorted sort type, define your sort criteria in the Sort event of the control.

In TreeView controls, each parent item's children form their own sorted list. For more information, see the *Resource Guide*.

Usage

In a painter

❖ **To specify how items should be sorted:**

- Select the desired sort type from the Sort drop-down list on the General page of the control's Properties view.

In scripts

The SortType property takes a value of the grSortType enumerated datatype. The following line specifies Unsorted for the items in a ListView.

```
lv_1.SortType = Unsorted!
```

SpacesVisible

Applies to

RichTextEdit controls

PocketBuilder	✗
PowerBuilder	✓

Description

When SpacesVisible is enabled, spaces in the text are marked by a dot in the RichTextEdit control. If this property is not enabled, spaces will appear as blanks only.

Spacing

Applies to

Graph controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

Spacing defines the gap (space) between data markers in a graph as a percent of the width of the markers. For example, in a bar graph, 100 is the width of one bar; 50 is half a bar.

Usage

In a painter

- ❖ **To change the spacing of data markers:**
 - Select the desired spacing percentage using the Spacing slider on the General page of the graph control’s Properties view.

In scripts

The Spacing property takes an integer value. The following line specifies 120 percent of the bar width as the space between bars in a bar Graph.

```
gr_1.Spacing = 120
```

Spin

Applies to

EditMask controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The Spin property specifies whether the control is defined as a spin control that contains up and down arrows that the user can click to cycle through fixed values.

Usage

In a painter

❖ To make an EditMask into a spin control:

- Select the Spin Control check box on the Mask tab page of the control's Properties view.

In scripts

The Spin property takes a boolean value.

The following line specifies that the user can cycle through values in an EditMask.

```
em_1.Spin = TRUE
```

StartTime

Applies to

CallLogEntry objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Indicates when the call started.

Usage

In scripts

The StartTime property takes a DateTime value. The following example writes the start and end times of a call log entry to a multiline edit box:

```
// Integer idx passed into function
CallLog l_log
```

```
CallLogEntry l_logentry

l_log = CREATE CallLog
l_logentry = l_log.getEntry (idx)
mle_1.text = "Start time: " + string(l_entry.StartTime)
mle_1.text += "End time: " + string(l_entry.EndTime)
```

StatePictureHeight

Applies to ListView, TreeView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description The StatePictureHeight property specifies the display height of all the state pictures. The size is specified in pixels.

If you choose the value (Default) in the painter, or set the value to 0, PocketBuilder uses the height of the first picture in the array as the height for all the pictures. The other choices in the painter, 16 and 32, are standard pixel heights for icons.

Usage In a painter

- ❖ To set the state picture height:
 - Select a value from the Height drop-down list on the State tab page of the control’s Properties view.

In scripts

The StatePictureHeight property takes an integer value. This value can only be set before the first call to the AddStatePicture function or after calling DeleteStatePictures. If this value is set to 0, then the size of the first picture is used to set the size of state pictures.

The following line sets the height for state pictures in a ListView to 16 pixels.

```
lv_1.StatePictureHeight = 16
```

For more information about scripting ListView controls, see “Using ListView controls” in the *Resource Guide*.

StatePictureMaskColor

Applies to

ListView, TreeView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The mask color is the color in the picture that is transparent when the picture is displayed.

Select the color to mask newly added user-defined bitmaps. You can change the mask color before adding each picture. Each image uses the mask color that was in effect when it was added.

Usage

In a painter

❖ To specify a picture mask color:

- Select a color from the Picture Mask Color drop-down list on the State Picture tab page of the control's Properties view.

To add your own colors to the color drop-down list, select Design>Custom Colors before displaying the Properties view.

In scripts

The StatePictureMaskColor property takes a long (-2 to 16,777,215) that specifies the numerical value of the background color. This property is used when each bitmap is added and, therefore, can be changed between AddStatePicture calls.

The StatePictureMaskColor value is a combination of values for the red, green, and blue components of the color. If you do not know the long value for a particular color, choose Design>Custom Colors to determine the red, green, and blue values and then call the RGB function to specify the color in a script.

The following example sets yellow as the mask color for user-defined bitmaps in a ListView.

```
lv_1.StatePictureMaskColor = RGB(255, 255, 0)
```

StatePictureName[]

Applies to

ListView, TreeView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

PocketBuilder stores ListView images in several indexed arrays of images. State pictures are displayed to the left of ListView items and their pictures, if they have them.

You can associate a state image with a ListView control only with scripts.

You identify a specific image by its index number.

Usage

In a painter

❖ To specify State images:

- 1 Select the State tab page from the ListView control’s Properties view.
- 2 Do one of the following:
 - In the rows provided in the Picture Name field, type the complete path and name of the files containing the desired pictures.
 - Use the Browse button.
 - Select one or more pictures from the Stock Pictures list.
- 3 Use the row numbers from this Picture Name list as the index number when setting the State picture index in scripts.

In scripts

The StatePictureName property takes a string value. You cannot use the StatePictureName property to update the image list during execution. Use the AddStatePicture function to add State pictures to a ListView control. For example:

```
integer index
index = lv_1.AddStatePicture("c:\ArtGal\ico\star.ico")
lv_1.StatePictureIndex = index
```

For more information about scripting ListView controls, see “Using ListView controls” in the *Resource Guide*.

StatePictureWidth

Applies to

ListView, TreeView controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The StatePictureWidth property specifies the display width of all the state pictures. The size is specified in pixels.

If you choose the value (Default) in the painter, or set the value to 0, PocketBuilder uses the width of the first picture in the array as the width for all the pictures. The other choices in the painter, 16 and 32, are standard pixel widths for icons.

Usage

In a painter

❖ To set the state picture width:

- Select a value from the Width drop-down list on the State tab page of the control's Properties view.

In scripts

The StatePictureWidth property takes an integer value. This value can only be set before the first call to the AddStatePicture function or after calling DeleteStatePictures. If this value is set to 0, then the size of the first picture is used to set the size of state pictures.

The following line sets the width for state pictures in a ListView to 16 pixels.

```
lv_1.StatePictureWidth = 16
```

For more information about scripting ListView controls, see “Using ListView controls” in the *Resource Guide*.

Status

Applies to

SMSMessage objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

After the SMSSession Send function is called, the Status property of the SMSMessage object indicates the current status of the message. You can get the value of this property using the GetMessageStatus function.

The Status property takes a value of the SMSMsgStatus enumerated variable. Values are shown in the following table. When the status indicates that there is a temporary failure, the problem causing the error is not expected to persist.

Enumerated value	Description
Message_Status_Unknown!	Message status is unknown.
Message_Status_ForwardedToSME!	Message successfully forwarded to destination.
Message_Status_ReceivedBySME!	Message successfully received by destination.
Message_Status_ReplacedBySC!	Message successfully replaced in the SMSC.
Message_Status_Reserved_Completed!	Message successfully reserved.
Message_Status_SCSpecific_Completed!	Message is specific to an SMSC and has been successfully delivered.
Message_Status_Congestion!	Temporary failure due to network congestion.
Message_Status_QualityUnavail_Temp!	Temporary failure because quality is not available.
Message_Status_SCSpecific_TmpError!	Temporary failure due to an SMSC-specific error. System still trying to deliver.
Message_Status_Reserved_TmpError!	Temporary failure because the message was reserved.
Message_Status_SMEBusy!	Temporary failure because the destination is busy.
Message_Status_SMEError!	Temporary failure due to an error at destination.
Message_Status_SMENotResponding!	Temporary failure because the destination is not responding.
Message_Status_SVCRejected!	Temporary failure because service was rejected.
Message_Status_Congestion_Trying!	Network congestion. System still trying to deliver.
Message_Status_QualityUnavail_Trying!	Quality is not available. System still trying to deliver.
Message_Status_Reserved_Trying!	Reserved. System still trying to deliver.
Message_Status_SCSpecific_Trying!	SMSC specific. System still trying to deliver.
Message_Status_SMEBusy_Trying!	Destination is busy. System still trying to deliver.
Message_Status_SMEError_Trying!	Destination error. System still trying to deliver.
Message_Status_SMENotResponding_Trying!	Destination not responding. System still trying to deliver.
Message_Status_SVCRejected_Trying!	Service rejected. System still trying to deliver.
Message_Status_ConnectionRejected!	Failure because destination rejected the connection.
Message_Status_DeletedByOrigSME!	Failure because the message was deleted by the originator.
Message_Status_DeletedBySC!	Failure because the message was deleted by Short Message Service Center (SMSC).
Message_Status_IncompatibleDest!	Failure because the destination is incompatible.
Message_Status_NoInternetWorking!	Failure because the internet is not available.
Message_Status_NoLongerExists!	Failure because the message no longer exists.

Enumerated value	Description
Message_Status_NotObtainable!	Failure because the destination cannot be attained.
Message_Status_QualityUnavail!	Failure because quality is not available.
Message_Status_RemoteProcError!	Failure due to a remote processor error.
Message_Status_Reserved_Error!	Failure due to a reserved error.
Message_Status_SCSpecific_Error!	Failure due to an SMSC-specific error.
Message_Status_VPExpired!	Failure because the validity period of the message expired.

Usage**In scripts**

The Status property takes a value of the SMSMsgStatus enumerated variable. This example uses the GetMessageStatus function to obtain the value and write it to a single line edit box:

```
SMSMsgStatus msgStat
integer li_ret

g_smsSess.Send(g_smsMsg, g_smsAddr)
sleep(5)
li_ret = g_smsSess.GetMessageStatus(g_smsMsg)
msgStat = g_smsMsg.Status
sle_status.text = "Message status: " + String(msgStat)
```

StdHeight

Applies to**HScrollBar**

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

If you enable the StdHeight property, the HScrollBar displays with the standard height for your system.

Usage**In a painter**❖ **To enable standard height:**

- Select the StdHeight check box on the General page of the control's Properties view.

In scripts

The StdHeight property takes a boolean value. At runtime, as long as StdHeight is TRUE, setting the Height property has no effect. If you set the StdHeight property to TRUE, the scroll bar displays with the standard height. If you set the StdHeight property to FALSE, the scroll bar displays with the height specified in the Height property.

The following line specifies that height for an HScrollBar, instead of being standard, is set to the height specified in the Height property.

```
hsb_1.StdHeight = FALSE
```

StdWidth

Applies to

VScrollBar controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

If you enable the StdWidth option, the VScrollBar displays with the standard width for your system.

Usage

In a painter

- ❖ **To enable standard width:**
 - Select the StdWidth check box on the General page of the control's Properties view.

In scripts

The StdWidth property takes a boolean value. At runtime, as long as StdWidth is TRUE, setting the Width property has no effect. If you set the StdWidth property to TRUE, the scroll bar displays with the standard width. If you set the StdWidth property to FALSE, the scroll bar displays with the width specified in the Width property.

The following line specifies that width for a VScrollBar, instead of being standard, is set to the width specified in the Width property.

```
vsb_1.StdWidth = FALSE
```

TabStop[]

Applies to

MultiLineEdit, ListBox controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The TabStop property array allows you to specify a repeating tab stop or tab stops at arbitrary positions. The tab stops are indicated by character positions.

If you specify one value, the tab stops are equally spaced using that value. If more than one tab stop is specified, tab stops are located in the character positions entered. The default is tab stops every 8 character positions.

Usage

In a painter

❖ To specify tab stops:

- Enter the character positions for each tab stop desired in the TabStop field on the General page of the control's Properties view.

In scripts

The TabStop[] property is a signed integer array containing the positions of the tab stops. The tab stops are in character positions.

The following lines define two tab stops at character positions 5 and 15.

```
lb_1.tabstop[1] = 5
lb_1.tabstop[2] = 15
```

TabOrder

Applies to

Visible controls within a window.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

TabOrder specifies the order in which the control will receive focus when the user tabs among controls within a window. Setting the TabOrder for a control to 0 means that the control cannot be tabbed to.

Usage

In a painter

- ❖ To set tab order for controls within a window:
 - 1 Select Format>Tab Order from the menu bar.

Numbers indicating the tab order for each visible control are displayed in red on the window.
 - 2 Select the number you want to change and type in a new number between 0 and 9999.

The actual value of the number does not matter; only the relative values among controls.
 - 3 Select Format>Tab Order from the menu bar again to save the tab order.

In scripts

The TabOrder property takes an integer value between 0 and 9999. The value of 0 removes the control from the tab order.

The following example sets the tab order for three controls. The EditMask control is tabbed to after the ListView control and before the CommandButton.

```
lv_1.TabOrder = 10
em_1.TabOrder = 15
cb_1.TabOrder = 20
```

TabPosition

Applies to

Tab controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

Tabs can appear on any side of the Tab control (top, bottom, left, right) or on opposite sides.

When you select two sides, for example, top and bottom, the selected tab divides the tabs so that tabs before it appear on one side and tabs after it appear on the opposite side. The selected tab itself appears on the first side of the pair.

Usage

In a painter❖ **To set the position of tab pages in a tab control:**

- Select the desired type of position from the TabPosition drop-down list on the General page of the tab control's Properties view.

In scripts

The datatype of the TabPosition property is the TabPosition enumerated datatype.

The following example positions tabs on the top and bottom of the Tab control. Tabs before the selected tab and the selected tab itself are on top. Tabs after the selected tab are on the bottom.

```
tab_1.TabPosition = TabsOnTopAndBottom!
```

TabTextColor

Applies to

TabPage objects and UserObjects when they are tab pages

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

The TabTextColor property allows you to select the color for the tab's text.

Usage

In a painter

To add your own colors to the color drop-down list, select Design>Custom Colors before displaying the Properties view.

❖ **To change the tab text color:**

- 1 Select the desired tab page on the tab control.
- 2 Select the TabPage tab in the Properties view.
- 3 Select the desired color in the TabTextColor drop-down list.

In scripts

The TabTextColor property takes a long value (-2 to 16,777,215) that specifies the numerical value of a color. The TabTextColor value is a combination of values for the red, green, and blue components of the color.

If you do not know the long value for the color, choose Design>Custom Colors to determine the red, green, and blue values and then call the RGB function to specify the color in a script.

The following example sets yellow as the text color for a tab.

```
tab_1.tabpage_2.TabTextColor = RGB(255, 255, 0)
```

TabBackColor

Applies to

TabPage objects and UserObjects when they are tab pages

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

The TabBackColor property allows you to select the color of the tab on the tab page.

Windows XP

This property is not supported on Windows XP.

Usage

In a painter

To add your own colors to the color drop-down list, select Design>Custom Colors before displaying the Properties view.

❖ **To set the background color for the tab**

- 1 Select the desired tab page on the tab control.
- 2 Select the TabPage tab in the Properties view.
- 3 Select the desired color in the TabBackColor drop-down list.

You can set the color of the body of the tab page on its General tab.

In scripts

The TabBackColor property takes a long value (-2 to 16,777,215) that specifies the numerical value of a color. The TabBackColor value is a combination of values for the red, green, and blue components of the color.

If you do not know the long value for the color, choose Design>Custom Colors to determine the red, green, and blue values and then call the RGB function to specify the color in a script.

The following example sets blue as the background color for a tab.

```
tab_1.tabpage_2.TabBackColor = RGB(0, 0, 255)
```

TabsVisible

Applies to

RichTextEdit controls

PocketBuilder	✗
PowerBuilder	✓

Description

When the TabsVisible property is enabled, a text symbol will appear for tabs in text in the RichTextEdit control. This property can also be enabled and disabled by the user at runtime from the toolbar and from the Properties item of the pop-up menu, if the PopMenu property is enabled.

Tag

Applies to

All controls, user objects, and menus

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The Tag property can hold any text you want to associate with the control. It is up to you how you use that text.

Usage

In a painter

❖ To specify a tag for a control:

- Enter the desired text in the Tag field on the General page of the object's Properties view.

In scripts

The tag property takes a string value.

The following line uses the object's Tag property to set MicroHelp in a PowerBuilder application's MDI frame. (This code could be in a GetFocus event or, for a Menu object, the Selected event.):

```
w_frame.SetMicroHelp(This.Tag)
```

Tap_And_Hold_Indicator

Applies to Windows and all draggable controls. This property does not apply to menu or drawing objects.

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✗

Description When the Tap_And_Hold_Indicator property is enabled, red or blue dots in a circular animation display to confirm that the user has performed a tap and hold action with the stylus on the control. This action on a Pocket PC simulates a right mouse-click. The property is enabled by default to conform with typical Pocket PC behavior.

Usage **In a painter**
To disable the property, clear the Tap and Hold Indicator check box on the Other property page of the window or control.

In a script
The Tap_And_Hold_Indicator property takes a boolean value. The following example sets the Tap_And_Hold_Indicator property of a ListView item to false, which prevents the animation from displaying when the user holds the stylus on the item:

```
lvitem_1.Tap_And_Hold_Indicator = false
```

Text

Applies to

Menus and controls that display text

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

SMSMessage objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

For menu objects and text controls The Text property specifies the text displayed in the menu object or control.

If a Menu item has a shortcut key (for example, F1 or Alt+a), Text includes the shortcut key. If the Text property of a Menu item is a single dash (-), the item displays as a separator (a horizontal line the width of the menu), and all other properties for the item are ignored.

For SMSMessage objects The Text property specifies the text data portion of the message. This value can be null.

Usage

In a painter (menu objects and text controls)

❖ To specify text to be displayed:

- Enter the desired text in the Text field on the General page of the objects' Properties view.

In scripts (menu objects and text controls)

The Text property takes a string value. The following line specifies that the text of a check box is Male.

```
cb_1.Text = "Male"
```

In scripts (SMSMessage objects)

The following line specifies that the text of an SMSMessage object is Hello World.

```
mySmsMsg.Text = "Hello World"
```

TextCase

Applies to

EditMask, MultiLineEdit, SingleLineEdit controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The TextCase property lets you constrain the case of text entered by the user. The text can be displayed as the user types it, as all lowercase, or as all uppercase.

Usage

In a painter

❖ **To select the case used to display text entered by users:**

- Select the desired text case from the TextCase drop-down list on the General tag page in the control's Properties view.

In scripts

The TextCase property takes a value of the TextCase enumerated datatype. The following line sets the case for a MultiLineEdit to all uppercase.

```
mle_1.TextCase = Upper!
```

TextColor

Applies to

Controls and objects that display text

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The TextColor property specifies the color to be used for text in the control.

Usage

In a painter

❖ **To set the text color for controls:**

- Select the desired color from the TextColor drop-down list on the Font tab page of the Properties view.

❖ **To set the text color for graph objects:**

- Select the desired color from the TextColor drop-down list on the General page of the Properties view.

❖ **To set the text color for text objects within graphs:**

- 1 Select the Text tab page from the graph's Properties view.
- 2 Select the desired text object from TextObject drop-down list.
- 3 Select a color from the TextColor drop-down list.

In scripts

The TextColor property is a long indicating the color to be used for the background for an object. If you do not know the long value for the color, choose Design>Custom Colors to determine the red, green, and blue values and then call the RGB function to specify the color in a script.

In graphs, the TextColor property is a property of the graph object as well as of grDistAttr objects within the graph. For example, the following line sets text color for all the text objects in the Series Axis:

```
gr_1.Series.DispAttr.TextColor = RGB(0,0,255)
```

TextSize

Applies to

Controls that can display text

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The TextSize property specifies the point size of the text in the control.

Usage

In a painter❖ **To set the size of all the text in a control:**

- Display the Font tab page of the control's Properties view and select the desired point size from the Size drop-down list, or select the control and then set the point size using the Font Size list box on the StyleBar.

❖ **To set the size of a text object in a graph control:**

- 1 Display the Text tab page of the graph control's Properties view and select the desired text object from the Text Object list.

- 2 Select the desired point size from the TextSize list.

In scripts

The TextSize property takes an integer value that indicates the point size. The following example sets the point size of a static text control:

```
st_1.TextSize = 12
```

This example sets the point size of the label of the Value axis of a graph control.

```
gr_1.Values.LabelDispAttr.TextSize = 12
```

ThreeState

Applies to

CheckBox controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The ThreeState property specifies whether or not the control can have three states. Typically, the state toggles between *selected* and *not selected*. For check boxes, if the ThreeState property has been enabled, the state of the control also toggles to a *third state*. A grayed out mark is displayed for the *third state*.

Usage

In a painter

- ❖ **To allow the check box to have three states:**
 - Select the ThreeState check box on the General page of the control's Properties view.

In scripts

The ThreeState property takes a boolean value. The following lines specify that a CheckBox can have three states and that it starts out in the third state.

```
cbx_1.ThreeState = TRUE  
cbx_1.ThirdState = TRUE
```

ThirdState

Applies to

CheckBox controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The ThirdState property specifies whether the CheckBox is in the third state (neither selected nor unselected).

For a check box to be in the third state, the ThreeState property must also be enabled.

Usage

In a painter

- ❖ **To specify that a check box is in the third state:**
 - Check both the ThreeState and the ThirdState check boxes on the General page of the control's Properties view.

In scripts

The ThirdState property takes a boolean value. The following lines specify that a CheckBox can have three states and that it starts out in the third state.

```
cbx_1.ThreeState = TRUE
cbx_1.ThirdState = TRUE
```

Title

Applies to

DataWindow controls, Graph controls, windows

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The Title property specifies the title text of the control or window. In a window or DataWindow control, this value is displayed only if the TitleBar property is also enabled.

Usage

In a painter

- ❖ **To specify title text:**
 - Type the title text in the Title field and select the TitleBar check box on the General page of the control's Properties view.

In scripts

The Title property takes a string value. The following lines set a title for a DataWindow control dw_1.

```
dw_1.TitleBar = TRUE
dw_1.Title = "Monthly Report"
```

TitleBar

Applies to

DataWindow controls, Windows

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The TitleBar property specifies whether the DataWindow control or window displays a title bar. The user can move a window or DataWindow control only if it has a title bar.

If the window type is a main or MDI frame window with or without MicroHelp, the TitleBar property is always enabled. When the title bar is enabled, you can choose whether to include the control menu and the maximize and minimize boxes in the title bar.

Usage

In a painter

- ❖ **To display a title bar:**
 - Select the TitleBar check box on the General page of the DataWindow control's or window's Properties view.

In scripts

For DataWindow controls, the TitleBar property can be modified in a script. It cannot be modified for Windows.

The TitleBar property takes a boolean value. The following line specifies that a title bar will appear in a DataWindow control dw_1.

```
dw_1.TitleBar = TRUE
```

TodayDisplayApp

Applies to

Application object

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Specifies an application to control the display of a custom PocketBuilder Today item. You must use the full device path to the display application.

Typically, you would use the application where you set the custom Today item as the item's display application, but this is not required. The display application should not have a visual user interface. However, it can be used to define updated text and formatting for the custom Today item.

A display application is especially useful if you want to change the custom item display based on the time elapsed, or on a counter for events, such as the number of updates to a database. Adding a display application adds to the memory burden of a device, since the PocketBuilder VM remains loaded in memory at all times (or until the custom Today item is disabled or removed).

Usage

In a painter

❖ To select a display application for a Today item:

- Type the name of the display application in the Display App text box on the Today Item page of the Application object's Properties view.

In scripts

The TodayDisplayApp property takes a string. The following example sets the display application for the custom Today item associated with the SyncDisplay application:

```
SyncDisplay.TodayDisplayApp="\Program Files\timer.exe"
```

TodayDisplayText

Applies to

Application object

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Specifies the text for a custom PocketBuilder Today item.

Usage

In a painter

- ❖ **To select a display text for a Today item:**
 - Type the name of the display text in the Display Text text box on the Today Item page of the Application object’s Properties view.

In scripts

The TodayDisplayText property takes a string. The following example sets the display text for the custom Today item associated with the SyncDisplay application:

```
SyncDisplay.TodayDisplayText = "Sync Update Count is " &  
+ string(counter)
```

TodayOrder

Applies to

Application object

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Specifies the position in the Today screen of a custom PocketBuilder Today item.

Typically the top item in the Today screen is the date, followed by owner information, calendar appointments, the mail inbox, and tasks. The date does not have an order number in the device registry, so if it is displayed, it is always at the top of the Today screen. Otherwise, the first item in the Today screen has the order number 0 and the Today screen can accommodate a maximum of 12 items.

When you deploy a custom item with a `TodayOrder` value of 0, the previous item in the Today screen with a value of 0 is moved to the bottom of the Today screen.

Usage

In a painter

- ❖ **To select a position for a custom Today item:**
 - Type a number from 0 to 11 in the Order text box on the Today Item page of the Application object's Properties view.

In scripts

The `TodayOrder` property takes an integer. The following example sets the custom Today item associated with the `SyncDisplay` application as the sixth item in the Today screen:

```
SyncDisplay.TodayOrder = 5
```

TodayRunApp

Applies to

Application object

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Specifies the application that is launched when the user taps the custom PocketBuilder Today item. You must use the full device path to the run application.

If you select a run application, the icon assigned to the run application is displayed next to the custom Today item on the device.

Usage

In a painter

- ❖ **To select an application to launch when the user taps the custom Today item:**
 - Type the name of the application in the Run App text box on the Today Item page of the Application object's Properties view.

In scripts

The TodayRunApp property takes a string. The following example sets the application to launch when a user taps the custom Today item associated with the SyncDisplay application:

```
SyncDisplay.TodayRunApp = "\\Program Files\\sync_run.exe"
```

ToolBar

Applies to

RichTextEdit controls

PocketBuilder	✗
PowerBuilder	✓

Description

When the ToolBar property is enabled, a toolbar for formatting text displays above the editing area of the RichTextEdit control.

ToolbarAlignment

Applies to

Windows

PocketBuilder	✗
PowerBuilder	✓

Description

In an MDI frame window, Alignment specifies where the toolbar displays.

ToolbarHeight

Applies to

Windows

PocketBuilder	✗
PowerBuilder	✓

Description

For MDI frame windows, the ToolbarHeight property specifies the height of the toolbar when it is a floating toolbar.

ToolbarVisible

Applies to

Windows

PocketBuilder	✗
PowerBuilder	✓

Description

For MDI frame windows, ToolbarVisible specifies whether the toolbar is displayed.

ToolbarWidth

Applies to

Windows

PocketBuilder	✗
PowerBuilder	✓

Description

For MDI frame windows, the ToolbarWidth property specifies the width of the toolbar when it is a floating toolbar.

ToolbarX

Applies to

Windows

PocketBuilder	✗
PowerBuilder	✓

Description

The ToolbarX property specifies the X coordinate in PowerBuilder units of the toolbar when it is a floating toolbar. The X coordinate is the distance from the left edge of the window or screen.

ToolbarY

Applies to

Windows

PocketBuilder	✗
PowerBuilder	✓

Description

The ToolbarY property specifies the Y coordinate in PowerBuilder units of the toolbar when it is a floating toolbar. The Y coordinate is the distance from the top of the window or screen.

TopMargin

Applies to

RichTextEdit controls

PocketBuilder	✗
PowerBuilder	✓

Description

The TopMargin property specifies the size in inches of the top margin on the printed page.

Underline

Applies to

Controls that display text

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

Underline is a property of text in a control.

Usage

In a painter

- ❖ **To underline all text items in a control:**
 - Select the Underline check box on the Font tab page of the control’s property page, or select the control and then click the U button on the StyleBar.

❖ **To underline a text object in a graph control:**

- 1 Display the Text tab page of the graph control's Properties view and select the desired text object from the Text Object list.
- 2 Select the Underline check box on the Text tab page.

In scripts

The Underline property takes a boolean value. The following example underlines the text in a StaticText control:

```
st_1.Underline = TRUE
```

This example underlines the label of the Value axis of a graph control.

```
gr_1.Values.LabelDispAttr.Underline = TRUE
```

UndoDepth

Applies to

RichTextEdit controls

PocketBuilder	✗
PowerBuilder	✓

Description

The UndoDepth property specifies the maximum number of editing changes that the Undo function will undo. Each time you call Undo, one more editing change is restored. The CanUndo function returns FALSE when there are no more changes to undo.

UnitsPerColumn

Applies to

Windows and user objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

UnitsPerColumn specifies the number of PowerBuilder units you want to scroll right or left when the user clicks the left or right arrow in the horizontal scroll bar in a window or user object. The default is 0 (1/100 of the width of the window or user object). PocketBuilder controls horizontal scrolling automatically when Units Per Column is 0.

PocketBuilder multiplies Units Per Column by Columns Per Page to determine the number of PowerBuilder units to scroll the window horizontally when the user clicks in the scroll bar.

For information on calculating ColumnsPerPage and UnitsPerColumn, see “Scrolling in windows and user objects” on page 569.

Usage note

To control the vertical scroll bar in a window or user object, use the UnitsPerLine and LinesPerPage properties.

Usage

In a painter

- ❖ **To specify the UnitsPerColumn property:**
 - Enter the desired number of PowerBuilder units in the UnitsPerColumn field on the Scroll tab page of the window’s Properties view.

In scripts

The UnitsPerColumn property takes an integer value.

The following statement sets Units Per Column to 12, which is appropriate for a content width of 1650.

```
This.UnitsPerColumn = 12
```

UnitsPerLine

Applies to Windows, user objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

UnitsPerLine specifies the number of PowerBuilder units you want to scroll up or down when the user clicks the up or down arrow in the vertical scroll bar in a window or user object. The default is 0 (1/100 of the window or user object height). When UnitsPerLine is 0, PocketBuilder controls vertical scrolling automatically.

PocketBuilder multiplies UnitsPerPage by UnitsPerLine to determine the number of PocketBuilder units to scroll the window or user object vertically when the user clicks in the scroll bar.

For information on calculating LinesPerPage and UnitsPerLine, see “Scrolling in windows and user objects” on page 569.

Usage note

To control horizontal scrolling in a window or user object, use the UnitsPerColumn and ColumnsPerPage properties.

Usage**In a painter**❖ **To set the UnitsPerLine property:**

- Enter the desired number of PowerBuilder units in the UnitsPerLine field on the Scroll tab page of the window’s Properties view

In scripts

The UnitsPerLine property takes an integer value.

The following statement sets UnitsPerLine to 17, which is appropriate for a content length of 2400.

```
lb_1.UnitsPerLine = 17
```

UseCodeTable

Applies to

EditMask controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description When an EditMask control has been defined as a spin control (that is, a control with up and down arrows the user clicks to cycle through predefined values), a code table can be used to validate data.

The UseCodeTable property specifies whether the control uses a code table to validate data.

- Usage**
- In a painter**
- ❖ **To specify use of a code table for an EditMask control:**
 - 1 Select the Spin Control and Code Table check boxes on the Mask tab page of the control's Properties view.

An area appears on the lower half of the tab page where you can enter values for the code table.
 - 2 Specify Display Values and their corresponding Data Values.

Use the Insert button to insert items within this list.

In scripts

The UseCodeTable property takes a boolean value. This example specifies that the EditMask control should use its code table to validate data.

```
em_1.UseCodeTable = TRUE
```

You can specify the contents of the code table in scripts by using the DisplayData property. Enter the Display values and their corresponding Data values as a text string, with the Display and Data pairs separated by tabs and the pairs separated by slashes. For example:

```
em_1.DisplayData = "Black 1/White 2/Red 3"
```

ValidityPeriod

Applies to SMSMessage objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description Defines the time period for which the SMS message is valid, starting with the time when the SMSC receives the message. This value can be null.

Usage

In scripts

The `ValidityPeriod` property takes a `DateTime` value. This example gets the value of the property from a single line edit box:

```
datetime dt
dt = DateTime(sle_validity.text)
g_smsMsg.ValidityPeriod = dt
```

VDOP

Applies to

GPSSatellitesInView objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The Vertical Dilution of Precision (VDOP) property indicates the level of confidence in the accuracy of measurements related to the vertical position of the satellite, based on current satellite geometry. A lower value indicates greater confidence.

Usage

In scripts

The `VDOP` property takes a real value. The following example assumes that a `VDOP` value of 3 or less is required:

```
mySatsInView = MyGPS.GetSatellitesInView()
if (mySatsInView.VDOP <= 3) then
    // continue
else
    // take another reading
end if
```

Vendor

Applies to

SerialGPS objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

The Vendor property specifies the vendor of the GPS device.

Usage

In a painter

Type the value in the Vendor box on the General page of the Object’s Properties view.

In scripts

The Vendor property takes a string value. This example sets the Vendor property to Pharos:

```
string myVendor
SerialGps myGPS
Integer rc

MyGPS = CREATE SerialGPS
rc = myGPS.Open()
...
MyGPS.Vendor = "Pharos"
```

Version

Applies to

POOM objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

This is a read-only property that indicates the version of Microsoft Pocket Outlook.

Usage**In scripts**

The Version property is a string. The following example gets the version for a POOM object and writes it to a single line edit box:

```
sle_1.text = po_1.version
```

View

Applies to**ListView controls**

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

A ListView has four ways to display its items:

- **Large icon view** Items are arranged from left to right and the user can move items around when drag and drop is enabled. Each item's picture is taken from the large picture list and the item label is below the picture.
- **Small icon view** Same as large icon view except each item's picture is taken from the small picture list and the item label is to the right of the picture.
- **List view** Items are arranged from top to bottom. Each item's picture is taken from the small picture array.
- **Report view** Items are arranged from top to bottom with one or more columns of information for each item. You must write a script to set up the columns.

Usage**In a painter**❖ **To select the view type:**

- Select the desired view type from the View drop-down list on the General page of the control's Properties view.

In scripts

The View property takes a value of the ListViewView enumerated datatype.

The following line specifies that small pictures appear for the items in the ListBox.

```
lv_1.View = ListViewSmallIcon!
```

ViewStyle

Applies to

Signature controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Specifies the type of user input expected. Values are:

- SIGViewStyleWriting! – Words in user input are converted to text
- SIGViewStyleTyping! – User inputs data using the SIP
- SIGViewStyleDrawing! – No attempt is made to convert user input into text

Usage

In a painter

❖ **To select a view style for the control:**

- Select the value you want from the View Style drop-down list on the General page of the control's Properties view.

In scripts

The ViewStyle property takes an enumerated datatype. The following example sets the view style for the control to allow drawing only:

```
sig_mine.ViewStyle = SigViewStyleDrawing!
```

Visible

Applies to

Controls, windows, user objects, menus

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The Visible property specifies whether the object, window object, or Menu object is visible.

Usage**In a painter**❖ **To set the Visible property:**

- Select the Visible check box on the General page of the object's Properties view.

In scripts

The Visible property takes a boolean value. The following line specifies that MultiLineEdit mle_1 is visible.

```
mle_1.Visible = TRUE
```

You can use the Show and Hide functions to change the visibility of an object.

Usage note

You cannot use the Visible property or the Show or Hide functions to show or hide an MDI sheet or a drop-down or cascading menu or any menu that has an MDI frame window as its parent window.

VScrollBar

Applies to

DataWindow, MultiTextEdit, DropDownListBox, ListBox controls, windows, user objects

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✗
PowerBuilder	✓

Description

When the VScroll Bar property is enabled, PocketBuilder adds a vertical scroll bar to the right of a window or other control when the contents of the object are outside the borders.

Usage**In a painter**❖ **To allow display of a vertical scroll bar:**

- Select the VScrollBar check box on the General or Scroll tab page of the object's Properties view.

In scripts

The VScrollBar property is a boolean value.

This example displays a vertical scroll bar in a DataWindow control.

```
dw_1.VScrollBar = TRUE
```

VTextAlign

Applies to

PictureButton controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The HTextAlign property specifies how the text label for the PictureButton control is aligned in relation to the picture.

Usage

In a painter

- ❖ **To set the vertical alignment of text:**
 - Select the desired alignment from the VTextAlign drop-down list on the General tab of the control's Properties view.

In scripts

The VTextAlign property takes a value of the VTextAlign enumerated datatype.

The following example specifies bottom alignment for text in a PictureButton.

```
pb_1.VTextAlign = Bottom!
```

Weight

Applies to

Controls that can display text

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The Weight property specifies the stroke weight of the text in the control.

Usage**In a painter**❖ **To set the stroke weight of all text in a control:**

- Display the Font tab page of the control's Properties view and select the Bold check box, or select the control and then click the B button on the StyleBar.

❖ **To set the stroke weight of a text object in a graph control:**

- 1 Display the Text tab page of the graph control's Properties view and select the desired text object from the Text Object list box.
- 2 Select the Bold check box.

In scripts

The Weight property takes an integer value. 400 indicates a normal weight and 700 indicates a bold weight. The following example sets the text labels of the tab pages of a tab control to bold:

```
tab_1.Weight = 700
```

Width

Applies to

Visible controls, windows

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The Width property specifies the width of a control or window in PowerBuilder units.

Usage**In a painter**❖ **To set the width of a control or window**

- Enter the desired width in the Width edit box on the Other tab page of the object's Properties view, or select the control or window and resize it with your cursor.

In scripts

The Width property takes an integer value specifying the width of an object in PowerBuilder units. The following example sets width of a DataWindow control dw_1.

```
dw_1.Width = 750
```

It is illegal to resize a minimized or maximized sheet or frame. Changing the Width or Height property for a minimized or maximized window is not supported.

WindowState

Applies to

Windows

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The WindowState property specifies how the window is first displayed. The state can be:

- **Maximized** Enlarge the window to its maximum size.
- **Minimized** Shrink the window to an icon.
- **Normal (Default)** Display the window as it is defined in the painter.

Usage

In a painter

- ❖ **To set the window state:**
 - Select the desired state from the WindowState drop-down list on the Position tab page of the window’s Properties view.

In scripts

You cannot specify the initial state of the window before it has been opened. You can change its display state afterwards while the window is open.

The WindowState property takes a value of the WindowState enumerated datatype. The following line sets the Maximized state for the current window.

```
This.WindowState = Maximized!
```

WindowType

Applies to

Windows

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The value of this property specifies the type of window.

Table 3-1: Window types

Main	A standalone overlapped window that can be independent of all other windows.
Response	A window that displays to obtain information from the user and cannot lose focus or be closed until the user responds.

Usage

In a painter

❖ To specify the window type:

- Select the desired type from the WindowType drop-down list on the General page of the window's Properties view.

In scripts

You cannot change a window's WindowType property dynamically at runtime.

WordWrap

Applies to

RichTextEdit controls

PocketBuilder	✗
PowerBuilder	✓

Description

When WordWrap is enabled, text wraps automatically to the next line when it reaches the right margin of the RichTextEdit control.

Wrap

Applies to

Toolbar controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Determines whether the toolbar items wrap onto a second line. Values are:

- TRUE – (Default) Toolbar items wrap
- FALSE – Toolbar items do not wrap

Usage

In a painter

❖ To select an alignment for a toolbar:

- Select or clear the Label Wrap check box on the General page of the control's Properties view.

In scripts

The Wrap property takes a boolean. The following example wraps the items in a toolbar onto a second line if necessary:

```
tlbr_myToolbar.Wrap = true
```

WrapMode

Applies to

Signature controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Specifies whether text is wrapped to a window or page. Values are:

- SigWrapModePage! – Wraps text to page
- SigWrapModeWindow! – Wraps text to window (default)

Usage**In a painter**❖ **To select a wrap mode for the control text:**

- Select the value you want from the Wrap Mode drop-down list on the General page of the control's Properties view.

In scripts

The WrapMode property takes an enumerated datatype. The following example specifies that text in the control wraps to the page:

```
sig_1.WrapMode = SigWrapModePage!
```

X**Applies to**

Controls, windows

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The X property specifies the X coordinate of an object or control in PowerBuilder units.

The X coordinate is the distance from the left edge of the window or custom user object. If the object is a main window or custom user object, the distance is relative to the screen. If it is not a main window, the distance is relative to the parent window.

Usage**In a painter**❖ **To set the X coordinate:**

- Enter the desired X coordinate, in PowerBuilder units, in the X field of the Other tab page of the object's Properties view, or drag and drop the control to the desired location.

In scripts

The X property takes an integer value. The following line sets the distance from the left edge of a window for a DataWindow control dw_1.

```
dw_1.X = 215
```

You can also set the X and Y properties of a control using the Move function. It is illegal to move a maximized sheet or frame. Changing the X or Y property for a maximized window is ignored.

Y

Applies to

Controls, windows

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✓

Description

The Y property specifies the Y coordinate of an object or control in PowerBuilder units. The Y coordinate is the distance from the top of the window or user object. If the object is a main window or custom user object, the distance is relative to the screen. If it is not a main window, the distance is relative to the parent window.

Usage

In a painter

- ❖ **To set the Y coordinate:**
 - Enter the desired Y coordinate, in PowerBuilder units, in the Y field of the Other tab page of the object's Properties view, or drag and drop the control to the desired location.

In scripts

The Y property takes an integer value. The following line sets the distance from the top of the window for a DataWindow control dw_1.

```
dw_1.Y = 215
```

You can also set the X and Y properties of a control using the Move function. It is illegal to move a maximized sheet or frame. Changing the X or Y property for a maximized window is ignored.

ZoomPercent

Applies to

Signature controls

PocketBuilder on Pocket PC	✓
PocketBuilder on Smartphone	✓
PowerBuilder	✗

Description

Specifies the zoom percent for the control. The default zoom percent is 100.

Usage

In a painter

❖ To select a zoom percent for the control:

- Set the value you want in the ZoomPercent spin control on the General page of the control's Properties view.

In scripts

The ZoomPercent property takes an integer. The following example sets the zoom to 75 percent:

```
sig_mine.ZoomPercent = 75
```


About Display Formats and Scrolling

About this chapter

This chapter describes how to use specific display formats with PocketBuilder controls and provides information about scrolling in PocketBuilder windows and user objects.

Contents

Topic	Page
Using colors with display formats for PocketBuilder controls	563
Using date display formats with PocketBuilder controls	564
Using number display formats with PocketBuilder controls	565
Using string display formats with PocketBuilder controls	567
Using time display formats with PocketBuilder controls	567
Scrolling in windows and user objects	569

Using colors with display formats for PocketBuilder controls

Applies to

Display formats

Description

You can define a color for each display format section by specifying a color keyword before the format. The color keyword is the name of the color, or a number that is the color's RGB value, enclosed in square brackets. For example:

```
[RED] m/d/yy
[255] m/d/yy
```

The following table lists the named color keywords.

Table 4-1: Named color keywords

[BLACK]	[MAGENTA]
[BLUE]	[RED]
[CYAN]	[WHITE]
[GREEN]	[YELLOW]

The formula for combining primary color values into a number is:

$$256 * 256 * blue + 256 * green + red = number$$

where the amount of each primary color is specified as a value from 0 to 255. For example, to specify cyan, substitute 255 for blue, 255 for green, and 0 for red. The result is 16776960.

The following table lists the blue, green, and red values you can use in the formula to specify other colors.

Table 4-2: Values used to specify colors

Blue	Green	Red	Number	Color
0	0	255	255	Red
0	255	0	65280	Green
0	128	255	32768	Dark Green
255	0	0	16711680	Blue
0	255	255	65535	Yellow
0	128	128	328896	Brown
255	255	0	16776960	Cyan
192	192	192	12632256	Light gray

Using date display formats with PocketBuilder controls

Applies to	Display formats
Description	A date display format can have two sections. The first section is required. The second section is optional and specifies how to represent NULLs. date-format;null-format

The following table shows characters that have special meaning in date display formats.

Table 4-3: Special characters in date display formats

Character	Meaning	Example
d	Day number with no leading zero	9
dd	Day number with leading zero if appropriate	09
ddd	Day name abbreviation	Mon
dddd	Day name	Monday
m	Month number with no leading zero	6

Character	Meaning	Example
mm	Month number with leading zero if appropriate	06
mmm	Month name abbreviation	Jun
mmmm	Month name	June
yy	Two-digit year	97
yyyy	Four-digit year	1997

Colons, slashes, and spaces display as entered in the mask.

Usage

If users specify a two-digit year in a DataWindow object, PocketBuilder assumes the date is the 20th century if the year is greater than or equal to 50. If the year is less than 50, PocketBuilder assumes the 21st century.

For example:

- 1/1/85 is interpreted as January 1, 1985
- 1/1/40 is interpreted as January 1, 2040

Examples

The following table shows how the date Friday, Jan. 30, 2003, displays when different format masks are applied.

Table 4-4: Date format examples

Format	Displays
[red]m/d/yy	1/30/03 in red
d-mmm-yy	30-Jan-03
dd-mmmm	30-January
mmm-yy	Jan-03
dddd, mmm d, yyyy	Friday, Jan 30, 2003

Using number display formats with PocketBuilder controls

Applies to

Display formats

Description

A number display format can have up to four sections. Only the first is required.

Positive-format;negative-format;zero-format>null-format

The following table shows characters that have special meaning in number display formats.

Table 4-5: Special characters in number display formats

Character	Meaning
#	A number
0	A required number; a number will display for every 0 in the mask

Dollar signs, percent signs, decimal points, parentheses, and spaces display as entered in the mask.

These keywords tell PocketBuilder to determine an appropriate format based on system settings:

- [General]
- [Currency]

Examples

The following table shows how the values 5, -5, and .5 display when different format masks are applied.

Table 4-6: Number format examples

Sample format	5	-5	.5
[General]	5	-5	0.5
0	5	-5	1
0.00	5.00	-5.00	0.50
#,##0	5	-5	1
#,##0.00	5.00	-5.00	0.50
\$#,##0; (\$#,##0)	\$5	(\$5)	\$1
\$#,##0; -\$#,##0	\$5	-\$5	\$1
\$#,##0; [RED] (\$#,##0)	\$5	(\$5)	\$1
\$#,##0.00; (\$#,##0.00)	\$5.00	(\$5.00)	\$0.50
\$#,##0.00; [RED] (\$#,##0.00)	\$5.00	(\$5.00)	\$0.50
0%	500%	-500%	50%
0.00%	500.00%	-500.00%	50.00%
0.00E+00	5.00E+00	-5.00E+00	5.00E-01

Using string display formats with PocketBuilder controls

Applies to
Description
Examples

Display formats

A string display format can have two sections. The first section is required. The second section is optional and specifies how to represent NULLs.

string-format;null-format

The following table shows characters that have special meaning in string display formats.

Table 4-7: Special characters in string display formats

Character	Meaning
@	A character

All other characters (including spaces) display as entered in the mask.

This format mask:

[red] (@@@) @@@-@@@@

displays the string 800YESCELT in red as:

(800) YES-CELT

Using time display formats with PocketBuilder controls

Applies to
Description

Display formats

A time display format can have two sections. The first section is required and contains the format for times. The second section is optional and specifies how to represent NULLs.

time-format;null-format

The following table shows characters that have special meaning in time display formats.

Table 4-8: Special characters in time display formats

Character	Meaning
h	Hour with no leading zero (for example, 1)
hh	Hour with leading zero if appropriate (for example, 01)

Character	Meaning
m	Minute with no leading zero (must follow h or hh)
mm	Minute with leading zero if appropriate (must follow h or hh)
s	Second with no leading zero (must follow m or mm)
ss	Second with leading zero (must follow m or mm)
ffffff	Microseconds with no leading zeros. You can enter one to six f's; each f represents a fraction of a second (must follow s or ss)
AM/PM	Two-character, upper-case abbreviation (AM or PM as appropriate)
am/pm	Two-character, lower-case abbreviation (am or pm as appropriate)
A/P	One-character, upper-case abbreviation (A or P as appropriate)
a/p	One-character, lower-case abbreviation (a or p as appropriate)

Colons, slashes, and spaces display as entered in the mask.

The keyword [Time] tells PocketBuilder to use the time format specified as the default operating system value for the regional settings.

Usage

24-hour format is the default. Times display in 24-hour format unless you specify AM/PM, am/pm, A/P, or a/p.

Examples

The following table shows how the time 9:45:33:234567 PM displays when different format masks are applied.

Table 4-9: Time format examples

Format	Displays
h:mm AM/PM	9:45 PM
hh:mm A/P	09:45 P
h:mm:ss am/pm	9:45:33 pm
h:mm	21:45
h:mm:ss	21:45:33
h:mm:ss:f	21:45:33:2
h:mm:ss:fff	21:45:33:234
h:mm:ss:ffffff	21:45:33:234567
m/d/yy h:mm	1/30/03 21:45

Scrolling in windows and user objects

For scrolling purposes, PocketBuilder divides the window content into 100 lines and 100 columns. Lines, columns, and pages for scrolling do not correlate with any visible aspect of the window (such as the viewable area).

A *line* or a *column* is the amount scrolled by clicking a scroll bar arrow. There are 100 lines and 100 columns in the control being scrolled, regardless of the area occupied by the content of the window or user object. To get to the end of the scroll bar, the user can click 100 times on the scroll bar arrow.

A *page* is the amount scrolled by clicking in the scroll bar, not on the scroll bar arrows.

Vertical versus
Horizontal scrolling

The procedures in the following sections define vertical scrolling, determined by the `UnitsPerLine` and `LinesPerPage` properties, but the same formulas apply to horizontal scrolling, determined by the `UnitsPerColumn` and `ColumnsPerPage` properties.

Relating scrolling to
height of content

If you want the bottommost content in the window to be visible when the user reaches the end of the scroll bar, you need to set the value of the control's `UnitsPerLine` property so that 100 lines cover the entire contents.

❖ To determine the value for `UnitsPerLine`:

- 1 Resize the window to include all the contents.
- 2 Look at the value of the `Height` option on the `Position` tab page of the window's `Properties` view.

The height is shown in PowerBuilder units (PBUs).
- 3 Divide 75% of the value of the `Height` option by 100 to get the number of PBUs each line should include:

$$\text{UnitsPerLine} = \text{height} * .75 / 100$$

Using 75% of the total height in this calculation keeps the end of the contents visible when the scroll bar reaches the end instead of scrolling just out of sight.

Relating scrolling to
page size

When the user clicks in the scroll bar, not on the scroll bar arrows, the control scrolls by a page. The page size is calculated using this formula:

$$\text{pagesize} = \text{LinesPerPage} * \text{UnitsPerLine}$$

Therefore, you can use the `LinesPerPage` property in conjunction with the `UnitsPerLine` property to set the page size for scrolling.

❖ **To determine the value for LinesPerPage**

- 1 Calculate the value of the UnitsPerLine property, as shown above.
- 2 Size the window to its desired final size.
- 3 Determine the height of the visible window area by looking at the value of the Height option on the Position tab page of the window's Properties view.
- 4 Decide how much of the window you want to scroll every time the scroll bar is clicked. This will give you the page size in PBUs.

For example, if the visible window area height is 1200 PBUs and you want 1/4 of the window to scroll with each click, then the desired page size is 300 PBUs.

- 5 Calculate the value of the LinesPerPage property.

For example:

$$\text{LinesPerPage} = 300 / \text{UnitsPerLine}$$

Scrolling using a fixed
number of clicks

Alternatively, if you want to let the user get to the bottom of the content in a given number of clicks, regardless of the visible window area, set LinesPerPage using this formula:

$$\text{LinesPerPage} = 100 / \text{number of clicks}$$

Index

A

Accelerator property 337
Activation property 338
Address properties for POOMContact 193
Address property 338
Address property for POOMRecipient 194
AddressType property 339
ADOResultSet object 6
Alignment property 340
AllDayEvent property for
 POOMAppointment 190
AllowEdit property 340
Altitude property 341
Anniversary property for POOMContact 192
AppName property 6
AppointmentDuration property for
 POOMAppointment 189
AppointmentEnd property for
 POOMAppointment 189
AppointmentStart property for
 POOMAppointment 189
AssistantName property for POOMContact 192
AssistantTelephoneNumber property for
 POOMContact 192
AutoArrange property 342
AutoHScroll property 342
Automatic property 343
AutoScale property 344
AutoSize property 344
AutoSkip property 345
AutoVScroll property 346
Azimuth property 347

B

BackColor property 347
BarcodeScanner base class 11
BeginX property 349

BeginY property 349
BiometricScanner base class 13
Birthday property for POOMContact 192
Body 165
Body property 350
Body property for NotificationBubble 165
Body property for POOMAppointment 189
Body property for POOMContact 192
Body property for POOMTask 197
BodyInk property for POOMAppointment 189
BodyInk property for POOMContact 192
BodyInk property for POOMTask 197
BoldSelectedText property 351
Border property 352
BorderColor property 353
BorderStyle property 354
BottomMargin property 354
BringToTop property 355
Business properties for POOMContact 192
BusyStatus property for POOMAppointment 189
ButtonHeader property 355

C

CallLogEntry object 15
CallState property 356
Camera object 16
CameraName property 357
CameraType property 357
Cancel property 358
Caption property 359
Caption property for NotificationBubble 165
Categories property for POOMAppointment 189
Categories property for POOMContact 192
Categories property for POOMTask 197
Category property 359
CategorySort property 360
Center property 361
CharSet property 87

- Checked property 361
- Children property for POOMContact 192
- Close property 362
- Colors with display formats 563
- ColumnsPerPage property 363
- CompanyName property for POOMContact 192
- CompletedDate property for POOMTask 197
- Connection object 31
- ContentsAllowed property 366
- ContextInformation object 31
- ContextKeyword object 33
- ControlMenu property 366
- conventions xvii
- CORBACurrent object 34
- CORBAObject object 34
- CORBASystemException classes 217
- CORBAUserException class 91
- CornerHeight property 366
- CornerWidth property 367
- CPUType property 87
- CreateOnDemand property 368

D

- DataObject property 35, 368
- DataSource property 369
- DataType property 370
- DataWindow control properties 44
- Date display formats 564
- DayOfMonth property for POOMRecurrence 195
- DayOfWeekMask property for POOMRecurrence 195
- Default property 371
- DefaultSize property 372
- Degree property 373
- DeleteItems property 373
- Department property for POOMContact 192
- Depth property 374
- DeviceHandle property 375
- DialingDirectory object 66, 67
- DisabledName property 375
- DisableDragDrop property 376
- DisableNoScroll property 377
- DisplayData property 78
- DisplayEveryNLabels property 377
- DisplayExpression property 378

- DisplayName property 379
- DisplayOnly property 379
- DisplayType property 380
- DivideByZeroError object 217
- DocumentName property 380
- DragAuto property 381
- DragIcon property 381
- DriverName property 382
- DriverVersion property 382
- DropLines property 383
- Dropped property 384
- DueDate property for POOMTask 197
- Duration property 384
- Duration property for NotificationBubble 165
- Duration property for POOMRecurrence 195
- DWRuntimeError object 217

E

- EditLabels property 385
- Elevation property 386
- Email properties for POOMContact 192
- Enabled property 387
- Encoding property 388
- EndTime property 388
- EndTime property for POOMRecurrence 195
- EndX property 389
- EndY property 389
- ErrorLogging object 91
- Escapement property 390
- Exception object 91
- ExtendedSelect property 391

F

- FaceName property 392
- FileDirect object 92
- FillColor property 393
- FillPattern property 393
- FirstName property for POOMContact 192
- FixedLocations property 394
- FixedWidth property 395
- FixQuality property 396
- FixTime property 397

Flags property 397
 Flags property for NotificationBubble 166
 FocusOnButtonDown property 399
 FocusRectangle property 399
 Folder property 400
 FontCharSet property 401
 FontFamily property 402
 FontPitch property 402
 Format property 403
 Frame property 404
 FreeDBLibraries property 405

G

GeoidalHeight property 406
 GetItemAtPointer function 299
 GPS base class 222
 GPSCoordinate object 93
 GPSFix object 94
 GPSSatellitesInView object 96
 GraphType property 406
 GroundSpeed property 407

H

HasButtons property 407
 HasLines property 408
 HasNoEndDate property for POOMRecurrence 195
 HDOP property 409
 HeaderFooter property 409
 Heading property 410
 Height property 411
 Hemisphere property 410
 HideSelection property 412
 Home properties for POOMContact 192
 HPBiometricScanner object 113
 HScrollBar property 412
 HSplitScroll property 413
 HTextAlign property 414

I

Icon property 414

IgnoreDefaultButton property 415
 Importance property for POOMTask 197
 Increment property 416
 Indent property 417
 Inet object 124
 InitiallyIconic property 417
 InitiallyIconic property for NotificationBubble 166
 InputEditMode property 418
 InputFieldBackColor property 419
 InputFieldNamesVisible property 420
 InputFieldsVisible property 422
 Instance property for POOMRecurrence 195
 InternetResult object 125
 Interval property for POOMRecurrence 195
 Invert property 420
 IsComplete property for POOMTask 197
 IsFixValid property 421
 IsHeadingValid property 421
 IsMeeting property for POOMAppointment 190
 IsRecurring property for POOMAppointment 190
 IsRecurring property for POOMTask 197
 IsReminderSet property for POOMAppointment 190
 IsReminderSet property for POOMTask 197
 IsSensitive property for POOMAppointment 190
 IsSensitive property for POOMTask 197
 IsTeamTask property for POOMTask 197
 Italic property 422
 Item property 423
 ItemGroup property 423
 ItemPictureIndex property 424, 425
 ItemState property 426
 ItemStyle property 427

J

JaguarORB object 126
 JobTitle property for POOMContact 193

L

Label property 428
 LabelWrap property 430
 Language property 88
 LargePictureHeight property 433

Index

LargePictureMaskColor property 434
LargePictureName property 430
LargePictureWidth property 432
LastName property for POOMContact 193
Latitude property 435
LeftMargin property 436
LeftText property 436
Legend property 437
Limit property 437
LineCallParamFlags property 438
LineColor property 439
LineMediaMode property 440
LinesAtRoot property 441
LinesPerPage property 441
LineStyle property 442
LinkUpdateOptions property 443
LiveScroll property 443
Location property for POOMAppointment 190
Longitude property 444
LowerBound property 10

M

MachineCode property 88
MagneticVariation property 445
MagneticVariationDirection property 445
MajorDivisions property 447
MajorGridLine property 446
MajorTic property 447
Map3DColors property 448
Mask property 449
MaskDataType property 451
MaxBox property 452
MaximumValue property 452
MaxPosition property 453
MaxValDateTime property 453
MenuBar property 454
MenuName property 455
MessageID property 415
MessageStatus property 524
MiddleName property for POOMContact 193
MinBox property 456
MinimumValue property 457
MinMax property 457
MinorDivisions property 459

MinorGridLine property 458
MinorTic property 460
MinPosition property 461
Minute property 461
MinValDateTime property 462
MonthOfYear property for POOMRecurrence 196
MsgMode property 463
Multiline property 464
MultiSelect property 463

N

Name property for POOMRecipient 194
NotificationBubble object 165
NotificationID property 466
NotificationID property for NotificationBubble 166
NullObjectError object 217
Number display formats 565
NumberOfSatellites property 467

O

Occurrences property for POOMRecurrence 196
OfficeLocation property for POOMContact 193
OID property for POOMAppointment 190
OID property for POOMContact 193
OID property for POOMTask 197
OLERuntimeError object 217
OLETxnObject object 168
Options property 467
OriginalSize property 468
OriginLine property 469
OSFixesRevision property 88
OSMajorRevision property 88
OSMinorRevision property 88
OSType property 88
OutlookCompatible property 469
OutlookCompatible property for POOM 187
OverlapPercent property 470

P

PageStyle property 471

Password property 472
 PatternEndDate property for
 POOMRecurrence 196
 PatternStartDate property for
 POOMRecurrence 196
 PBFixesRevision property 88
 PBMajorRevision property 88
 PBMinorRevision property 88
 PBType property 88
 PenMode property 472
 PerpendicularText property 473
 Perspective property 474
 PhoneCall object 170
 PhoneNumber property 474
 PhoneNumberType property 475, 476
 PictureHeight property 477
 PictureIndex property 478
 PictureMaskColor property 480
 PictureName property 481, 482
 PictureOnRight property 483
 PicturesAsFrame property 477
 PictureWidth property 484
 Pointer property 485
 POOM object 187
 POOMAppointment object 189
 POOMContact object 191
 POOMRecipient object 194
 POOMRecurrence object 194
 POOMTask object 196
 PopMenu property 485
 Port property 485
 Position property 486, 487
 PowerTips property 488
 PowerTipText property 487
 PrimaryLine property 489
 PRN property 489
 ProfileCall object 199
 ProfileClass object 200
 ProfileLine object 201
 ProfileRoutine object 203
 Profiling object 205
 ProtocolType property 490

R

RaggedRight property 491
 RawData property 492
 RecurrenceType property for
 POOMRecurrence 196
 ReminderMinutesBeforeStart property for
 POOMAppointment 190
 ReminderOptions property for
 POOMAppointment 190, 198
 ReminderSoundFile property for
 POOMAppointment 190
 ReminderSoundFile property for POOMTask 198
 ReminderTime property for POOMTask 198
 Resizable property 493
 ResultSets object 214
 ReturnsVisible property 493
 RightMargin property 493
 RightToLeft property 7, 494
 Rotation property 494
 RoundTo property 496
 RoundToUnit property 497
 RulerBar property 495
 RuntimeError object 217

S

Satellite property 498
 ScaleType property 498
 ScaleValue property 499
 ScannedData property 500
 ScannedSymbolology property 500
 ScannedTimeStamp property 501
 ScannerType property 501
 Scrolling in windows and user objects 569
 Scrolling property 502
 SecondaryLine property 502
 SelectedTab property 503
 SerialGPS object 222
 SerialPort property 504
 Series property 505
 SeriesSort property 505
 SetStep property 506

ShadeBackEdge property 507
ShowHeader property 508
ShowList property 507
ShowPicture property 508
ShowSIPButton property 509
ShowText property 510
Signature control 223
SIPOnFocus property 510
SmallPictureHeight property 511
SmallPictureMaskColor property 512
SmallPictureName property 513
SmallPictureWidth property 514
SmartMinimize property 515
SMSAddress object 232
SMSMessage object 233
SMSProtocol object 234
SMSSession object 236
SNR property 516
SocketBarcodeScanner object 237
Sorted property 516
SortType property 517
SpacesVisible property 518
Spacing property 518
Spin property 519
SSLCallback object 239
SSLServiceProvider object 239
StartDate property for POOMTask 198
StartTime property 519
StartTime property for POOMRecurrence 196
StatePictureHeight property 520
StatePictureMaskColor property 521
StatePictureName property 522
StatePictureWidth property 523
StdHeight property 525
StdWidth property 526
String display formats 567
Subject property for POOMAppointment 190
Subject property for POOMTask 198
SymbolBarcodeScanner object 249
TabStop property 527
TabsVisible property 531
TabTextColor property 529
Tag property 531
TelephoneNumber properties for POOMContact 192
Text property 533
TextCase property 534
TextColor property 534
TextSize property 535
ThirdState property 537
ThreeState property 536
Throwable object 259
Time display formats 567
Timing object 260
Title property 537
Title property for POOMContact 193
TitleBar property 538
TodayDisplayApp property 539
TodayDisplayText property 540
TodayOrder property 540
TodayRunApp property 541
Toolbar control 260
ToolBar property 542
ToolbarAlignment property 542
ToolbarHeight property 542
ToolbarItem object 262
ToolbarTips property 8
ToolbarUserControl property 8
ToolbarVisible property 543
ToolbarWidth property 543
ToolbarX property 543
ToolbarY property 544
TopMargin property 544
TraceActivityNode object 264
TraceBeginEnd object 265
TraceError object 266
TraceESQL object 267
TraceFile object 269
TraceGarbageCollect object 270
TraceLine object 272
TraceObject object 273
TraceRoutine object 274
TraceTree object 276
TraceTreeError object 277
TraceTreeESQL object 278
TraceTreeGarbageCollect object 280

T

TabBackColor property 530
TabOrder property 527
TabPosition property 528

TraceTreeLine object 281
 TraceTreeNode object 282
 TraceTreeObject object 284
 TraceTreeRoutine object 285
 TraceTreeUser object 286
 TraceUser object 288
 Transaction object 289
 TransactionServer object 291
 typographical conventions xvii

U

Underline property 544
 UndoDepth property 545
 UnitsPerColumn property 545
 UnitsPerLine property 546
 UpperBound property 10
 UseCodeTable property 547

V

ValidityPeriod property 548
 VDOP property 549
 Vendor property 364, 550
 Version property 550
 Version property for POOM 187
 View property 551
 ViewStyle property 552
 Visible property 552
 VScrollBar property 553
 VTextAlign property 554

W

WebPage property for POOMContact 193
 Weight property 554
 Width property 555
 WindowState property 556
 WindowType property 557
 WordWrap property 557
 Wrap property 558
 WrapMode property 558

X

X property 559

Y

Y property 560
 Yomi properties for POOMContact 193

Z

ZoomPercent property 561

