Supported PB Features for Appeon Web

Appeon® for PowerBuilder® 2013 R2

FOR WINDOWS, UNIX & LINUX

DOCUMENT ID: ADC20235-01-0680-01

LAST REVISED: February 11, 2014

Copyright © 2000-2014 by Appeon Corporation. All rights reserved.

This publication pertains to Appeon 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.

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 Appeon Corporation.

Appeon, the Appeon logo, Appeon Developer, Appeon Enterprise Manager, AEM, Appeon Server and Appeon Server Web Component are registered trademarks of Appeon Corporation.

Sybase, Adaptive Server Anywhere, Adaptive Server Enterprise, iAnywhere, PowerBuilder, Sybase Central, and Sybase jConnect for JDBC are trademarks or registered trademarks of Sybase, Inc.

Java and JDBC are trademarks or registered trademarks of Sun Microsystems, 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.

Appeon Corporation, 1/F, Shell Industrial Building, 12 Lee Chung Street, Chai Wan District, Hong Kong.

Contents

1 About This Book

1.1 Audience

This book is for PowerBuilder developers who use Appeon for PowerBuilder to build Web applications. It describes what PowerBuilder features are supported and can be converted to the Web and what features are unsupported.

1.2 How to use this book

There are fourteen chapters in this book.

Chapter 1: About This Book

A general description of this book.

Chapter 2: Basic Requirements and Recommendations

Please make sure your application meets the requirements outlined in this section.

Chapter 3: Web Browser Limitations

Lists the limitations for using Internet Explorer.

Chapter 4: Application Techniques

Instructions on how to apply Appeon-supported application techniques including distributed application support and program access techniques.

Chapter 5: Web Enhancements and Differences

Lists the Web enhancements that an Appeon deployed application can take, and the possible differences it will have with the original PowerBuilder application.

Chapter 6: System Objects and Controls

Provides the supported feature list for PowerBuilder system objects and controls.

Chapter 7: PowerScript Reference

Provides the supported feature list for PowerBuilder programming language and PowerScript.

Chapter 8: Embedded SQL

Provides the supported feature list for database servers, database operators, and database statements.

Chapter 9: Functions

Lists the supported and unsupported system functions, and describes what type of user functions are supported or unsupported.

Chapter 10: Events

Lists the supported event types and system messages.

Chapter 11: DataWindow

Gives a detailed list on the supported features for DataWindow.

Chapter 12: DBParm parameters in DataWindow

Lists the supported DBParm parameters.

Chapter 13: Calling Web Services

Provides the supported interfaces that enable developers to extend the functionality of PowerBuilder.

Chapter 14: Undetected Unsupported features

Lists Appeon unsupported features that cannot be detected in Appeon Developer.

1.3 Related documents

Appeon provides the following user documents to assist you in understanding Appeon for PowerBuilder and its capabilities:

- Introduction to Appeon:

  Gives general introduction to Appeon for PowerBuilder and its editions.

- Getting Started (for Appeon Mobile):

  Guides you though installing PowerBuilder and Appeon for PowerBuilder, and developing and deploying a mobile application.

- New Features Guide:

  Introduces new features and changes in Appeon for PowerBuilder.

- Appeon Mobile Tutorials:

  Gives instructions on deploying, running, and debugging the mobile application, distributing native mobile apps, and configuring the Appeon Server cluster.

- Appeon Mobile (Offline) Tutorials:

  Gives instructions on setting up the Appeon Mobile (Offline) environment, and configuring, deploying, running, updating, and debugging the offline application.

- Appeon Installation Guide:

  Provides instructions on how to install Appeon for PowerBuilder successfully.

- Installation Guide on Cloud Platform:

  Provides instructions on how to install Appeon for PowerBuilder on the cloud-based platform such as Windows Azure and AWS EC2 and S3.

- Mobile UI Design & Development Guide:

  Introduces general guidelines on designing and developing the mobile app and UI.

- Migration Guidelines for Appeon Web:

  A process-oriented guide that illustrates the complete diagram of the Appeon Web migration procedure and various topics related to steps in the procedure, and includes a tutorial that walks you through the entire process of deploying a small PowerBuilder application to the Web.

- Supported PB Features:

  Provides a detailed list of supported PowerBuilder features which can be converted to the Web/Mobile with Appeon as well as which features are unsupported.

- Appeon Developer User Guide:

  Provides instructions on how to use the Appeon Developer toolbar in Appeon for PowerBuilder.

- Workarounds & API Guide:

  Provides resolutions for unsupported features and various APIs to facilitate you to implement the features (including Web and mobile) that are not easy or impossible to implement in the PowerBuilder IDE.

- Appeon Workspace User Guide:

  Gives a general introduction on Appeon Workspace and provides detailed instructions on how to use it.

- Appeon Server Configuration Guide:

  Provides instructions on how to configure Appeon Server Monitor, establish connections between Appeon Servers and database servers, and configure AEM for maintaining Appeon Server and the deployed applications.

- Web Server Configuration Guide:

  Describes configuration instructions for different types of Web servers to work with the Appeon Server.

- Troubleshooting Guide:

  Provides information on troubleshooting issues; covering topics, such as product installation, application deployment, AEM, and Appeon application runtime issues.

- Appeon Performance Tuning Guide:

  Provides instructions on how to modify a PowerBuilder application to achieve better performance from its corresponding Web/mobile application.

- Testing Appeon Web Applications with QTP:

  Provides instructions on how to test Appeon Web applications with QTP.

1.4 If you need help

If you have any questions about this product or need assistance during the installation process, access the Technical Support Web site at http://www.appeon.com/support.

2 Basic Requirements and Recommendations

2.1 Overview

Appeon® for PowerBuilder® is able to deploy most PowerBuilder features to Web successfully. However there are still some unsupported features to remove, general guideline to follow and common pitfalls to avoid during the Web migration process. An application that contains unsupported features and does not adhere to some basic requirements and architectural requirements, may fail to migrate to the Web and suffer from performance problems.

This book will help you:

1. Figure out what PowerBuilder features are supported in Appeon and what are not;
2. Understand Appeon Web features.

2.2 Basic Requirements

There are some basic requirements for the PowerBuilder application which is going to move to the Web with Appeon.

1. The PBL size is suggested less than 500MB. Appeon has successfully tested a PFC application that was artificially inflated to over 500MB.
2. The code is upgraded to be 100% compatible with Appeon-supported environment. For details, refer to the Chapter 3, *Installation Requirements* in *Appeon Mobile Installation Guide for .NET*.
3. The application can generate .exe files (with no errors) in the PowerBuilder version that you install Appeon Developer to.
4. The application has been tested thoroughly to be bug-free.

2.3 Requirements for Upgrading PowerBuilder Applications

The application code must be 100% PowerBuilder 9 or above.

Non-PowerBuilder 9 or above code, including code that is defined as obsolete in PowerBuilder 9 or above help, must be replaced/upgraded before you can begin to use Appeon for PowerBuilder. In addition, supported PowerBuilder 9 or above code is demonstrated in the PowerBuilder Help. Undocumented syntax is unsupported.

All objects that are referenced by PBLs need to be available - PBLs cannot point to objects that do not exist. If a PBL does reference an unavailable object, the Appeon Web Application Deployment process will crash.

Upgrading DataWindow objects to PowerBuilder 9 or above

- In PowerBuilder 9 or above, the DataWindow column name cannot be null (in previous versions of PowerBuilder, this was allowed). Ensure that the column names of DataWindows are not null.

- In the earlier versions of PowerBuilder, the edit style of a DataWindow could be null. The user must make sure that after upgrading to PowerBuilder 9 or above, the edit styles of the DataWindows are not null.
- In the earlier versions of PowerBuilder, the user could use the GetChild function to provide a reference to a child DataWindow in the DataWindow itself.
- If an application is updated to PowerBuilder 9 or above, some DataWindow columns may have duplicate names. Duplicate names for columns in a DataWindow are unsupported.
- If a DataWindow is not correctly upgraded to a PowerBuilder 9 or above format, the problematic DataWindow(s) will be reported with the following information: PBL > DATAWINDOW NAME.

Upgrading Menu objects to PowerBuilder 9 or above

- With menu inheritance, the source code in earlier versions of PowerBuilder is different from the source code in PowerBuilder 9 or above. The user must open the menu objects with inheritance in PowerBuilder 9 or above and upgrade the objects to PowerBuilder 9 or above objects.

Upgrading color specification to PowerBuilder 9 or above

- A color value in an earlier PowerBuilder version may map to a different color of the same value in PowerBuilder 9 or above. If you prefer the color of the earlier version, be sure to verify that the correct colors are selected when upgrading the application to a PowerBuilder 9 or above application.

2.4 Application Language Recommendations

Appeon conducted comprehensive test for the Web application under the following language environment. Other languages can potentially be supported, but they have not been tested.

- English
- Simplified Chinese
- Traditional Chinese
- Korean
- Japanese

3 Web Browser Limitations

Appeon Web applications run well in the Internet Explorer with two limitations:

1. Multiple Appeon applications cannot be run in the same Internet Explorer session. Because of this limitation, you will have the following limitations in IE 7 and IE 8\9.
    - With IE 7, tab browsing multiple Appeon applications that are deployed with same Appeon version is unsupported since all tabs are sharing the same session in IE 7.
    - With IE 8\9, by default, you can run no more than two Appeon web applications in different tab pages, or different IE browsers. The third Appeon web application can not be run with the default IE settings. If you need to run multiple Appeon Web applications (more than 3) with IE 8\9 you can
        - Click the New Session item on the File menu of IE 8\9. Then access Appeon applications in the new opened IE browser.
        - Run iexplore.exe with the -nomerge parameter so that whenever you open a new IE browser, a new session will be created. For example, you can change the shortcut of IE 8\9 to "C:\Program Files\Internet Explorer\iexplore.exe" –nomerge.
2. When the Internet Explorer is refreshed, you need to log in the Web application again.

If you are an Internet Explorer 7 user, there are more known issues:

- IE 7 supports page zoom, enlarging individual Web pages, including both text and graphics, to either focus on specific content or to make content more accessible to those with vision limitations, however, this feature has no effect to Appeon application Web pages and places the UI elements in disorder.
- When a response window of the Appeon Web application is opened in IE 7, if you click the area out of the response window, the focus will be moved away from the window and you will not be able to set focus to the response window again.
- When an Appeon Web application is opened in IE 7, if you move the focus which is currently on the treeview of an MDIClient window to a different tab which has opened a non-Appeon application, and then switch back to the Appeon Web application, the toolbar of IE 7 will not be able to get the focus.
- When IE browser is displayed in full screen, the pop-up menu on the Appeon Web application will be closed automatically, because IE 7 automatically switches the focus.
- When an Appeon Web application is opened in IE 7, thumbnail images are unsupported.
- If the shortcut defined in PowerBuilder application is the same as the shortcut of IE 7, shortcuts in IE 7 have a higher priority when it is converted to the Web.

4 Application Techniques

4.1 Distributed Application Support (EAServer + Windows Only)

Work around Unsupported Features

Appeon supports N-Tier NVOs as well as the distributed DataWindow technique, which synchronizes a DataWindow control on a client with a DataStore on a server using GetFullState, SetFullState, GetChanges and SetChanges. These features can be applied to work around most unsupported features as well as connectivity to Web Services, Java Corba, EJB, C/C++ DLLs (Windows only), and ActiveX components, etc.

The basic requirement for migrating distributed PowerBuilder applications is as follows: all components directly called by the PowerBuilder client must be hosted in one Appeon Server that is used for Web deployment of the PowerBuilder client, and the component type must be a PowerBuilder NVO or a Java Corba component.

The N-Tier NVO that is hosted in one Appeon Server can serve as a "bridge" to call different kinds of middle-tier components that are hosted in different servers.

For instance, if your application calls both .NET components in Web Services and some Java components in EAServer, add an intermediate layer of PowerBuilder NVO components to EAServer: Encapsulate the .NET components and deploy them to EAServer as N-tier NVOs. Modify the application so that it can call the deployed NVO in EAServer instead of directly invoking the .NET components.

Distributed applications involve the use of the PowerBuilder Connection object. Refer to Connection object for information on supported and unsupported features.

4.1.1 N-Tier PowerBuilder NVOs (EAServer + Windows Only)

Supported

- The variables used in n-Tier PowerBuilder NVO can be the following standard PowerScript data types:

  Table 4.1:

Boolean	Blob	Char	Date*
ateTime	Decimal	Double	Int
Integer	Long	Real	String
Time	UInt	ULong*	UnsignedInt
UnsignedInteger*	UnsignedLong*		

- Functions in NVO components can use UnsignedLong or UnsignedInteger as arguments, but not as return values.
- Functions in NVO components can use array, structure and reference as arguments.
- The parameter and return value of an NVO public interface can be arrays, complex data types, or structures.

- Declaring external functions (such as API functions, DLLs) in the NVO is supported if Appeon Server runs on Windows. External Function DLLs in Appeon Server/EAServer PowerBuilder components have no issues from PowerBuilder's perspective, but if the external function is not thread-safe, that can certainly cause a problem. It's dependent on the DLL's implementation, which has nothing to do with Appeon Server/EAServer. If it's not threadsafe you can set the com.sybase.jaguar.component.thread.safe property of the PowerBuilder component to false and that will prevent multiple threads from trying to access the external function simultaneously.

Differences

- The precision of the decimal number, which is the return value of NVO components, is different from that in PowerBuilder and on the Web. This is caused by the different precision between PowerScript and JavaScript.

Unsupported

The n-Tier PowerBuilder NVOs cannot use the following features:

- PowerScript MessageBox function in the NVO
- Global variables
- Visual controls or objects
- Any visual control/object data type as the parameters for NVO functions and events

For calling and using remote EAServer PowerBuilder NVO components in the PowerBuilder client, some limitations are listed below:

- The proxy object for an NVO cannot use the same name as the NVO.
- It is impossible to use Dot Notation to refer to the instance variables of a remote NVO. Appeon suggests that you add functions to the remote NVO to get/set its instance variable values.
- You cannot trigger or post the events of a remote NVO in the PowerBuilder client.
- The script for an NVO or the script referred to in an NVO cannot contain UI-related scripts or objects.
- Web applications are unsupported to call NVOs that contain inheritance hierarchy.
- The parameter of an NVO public interface cannot be Date or Time in EAServer 6.x.
- The return value of an NVO public interface cannot be decimal in EAServer 6.x

4.1.2 N-Tier DataWindows (EAServer + Windows only)

Appeon Server installed to EAServer on Solaris/AIX can only support n-Tier DataWindows that do not call the GetChanges, GetFullState, SetChanges, and SetFullState functions.

Appeon Server installed to EAServer on Windows can support n-Tier DataWindows that call the GetChanges, GetFullState, SetChanges, and SetFullState functions, as long as the Appeon Workaround PBL is implemented.

For more instructions on how to use the Appeon Workaround PBL, please refer to Section 2.3.5, “Distributed DataWindows (EAServer only)” in *Workarounds & API Guide*.

4.2 Program Access Techniques

4.2.1 Using OLE in an application

OLE, originally an acronym for Object Linking and Embedding, is a facility that allows Windows programs to share data and program functionality. Appeon supports the use of OLE in three ways:

- OLE control
- OLECustomControl
- OLE object

Table 4.2:

OLE controls	**Supported**
	OLEControl control OLECustomControl control
	Unsupported
	OLE Control of DataWindow
OLE objects	**Supported**
	OLEObject object
	Unsupported
	OLEStorage OLEStream OLERuntimeError OLERuntime OLETxnObject Large Binary/Text database OLE Object of DataWindow
Calling OLE object functions	**Requirements**

	Using parentheses when calling an OLEObject method is required, even if the OLEObject method does not require any parameter. For example, when executing the Save function for a Microsoft Word document:
	`o1.ActiveDocument.Save //Unsupported`
	`o1.ActiveDocument.Save() //Supported`

	Unsupported
	1. OLEObject object does not support cascaded calling. The following example is unsupported: `OleObject.function1.function2()`

	2. The OLE object method cannot contain reference parameters.
Shortcut	**Unsupported**
	Using shortcuts in the OLECustomControl control are unsupported.
Requirements for client environment	**Requirements**
	When an OLE object is called in a deployed application, the location used to store the OLE object on the client should match the location specified in the original PowerBuilder application. ActiveX control(s), if used in the Web application, should be installed and registered on the client. There are two ways to install and register ActiveX controls: Manually install and register ActiveX controls that are associated with the Web application. Deploy ActiveX controls that are associated with the Web application to Appeon Web Server, so ActiveX controls can be automatically downloaded and registered on the client. For detail information on deploying ActiveX controls to the Web Server, refer to Section 4.2.1.2.3, "Additional Files" in *Appeon Developer User Guide*.

4.2.2 Using external functions

Table 4.3:

Data types of external function arguments	**Supported**			
	Array	Boolean	Blob	Char
	Date	DateTime	Decimal*	Integer
	Long	Real	String*	Structure
	Void			
	String The length of an argument or return value as String data type can not be larger than 512 characters. *Decimal* The precision of Decimal argument is the same as the precision of Double data type.			
	Unsupported			
	LongLong	Object	UInt	UsignedInt
	UnsignedInteger	UnsignedLong		
Data types of external function return values	**Supported**			
	Array	Boolean	Blob	Char
	Date	DateTime	Double	Decimal*
	Integer	Long	Real	String*
	Unsigned Integer	Unsigned Long	Void	

	String The length of an argument or return value as String data type can not be larger than 512 characters. *Decimal* The precision of Decimal argument is the same as the precision of Double data type.
	Unsupported LongLong, Structure, Object
DLLs or shared libraries Location	**Supported** When your application calls an external function, it must be able to find the DLL (Dynamic link library) or the shared library in which that function resides. To ensure this, you must update the library path environment variable on the client machine to include the directory where that DLL or shared library is stored. The directory is searched in the following order of precedence: 1. Windows system directory • Windows 2000/2003: The 32-bit Windows system directory. Use the GetSystemDirectory function to get the path of this directory. The name of this directory is SYSTEM32. or • Windows NT/2000: The 16-bit Windows system directory. There is no Win32 function that retrieves the path of this directory, but it is searched. The name of this directory is SYSTEM. 2. The Windows directory uses the GetWindowsDirectory function to get the path of this directory. 3. The directories that are listed in the PATH environment variable. When you make changes to the environment variables, the environment variables take effect only after you restart the machine.
Ensuring the resource file in an external function when running Web application	**Supported** 1. If the file is an Image, directly specify the file name. When you deploy the application with Appeon Developer, you need to deploy the Image file with the application files to the Web server. For instructions, refer to Section 4.2.1.2.3, “Additional Files” in *Appeon Developer User Guide*. 2. If the file is a DLL or OCX, there are two ways to ensure the resource file in an external function: • Directly specify the file name. When you deploy the application with Appeon Developer, you need to deploy the DLL or OCX file

	with the application files to the Web server. For instructions, refer to Section 4.2.1.2.3, “Additional Files” in *Appeon Developer User Guide*. • Package the file using the DLL/OCX file packaging wizard. Install them to the client before the application is run for the first time. Refer to Section 17.1.1, “Packaging DLL/OCX files” in *Appeon Developer User Guide*. 3. If the file type is not Image, DLL, or OCX, specify the file name and the absolute file path. The client that runs the application must have the file ready in the same path before executing the external function.
Others	**Supported**
	Calling external functions in Win32-based DLLs (including both the system DLLs and the user-defined DLLs) is supported. API for file and directory access operations is supported.
	Unsupported
	Graphs created by Windows API function may not be displayed on the Web. Calling external functions in the Timer event will result in unexpected errors on the Web if the external functions are called in other places of the application.

4.2.3 Using Run PowerScript function

Table 4.4:

Syntax	**Supported**
	Run(string {, windowstate}) The value of a string argument can be a filename without a path or extension. The following examples are supported: *run("notepad")* *run("notepad.exe")* *run("C:\winnt\system32\notepad")* *run("C:\winnt\system32\notepad.exe")*
Application programs location	**Supported**
	When you call the Run function and do not specify the path of the application program to the string argument, make sure that the application program is stored in one of the following directories. If it is not, an execution error will occur. The directory is searched in the following order: 1. Windows system directory

- Windows 2000/2003: The 32-bit Windows system directory. Use the GetSystemDirectory function to get the path of this directory. The name of this directory is SYSTEM32.

  or

- Windows NT/2000: The 16-bit Windows system directory. There is no Win32 function that retrieves the path of this directory, but it is searched. The name of this directory is SYSTEM.

2. The Windows directory

   Use the GetWindowsDirectory function to get the path of this directory.

3. The directories that are listed in the PATH environment variable.

   When you make changes to the environment variables, the environment variables take effect only after you restart the machine.

4.2.4 Building a mail-enabled application

Appeon supports MAPI (messaging application program interface), so you can enable your Web applications to send and receive messages using any MAPI-compliant electronic mail system.

Appeon supports the following mail related objects and structures to enable the mail service.

Table 4.5:

A mail-related system object	**Supported**		
	MailSession		
Mail-related structures	**Supported**		
	MailMessage	MailFileDescription	MailRecipient
	Unsupported		
	Properties of the three structures cannot be NULL.		
Object-level functions for the MailSession object	**Supported**		
	mailAddress	mailDeleteMessage	mailGetMessages
	mailHandle	mailLogoff	mailLogon
	mailReadMessage	mailRecipientDetails	mailResolveRecipient
	mailSaveMessage	mailSend	
Enumerated data types	**Supported**		
	MailFileType	MailLogonOption	MailReadOption
	MailRecipientType	MailReturnCode	

4.2.5 Using Drag and Drop

Drag and drop allows users to initiate activities by dragging a control and dropping it on another control. Except OLE, drag and drop is supported for other draggable controls and objects with features listed in the table below:

Table 4.6:

Drag and Drop properties	**Supported**			
	DragAuto	DragIcon		
Drag and Drop Events	**Supported**			
	BeginDrag	BeginRightDrag	DragDrop	DragEnter
	DragLeave	DragWithin		
	Unsupported			
	If the DragIcon is not specified for a control except the ListView or TreeView control, pressing the left mouse button without moving will trigger both DragEnter and DragWithin events in PowerBuilder, however, DragEnter event only on the Web.			
Drag and Drop Functions	**Supported**			
	Drag	DraggedObject		
Drag mode	**Unsupported**			
	Key event is unsupported if controls are in drag mode.			
UI differences during the drag and drop	**Difference**			
	(For all supported draggable controls except the TreeView and ListView controls) The icon of the mouse will not change, if it is not on the client area of a control. (For TreeView and ListView) When you drag a control, you cannot see the mouse icon during the drag. Appeon does not support dragging and dropping multiple controls simultaneously. You can only drag and drop one control at a time.			

4.3 Using Unicode

Appeon supports the Unicode character encoding in PowerBuilder 10 or above. You can display characters from multiple languages on the same page of a Web application.

String related functions

Appeon supports the string related functions in both PowerBuilder 9 and PowerBuilder 10 or above. The following changes, made since PowerBuilder 10, are also supported.

1. Modified processing of string-manipulation functions.
2. Asc and Char functions.
3. Blob and String conversion functions.

Support ANSI and Unicode database

Appeon supports both ANSI and Unicode database. For Unicode database it is recommended to use iAnywhere and jConnect driver because using JDBC-ODBC driver with Unicode databases may cause problems.

Calling external functions

Appeon supports both Unicode encoding and ANSI encoding when the Web application calls external functions.

5 Web enhancements and differences

5.1 Appeon enhancements and differences

5.1.1 Appeon security enhancement

If your PowerBuilder application has no coded username/password verification during application startup, Appeon's built-in user group management can assist you. Refer to Section 2.2.3, “Configuration during security management” in *Appeon Server Configuration Guide for .NET* or Appeon Server Configuration Guide for J2EE. When using Appeon's built-in security, you will be prompted to enter a user name and password in the Appeon Login dialog box.

5.1.2 Installing Appeon ActiveX control

There are two ways to install the Appeon ActiveX control:

- **Automatic installation**: Appeon ActiveX will be automatically downloaded and installed to the client when you access the Web application with correct IE settings and user privileges. For step-by-step instructions on configuring the IE settings and user privileges, refer to Section 6.2, “Configuration for client machine” in *Appeon Web Installation Guide for .NET*.

- **Manual installation**: Manually download or copy the zip package of ActiveX control to the client, decompress the zip package, right click the Update.exe file in the package and select "Run as administrator" to install the Appeon ActiveX control.

  The end user will be prompted to manually download the zip package when they access the Web application with insufficient user privileges or incorrect IE settings. You can also find the zip package (ax_install_x32.zip for 32-bit client or ax_install_x64.zip for 64-bit client) under the Appeon Server installation directory (for example, C:\inetpub\wwwroot\appeon \weblibrary_ax\support) and send it to the end users.

  Alternatively, the user can open the command prompt window and install the Appeon ActiveX with the command line, for example, C:\ax_install_x32\update.exe. More command-line options are provided below.

  Appeon command-line options

  Starting from Appeon 6.1, Appeon provides the following command-line options to enable you silently install or uninstall the Appeon ActiveX control. You should execute these command lines in the command prompt window.

  Table 5.1:

Option	Description
-u	Uninstalls the Appeon ActiveX.
-s	Silently executes the command. With this command you will not see any interface that relates to the installation/uninstallation of the Appeon ActiveX control. For example, to silently install the Appeon ActiveX control:

Option	Description

	`C:\ax_install_x32\Update.exe -s`
	For example, to silently uninstall the Appeon ActiveX control:
	`C:\ax_install_x32\Update.exe -s -u`

-f	Prompts the failure message. This option has to be used with -s. When the two options used together, it means that messages will display if there is any failure (such as errors and warnings) during the installation process of the Appeon ActiveX control.
-r	Reverses the old files. This option has to be used with -s.
-l	Prints the log file with a specified path. This option has to be used with -s. For example: `C:\ax_install_x32\Update.exe -s -l "c:\test.log"`

5.1.3 Appeon Server open interfaces

Overview

Appeon Server open interfaces give users the opportunity to manage services provided by Appeon Server through PowerBuilder code.

Appeon Server open interfaces

There are four open interfaces provided by Appeon Server. You can write code for calling the interface in the PowerBuilder application, but the code only takes effect in the Appeon-deployed application.

1. killAllSessions

   Description

   KillAllSessions kills all active sessions in an Appeon Server or an Appeon Server cluster and rolls back all associated transactions. To kill all sessions in an Appeon Server cluster, you need to first configure the cluster in AEM.

   Syntax

   proxyobject.killAllSessions (String *serverName*)

   Table 5.2:

Argument	Description
proxyobject	The instance of the proxy object generated for the component OpenInterface.
serverName	The name of Appeon Server for which you want to kill all sessions.

   Return value

   Boolean.

It returns true if it succeeds and false if it fails. If serverName is NULL or empty string (""), it kills all sessions and rolls back all associated transactions in an Appeon Server cluster.

2. rollbackAllTransactions

Description

RollbackAllTransactions rolls back all transactions in an Appeon Server or an Appeon Server cluster. To roll back all transactions in an Appeon Server cluster, you need to first configure the cluster in AEM.

Syntax

proxyobject.rollbackAllTransactions (String *serverName*)

Table 5.3:

Argument	Description
proxyobject	The instance of the proxy object generated for the component OpenInterface.
serverName	The name of the Appeon Server that you want to kill all sessions on.

Return value

Boolean.

It returns true if it succeeds and false if it fails. If serverName is NULL or empty string (""), it rolls back all transactions in an Appeon Server cluster.

3. getAllClients

Description

GetAllClients gets the IP addresses of all client machines which corresponds to the active sessions that are opened for the specified application in the specified Appeon Server.

Syntax

proxyobject.getAllClients (String *serverName,String appName)*

Table 5.4:

Argument	Description
proxyobject	The instance of the proxy object generated for the component OpenInterface.
appName	The name of the application that is deployed to the specified Appeon Server.
serverName	The name of the Appeon Server in which the active sessions are created. The value should be IP address or IP address with port number, for example, 192.0.0.123, or 192.0.0.123:8080.

Return value

String.

Returns the IP addresses that correspond to the active sessions opened for the specified application in the specified Appeon Server if it succeeds and one of the following negative values if an error occurs.

-1: function fails to execute

-2: the specified Appeon Server is not found

-3: the specified application is not found

If appName is NULL or empty string (""), the methods will apply to all applications in the specified Appeon Server. If serverName is NULL or an empty string (""), **GetAllClients** returns all the corresponding IP addresses for the specified application in all servers in an Appeon Server luster. Note you need to configure the cluster in AEM first.

4. getAllSessions

Description

GetAllSessions returns the detail information of active sessions with XML format, which are opened for the specific application in the specific Appeon Server.

Syntax

proxyobject.getAllSessions (String *serverName*,String *appName*)

Table 5.5:

Argument	Description
proxyobject	The instance of the proxy object generated for the component OpenInterface.
appName	The name of the application that is deployed to the Appeon Server.
serverName	The name of the Appeon Server that the sessions are created in.

Return value

XML string.

Method execution results are recorded in the error node.

Two parameters, code and text, are contained in the error node. The code parameter shows the execution result and the text parameter records the detail error information. If the execution result is not 0, only the error node is listed in the XML.

Values for the *code parameter:*

0: function succeeds in execution.

-1: function fails to execute.

-2: the specified Appeon Server is not found.

-3: the specified application is not found.

If appName is NULL or an empty string (""), **GetAllSessions** returns the detail information of active sessions opened for all applications in the specified Appeon Server. If serverName is NULL or an empty string (""), **GetAllSessions** returns the detail information of active sessions in an Appeon Server cluster. Note you need to configure the cluster in AEM first.

Applying Appeon Server open interfaces

There are two steps that should be performed when applying Appeon Server open interfaces:

1. Call the Appeon Server open interface in the PowerBuilder application.
2. Deploy the PowerBuilder application to Appeon Server the same way you would deploy a normal PowerBuilder application.

5.1.3.1 Calling Appeon Server open interfaces for J2EE application servers

Appeon Server open interfaces (methods) in EAServer 6.x, JBoss, JEUS, WebLogic and WebSphere are encapsulated in a standard EJB component named OpenInterface that is included in the ASInterface package. To invoke the open interfaces in PowerBuilder, you need to follow the step instructions of how to call Appeon EJBObject. Details refer to Section 2.3.7, “Calling EJB Component” in *Workarounds & API Guide*.

Following is general step guide to invoke the open interface in PowerBuilder.

Step 1: Load Appeon Workaround PBL to the application.

Step 2: Deploy Appeon Bridge.

Step 3: Connect to the EAServer where Appeon Server and Appeon Bridge are installed through Appeon EJBobject. For example:

```
string ls_prop[5]
string ls_serurl,ls_msg
ls_serurl = "http://192.0.3.35:8000/appeonbridge/Dispatch"
ls_prop[1]= "applicationA"
ls_prop[2]=
 "javax.naming.Context.INITIAL_CONTEXT_FACTORY='com.sybase.ejb.InitialContextFactory'"
ls_prop[3]= "javax.naming.Context.PROVIDER_URL='iiop://192.0.1.249:2000'"
ls_prop[4]= "username=admin@system"
ls_prop[5]= "password=appeon249"
ls_msg = io_ejb.connectserver(ls_serurl, ls_prop)
if ls_msg <> "" then
MessageBox("Connect Application Server error!", ls_msg)
return
end if
```

Step 4: Create an instance of the EJB component OpenInterface.

The JNDI name for EAServer is ejb.components.appeonserver.appeonserver.OpenInterfaceRemoteHome, and the JNDI name for WebSphere, WebLogic, JEUS, & JBoss is OpenInterfaceBean. For example:

```
String ls_jndi, ls_home, ls_method,ls_msg
ls_jndi = "ejb.components.appeonserver.appeonserver.OpenInterfaceRemoteHome"
ls_msg = io_ejb.createremoteinstance(ls_jndi, ls_home, "create", ref il_bean1)
if ls_msg <> "" then
MessageBox("create remote instance faild!", ls_msg)
return
end if
```

Step 5: Invoke one or more open interfaces in the component. For example:

```
string ls_msg
int c_val
string retval01
c_val = -1
io_ejb.regstring("")
io_ejb.regstring("")
ls_msg = io_ejb.invokeretstring (il_bean1, "getAllSessions",true, ref retval01)
if ls_msg = "" then
messagebox('',"retval=" + string(retval01))
else
messagebox("","retval=" + ls_msg)
end if
```

5.1.3.2 Calling Appeon Server open interfaces for IIS Server

Appeon Server open interfaces (methods) in .NET are encapsulated in a standard .NET/COM components. To call .NET/COM components, Appeon provides a non-autoinstantiated NVO - AppeonDotNetComponent - as the proxy object to call the server-side components. Details refer to Section 2.3.8, “Calling .NET/COM server components (.NET only)” in *Workarounds & API Guide*.

Following is general step guide to invoke the open interface in PowerBuilder.

Step 1: Load Appeon Workaround PBL to the application.

Step 2: Make sure the COM component have been registered using the regsvr32 tool and the components have been copied to the **%appeon%/AEM/components** folder on the Appeon Server machine.

Step 3: Create an instance of the AppeonDotNetComponent. For example:

```
String servername,appname
AppeonDotNetComponent loadDBList
Any la_l[]
Long lRet
serverName = ""
appname = ""
la_l[1] = servername //
la_l[2] = appname //
loadDBList = Create
AppeonDotNetComponent
```

Step 4: Specify the ComponentType and Set the TypeLib property and the ClassDescript property to empty. For example:

```
loadDBList.ComponentType = "4"
loadDBList.TypeLib = ""
loadDBList.ClassDescript = ""
```

Step 5: Invoke one or more open interfaces in the components. For example:

```
lRet = loadDBList.of_ExecInterface
("GetSessionCount",la_l)
If lRet = 0 Then
MessageBox("Get session count
successfully.","Session Count:"+string
(loadDBList.ReturnValue))
End If
```

5.1.4 Application differences

User interface and operation difference

- There may be slight differences in color and size between the Web UI and the PowerBuilder UI due to minor mismatching of the conversion algorithm for converting PowerBuilder measurements to Web measurements.
- Using the middle mouse key to operate the Web application is unsupported.
- Keyboard Navigation: tab order and arrow keys are partially supported.
- All Appeon system error messages appear in the Internet Explorer status bar rather than in a dialog box.
- While a Web application is running and the TextSize in the Internet Explorer is reset, the Web application will not change its font size accordingly.

Design-time property settings difference

- Appeon creates the resulting Web application based on design-time property settings of the application obtained from PowerBuilder. Therefore, even if the PowerBuilder application does not work properly at run-time, the Appeon-deployed Web application may still run properly. For instance, in a particular build, the PowerBuilder Edit Mask DataWindow column style did not run properly in the PBVM, but the Appeon for PowerBuilder Edit Mask DataWindow column style ran successfully because the Appeon framework for Web applications works properly.

Printing

- The manual feeder feature of Print is unsupported.

Section 508 compliance

All standard controls supported by Web applications are section 508 compliant but with the following noted issue on keyboard focus:

- When an Appeon Web application runs in Internet Explorer, the keyboard focus, by default, is on the Web application. You can press Alt+Tab to shift the focus to Internet Explorer, and then press Tab to shift the focus back to the application.
- You can press Alt to access the menu of the application (or Internet Explorer) when the keyboard focus is on the application (or Internet Explorer).

5.2 DataWindow enhancements and differences

5.2.1 Appeon DataWindow menu

Appeon DataWindow menu client functions

You can enable the Appeon DataWindow Menu by calling two Appeon client functions that are defined in the Appeon Workarounds PBL.

1. AppeonPopMenu function

Description

Pops up Appeon DataWindow menu at a specified position in a specified DataWindow control.

Syntax

AppeonPopMenu (datawindow *adw_dw*, Integer *nx*, Integer *ny*)

Argument	Description
adw_dw	The DataWindow control on which you want to pop up the Appeon DataWindow menu.
nx	The instance from the left edge of the DataWindow control.
ny	The instance from the up edge of the DataWindow control.

Return value

None.

Usage

- User customized RMB menus should have a higher priority than Appeon customized menus.
- The AppeonPopMenu function has a higher priority than the AppeonPopMenuOn function.
- Defining your RMB menu in RButtonDown event is not recommended because this will cause the confusion in the system. To work around this, define your RMB menu in the RButtonUp event.

2. AppeonPopMenuOn function

Description

Pops up Appeon DataWindow menu in a specified window when you right click the mouse button.

Syntax

AppeonPopMenuOn (datawindow *adw_dw*, Boolean *ab_show*)

Argument	Description
adw_dw	The DataWindow control on which you want to show the Appeon DataWindow menu.
ab_show	The Boolean value that decides whether to show Appeon DataWindow menu. • True - Enables the display of the Appeon DataWindow menu. • False - Disables the display of the Appeon DataWindow menu.

Return value

None.

Usage

- User customized RMB menus should have a higher priority than Appeon customized menus.
- The AppeonPopMenu function has a higher priority than the AppeonPopMenuOn function.
- Defining your RMB menu in RButtonDown event is not recommended because this will cause the confusion of the system. To work around this, define your RMB menu in the RButtonUp event.

How to invoke an Appeon DataWindow Menu

There are two ways to invoke an Appeon DataWindow menu:

1. Execute the AppeonPopMenu function for a specified DataWindow.
2. After the AppeonPopMenuOn function is executed for a specified DataWindow, right click on a DataWindow (the area clicked should not be a DataWindow field that has focus).

Note: If the area clicked is a DataWindow field that has focus, the DataWindow menu that pops up is the same as it would be in PowerBuilder, and contains the following items: Undo, Cut, Copy, Past, Delete, and Select All.

Appeon DataWindow Menu Items

Menu Items	Functionality
Find	Finds a record with user-specified text.
Find next	Finds the next record with the specified text.
Find previous	Finds the previous record with the specified text.
Sort	Displays a sort dialog for the user to specify sorting criteria and sorts the DataWindow.
Filter	Displays a filter dialog for the user to specify filtering criteria and filters the DataWindow records. This is done on the client side.
Print...	Prints the content in the DataWindow with a Client printer. • At the click of the Print button, a Print Setup dialog pops up with the names of all the Client printers listed for the printing job. • Using PDFPrinter on the Client for printing is not recommended. If the default printer on the Client is PDFPrinter, the printed result is saved to the "\My

Menu Items	Functionality
	Documents\Appeon Printer" folder on the Client. By default, the PDF file that is generated is only named with the .pdf extension and will have no name. If a previously printed .pdf document resides in the "Appeon Printer" folder then it will be overwritten the next time a document is printed to the folder. If the previously printed .pdf file is open, printing of the new document will fail.

5.2.2 DataWindow printing

Starting from Appeon 6.5 all the printing will be done on the client side. A printing command is directly sent to a printer at the client.

DataWindow printing

To print the DataWindows at the Client, you can choose any of the following methods. There are slight differences between the two methods.

1. Using the Print item in the Appeon DataWindow Menu. At the click of the Print... button in the Appeon DataWindow, a Print Setup dialog pops up with the names of all the Client printers listed. The user can select one of the printers for printing the DataWindow. For more information, refer to Appeon DataWindow menu.

2. Executing the DataWindow Print function. This is the same as in PowerBuilder applications: when the DataWindow Print function is executed in the Web application, the DataWindow is printed on the Client's default printer.

PDF printing

There are two ways to print the DataWindow into PDF format.

1. Call the print function of the DataWindow and set the default printer on the client side to a PDF printer.

2. Call SaveAs function of the DataWindow or Child DataWindow and set the saveas file type to PDF* format.

   Save as PDF specification

   a. The following settings in the Print Specification property window are supported as the printing parameter during the PDF generation: LeftMargin, RightMargin, BottomMargin, PaperSize, Display Buttons-Print.

   b. The PDF file is not editable and can only be created in a monitor with 32-bit color quality settings or above.

   c. The page frame size setting is restrained by the type of operating system. For example, in Windows Vista, the upper limit of customized page frame size is around 5500*5500. The page frame size can also be 1024*10000 by reducing page width and increasing page height in the meanwhile.

   d. For the best display quality it is recommended to set the display DPI to 96.

5.2.3 DataWindow user operation differences

User interface differences

- In some cases, text in the DataWindow will appear truncated or UI elements may not be fully visible in the Window. This is because the PowerBuilder units used to size the UI of the application, such as the DataWindow rows and columns, the Window object, etc, cannot be converted to the Web perfectly.
- The number of data records displayed per page in deployed DataWindows may be different from that in PowerBuilder.

DataWindow modify/update

- When you click a date field in a DataWindow, the format of the date is changed to format yyyy-mm-dd, regardless of what the original display format is.
- If a cell in a DataWindow contains a value that is composed of only spaces and no other characters, the update to the database will fail (the value is updated to the database as an empty string rather than blank spaces). The retrieving functionality still works correctly and is not affected. Workaround: If a cell must be empty, make sure its value is an empty string, not a string composed of only one or more spaces. This issue affects Sybase ASE, Oracle, and Microsoft SQL Server. It does not affect Sybase ASA/SQL Anywhere.
- In PowerBuilder, if the user drags the scroll bar while a DataWindow field is being edited, AcceptText will not be executed for the field. In a Web DataWindow, AcceptText is executed for the field.

More behavioral differences

- Motion effect of GIF files is unsupported.

5.3 Object Control Enhancements and Differences

5.3.1 Object/control enhancements

MultiLineEdit

- In the Web MultiLineEdit control, the user can drag and drop the selected text.
- In the Web MultiLineEdit control, the user can select characters by mouse dragging and right-click on the selected characters to display the popup menu to copy/cut/paste the characters.

SingleLineEdit

- In the Web SingleLineEdit control, the user can select characters by mouse dragging and right-click on the selected characters to display the popup menu to copy/cut/paste the characters.

5.3.2 Object/control user operation differences

Controls

Controls in the Web application get focus when SetFocus is triggered, while controls in the PowerBuilder application get focus when the LoseFocus event of the control (which is to lose focus) is triggered. For example, when focus is moved from Control A to Control B due to the user operation, on Web, Control B gets focus when the SetFocus event of Control B is triggered, while in PowerBuilder, Control B gets focus when the LoseFocus event of Control A is triggered. This does not affect the user operation, but may affect the execution result of script dependent on the focus.

DropdownListBox, DropdownPictureListBox, ListBox or PictureListBox

The size of the pictures displayed in a PictureListBox will always be 16px * 16px. It will not adjust according to the actual height and width available as it does in PowerBuilder.

Motion effect

Motion effect of GIF files is unsupported.

6 System Objects and Controls

6.1 Controls

Important Requirements

- Supported controls can be dynamically created (e.g. using the CREATE statement).
- Creating a new control in a child object by copying and pasting an existing control in the PowerBuilder painter is unsupported if the existing control is inherited from the parent object.

Supported

- The following list includes the controls Appeon supports:

Table 6.1:

CheckBox	CommandButton	DatePicker	DataWindow
DropDownListBox	DropDownPictureListBox	EditMask	GroupBox
HProgressBar	HScrollBar	HTrackBar	Line
ListBox	ListView	MonthCalendar	MultiLineEdit
OLEControl	OLECustomControl	Oval	Picture
PictureButton	PictureHyperLink	PictureListBox	RadioButton
Rectangle	RichTextEdit	RoundRectangle	SingleLineEdit
StaticHyperLink	StaticText	Tab	TreeView
VProgressBar	VScrollBar	VTrackBar	Window

- For information on properties, events and functions of supported controls, refer to the remaining sections in this chapter.
- For DataWindow controls, please refer to the DataWindow section.

Unsupported

- It is unsupported to change the skin of system standard controls on the Web.

6.1.1 Supported Controls

6.1.1.1 CheckBox control

6.1.1.1.1 Properties for CheckBox control

Table 6.2:

Property	Support Level	Example Code
Automatic	Supported	cbx_1.Automatic = TRUE
BackColor	Supported	cbx_1.BackColor = ll_value

Property	Support Level	Example Code
		cbx_1.BackColor = RGB(192,192,192)
BorderStyle	Supported	cbx_1.BorderStyle = StyleLowered! cbx_1.BorderStyle = StyleRaised!
BringToTop	Supported	cbx_1.BringToTop = TRUE
Checked	Supported	cbx_1.Checked = TRUE cbx_1.Checked = lb_value
DragAuto	Supported	cbx_1.DragAuto = TRUE
DragIcon	Supported	cbx_1.DragIcon = 'c:\archive\arrow.ico'
Enabled	Supported	cbx_1.Enabled=TRUE cbx_1.Enabled = lb_value
FaceName	Supported	cbx_1.FaceName = "" cbx_1.FaceName = ls_value
FontCharSet	Supported	cbx_1.FontCharSet = ANSI!
FontFamily	Supported	cbx_1.FontFamily = Roman!
FontPitch	Supported	cbx_1.FontPitch = Fixed!
Height	Supported	cbx_1.Height=889 cbx_1.Height = li_value
Italic	Supported	cbx_1.Italic=TRUE cbx_1.Italic = lb_value
LeftText	Supported	cbx_1.LeftText = TRUE
Pointer	Supported	cbx_1.Pointer='Cross!' cbx_1.Pointer='d:\archive\IBEAM.CUR'
RightToLeft	Supported	cbx_1.RightToLeft = TRUE
TabOrder	Supported	cbx_1.TabOrder = 10
Tag	Supported	cbx_1.Tag = ls_value
Text	Supported	cbx_1.Text = ls_value
TextColor	Supported	cbx_1.TextColor = long(88995) cbx_1.TextColor = RGB(0,0,255)
TextSize	Supported	cbx_1.TextSize = 15 cbx_1.TextSize = li_value
ThirdState	Supported	cbx_1.ThreeState = TRUE cbx_1.ThirdState = TRUE
ThreeState	Supported	cbx_1.ThreeState = TRUE cbx_1.ThirdState = TRUE
Underline	Supported	cbx_1.Underline = TRUE

Property	Support Level	Example Code
		cbx_1.Underline = lb_value
Visible	Supported	cbx_1.Visible =TRUE cbx_1.Visible = lb_value
Weight	Supported	cbx_1.Weight = 700 cbx_1.Weight = li_value In both PowerBuilder and Appeon Web applications, a weight value smaller or equal to 550 indicates a normal weight, and a weight value larger than 550 indicates a bold weight.
Width	Supported	cbx_1.Width=899 cbx_1.Width = li_value
X	Supported	cbx_1.X = 100 cbx_1.X = li_value
Y	Supported	cbx_1.Y = 500 cbx_1.Y = li_value
ClassDefinition	Unsupported	

6.1.1.1.2 Events for CheckBox control

Table 6.3:

Event	Support Level	Example Code
Clicked	Supported	Clicked()
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LoseFocus	Supported	LoseFocus()
RButtonDown	Supported	RbuttonDown(flags, xpos, ypos)
Help	Unsupported	
Other	Unsupported	

6.1.1.1.3 Functions for CheckBox control

Table 6.4:

Function	Support Level	Example Code
ClassName	Supported	ls_return = cbx_1.ClassName()

Function	Support Level	Example Code
Drag	Supported	cbx_1.Drag(Begin!)
GetParent	Supported	lobj_parent = cbx_1.GetParent()
Hide	Supported	li_return = cbx_1.Hide()
Move	Supported	cbx_1.Move(100,100)] li_return = cbx_1.Move(li_x,li_y)
PointerX	Supported	li_return = cbx_1.PointerX()
PointerY	Supported	li_return = cbx_1.PointerY()
PostEvent	Supported	cbx_1.PostEvent(RButtonDown!) lb_return = cbx_1.PostEvent(event)
Resize	Supported	cbx_1.Resize(100,20) li_return = cbx_1.Resize(li_x, li_y)
SetFocus	Supported	cbx_1.SetFocus() li_return = cbx_1.SetFocus() Unsupported: SetFocus(cbx_1)
SetPosition	Supported	li_return = cbx_1.SetPosition(position)
SetRedraw	Supported	li_return= cbx_1.SetRedraw(TRUE)
Show	Supported	li_return = cbx_1.Show()
TriggerEvent	Supported	cbx_1.TriggerEvent(RButtonDown!)

TypeOf	Supported	if cbx_1.TypeOf() = Checkbox! Then
		ls_return = "Checkbox!"
		else
		ls_return = "Not a Checkbox"
		end if

GetContextService	Unsupported	
Print	Unsupported	

6.1.1.2 CommandButton control

6.1.1.2.1 Properties for CommandButton control

Table 6.5:

Property	Support Level	Example Code
BringToTop	Supported	cb_1.BringToTop = TRUE
Cancel	Supported	cb_1.Cancel = TRUE
Default	Supported	cb_1.Default = TRUE
DragAuto	Supported	cb_1.DragAuto = TRUE

Property	Support Level	Example Code
DragIcon	Supported	cb_1.DragIcon = 'c:\archive\arrow.ico'
Enabled	Supported	cb_1.Enabled = lb_value
FaceName	Supported	cb_1.FaceName = "" cb_1.FaceName = "Courier"
Height	Supported	cb_1.Height = li_value
FontCharSet	Supported	cb_1.FontCharSet = ANSI!
FontFamily	Supported	cb_1.FontFamily = Roman!
FontPitch	Supported	cb_1.FontPitch = Fixed!
Italic	Supported	cb_1.Italic = lb_value
Pointer	Supported	cb_1.Pointer = 'Size!' cb_1.Pointer ='d:\archive\IBEAM.BMP.cur'
TabOrder	Supported	cb_1.TabOrder = 20
Tag	Supported	cb_1.Tag = ls_value
Text	Supported	cb_1.Text = ls_value
TextSize	Supported	cb_1.TextSize = li_value
Underline	Supported	cb_1.Underline = TRUE cb_1.Underline = lb_value
Visible	Supported	cb_1.Visible = TRUE cb_1.Visible = lb_value
Weight	Supported	cb_1.Weight = 700 cb_1.Weight = li_value
Width	Supported	cb_1.Width = 750 cb_1.Width = li_value
X	Supported	cb_1.X = 280 cb_1.X = li_value
Y	Supported	cb_1.Y = 1280 cb_1.Y = li_value
ClassDefinition	Unsupported	

6.1.1.2.2 Events for CommandButton control

Table 6.6:

Event	Support Level	Example Code
Clicked	Supported	Clicked()
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DragDrop	Supported	DragDrop(source)

Event	Support Level	Example Code
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LoseFocus	Supported	LoseFocus()
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
Help	Unsupported	
Other	Unsupported	

6.1.1.2.3 Functions for CommandButton control

Table 6.7:

Function	Support Level	Example Code
ClassName	Supported	ls_return = cb_1.ClassName()
Drag	Supported	cb_1.Drag(Begin!)
GetParent	Supported	PowerObject lpo_return lpo_return = cb_1.GetParent()
Hide	Supported	li_return = cb_1.Hide()
Move	Supported	cb_1.Move(200,800) li_return = cb_1.Move(li_x,li_y)
PointerX	Supported	li_return = cb_1.PointerX()
PointerY	Supported	li_return = cb_1.PointerY()
PostEvent	Supported	cb_1.PostEvent(Clicked!)
Resize	Supported	cb_1.Resize(200,800) li_return = cb_1.Resize(li_x, li_y)
SetFocus	Supported	li_return = cb_1.SetFocus() Unsupported: SetFocus(cb_1)
SetPosition	Supported	ll_return = cb_1.SetPosition(ToTop!)
SetRedraw	Supported	li_return = cb_1.SetRedraw(TRUE)
Show	Supported	li_return = cb_1.Show()
TriggerEvent	Supported	li_return = cb_1.TriggerEvent(Clicked!)

TypeOf	Supported	if cb_1.TypeOf() = commandbutton! then
		ls_return = "commandbutton!"
		end if

GetContextService	Unsupported	
Print	Unsupported	

6.1.1.3 DatePicker control

Difference

Left mouse clicking on the DatePicker control will trigger the Clicked, GetFocused events in sequence.

6.1.1.3.1 Properties for DatePicker control

Table 6.8:

Property	Support Level	Example Code
Accelerator	Supported	dp_1.Accelerator=67
Accessible Description	Supported	
AccessibleName	Supported	
AccessibleRole	Supported	
AllowEdit	Supported	dp_1.AllowEdit = TRUE
Border	Supported	dp_1.Border = TRUE
BorderStyle	Supported	dp_1.BorderStyle = StyleBox! dp_1.BorderStyle = StyleLowered! dp_1.BorderStyle = StyleRaised! Note: StyleShadowBox! is unsupported.
BringToTop	Supported	
CalendarBack Color	Supported	dp_1.CalendarBackColor = RGB(255, 255, 0)
CalendarFont Charset	Supported	db_1.CalendarFontCharset = ANSI!
CalendarFont Family	Supported	dp_1.CalendarFontFamily = Roman!
CalendarFont Name	Supported	dp_1.CalendarFontName = ls_value
CalendarFont Pitch	Supported	dp_1.CalendarFontPitch = Fixed!
CalendarFont Weight	Supported	dp_1.CalendarFontWeight = li_value
CalendarItalic	Supported	dp_1.CalendarItalic = lb_value
CalendarText Color	Supported	dp_1.CalendarTextColor = RGB(0,0,255)
CalendarTextSize	Supported	dp_1.CalendarTextSize = li_value
CalendarTitle BackColor	Supported	dp_1.CalendarTitleBackColor = RGB(128, 255, 128)
CalendarTrailing TextColor	Supported	dp_1.CalendarTrailingTextColor = RGB(128, 255, 128)

Property	Support Level	Example Code
CalendarUnderline	Supported	dp_1.CalendarUnderline = lb_value
CustomFormat	Supported	dp_1.CustomFormat = ls_value
DateValue	Supported	dp_1.DateValue = ld_value
DragAuto	Supported	dp_1.DragAuto = TRUE Notes: 1. On the Web, dragging a DatePicker control will not trigger the DropDown event if the DragAuto property is set to True, which means there is no Month Calender displayed on the Web when you Drag the DatePicker with setting the DragAuto property to True. 2. On the Web, if the DragAuto is set to True, clicking, dragging, and releasing the DatePicker control will trigger the following event in sequence: DragEnter, DragWithin, DragLeave.
DragIcon	Supported	dp_1.DragIcon = 'arrow.ico'
DropDownRight	Supported	dp_1.DropDownRight = true
Enabled	Supported	dp_1.Enabled = TRUE
FaceName	Supported	dp_1.FaceName = ""
FirstDayOfWeek	Supported	dp_1.FirstDayOfWeek = Monday!
FontCharSet	Supported	dp_1.FontCharSet = ANSI!
FontFamily	Supported	dp_1.FontFamily = Roman!
FontPitch	Supported	dp_1.FontPitch = Fixed!
FontWeight	Supported	dp_1.FontWeight = 700
Format	Supported	dp_1.Format = dtfShortDate!
Height	Supported	dp_1.Height = 750 dp_1.Height = li_value
Italic	Supported	dp_1.Italic=TRUE dp_1.Italic = lb_value
MaxDate	Supported	dp_1.MaxDate = Date("2007/06/30")
MinDate	Supported	dp_1.MinDate = Date("2006/09/01")
Pointer	Supported	dp_1.Pointer = 'Size!'
ShowUpDown	Supported	dp_1.ShowUpDown = true
TabOrder	Supported	dp_1.TabOrder = 30
Tag	Supported	dp_1.Tag = ls_value
Text	Supported	dp_1.Text = ls_value

Property	Support Level	Example Code
TextSize	Supported	dp_1.TextSize = li_value
TimeValue	Supported	lt_value = dp_1.TimeValue
TodayCircle	Supported	dp_1.TodayCircle = false
TodaySection	Supported	dp_1.TodaySection = false
Underline	Supported	dp_1.Underline = TRUE
Value	Supported	dp_1.Value = DateTime(Date("2005/07/01"),Time("12:00:00"))
Visible	Supported	dp_1.Visible = True
WeekNumbers	Supported	dp_1.WeekNumbers = true
Width	Supported	dp_1.Width = li_value Note: 1. If the Width value is set to -32768, obtaining the width property returns -32768 on the Web, however it returns 0 in PowerBuilder. 2. If the Width value is set to 32768, obtaining the width property returns 0 on the Web, however it returns -32768 in PowerBuilder.
X	Supported	dp_1.X = li_value Note: 1. If the X value is set to -32768, obtaining the X property returns -32768 on the Web, however it returns 0 in PowerBuilder. 2. If the Width value is set to 32768, obtaining the width property returns 0 on the Web, however it returns -32768 in PowerBuilder.
Y	Supported	dp_1.Y = li_value Note: 1. If the Y value is set to -32768, obtaining the Y property returns -32768 on the Web, however it returns 0 in PowerBuilder. 2. If the Width value is set to 32768, obtaining the width property returns 0 on the Web, however it returns -32768 in PowerBuilder.
ClassDefinition	Unsupported	
RightToLeft	Unsupported	

6.1.1.3.2 Events for DatePicker control

Table 6.9:

Event	Support Level	Example Code
Clicked	Supported	Clicked()
CloseUp	Supported	CloseUp()
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DoubleClicked	Supported	Doubleclicked(flags,xpos,ypos)
DragDrop	Supported	DragDrop()
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
DropDown	Supported	DropDown()
LoseFocus	Supported	GetFocus()
GetFocus	Supported	GetFocus()
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
ValueChanged	Supported	Valuechanged(flag,dtm)
Help	Unsupported	
PreCreateWindow	Unsupported	
UserString	Unsupported	
Other	Unsupported	

6.1.1.3.3 Functions for DatePicker Control

Table 6.10:

Function	Support Level	Example Code
ClassName	Supported	ls_value = dp_1.ClassName()
Drag	Supported	dp_1.Drag(dragmode) Note: If you use Drag(Begin!), single mouse clicking will trigger DragEnter and DragDrop event in sequence on the Web. This is different from the event sequences in PowerBuilder.
GetCalendar	Supported	dp_1.getcalendar( )
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = dp_1.GetParent()
GetText	Supported	ls_text = dp_1.GetText()
GetToday	Supported	currentdate = dp_1.GetToday()
GetValue	Supported	li_value = dp_1.GetValue(dt)
Hide	Supported	li_return = dp_1.Hide()

Function	Support Level	Example Code
Move	Supported	li_return = dp_1.Move(li_x,li_y)
PointerX	Supported	li_value = dp_1.PointerX()
PointerY	Supported	li_value = dp_1.PointerY()
PostEvent	Supported	dp_1.PostEvent(Constructor!)
Resize	Supported	li_return = dp_1.Resize(li_width,li_height)
SetFocus	Supported	li_return = dp_1.SetFocus()
SetPosition	Supported	li_return = dp_1.SetPosition(position)
SetRedraw	Supported	li_return= dp_1.SetRedraw(FALSE)
SetToday	Supported	li_return = dp_1.SetToday(currentdate)
SetValue	Supported	dp_1.SetValue(dt)
Show	Supported	li_return = dp_1.Show()
TriggerEvent	Supported	dp_1.TriggerEvent(Constructor!)

TypeOf	Supported	if dp_1.TypeOf() = DatePicker! Then
		ls_return = 'DatePicker!'
		else
		ls_return = 'isvalid'
		end if

GetContextService	Unsupported	
Print	Unsupported	

6.1.1.4 DropDownListBox control

Important Requirements

In PowerBuilder, if a DropDownListBox has no item, an empty row will display in the ListBox portion when the user clicks the down arrow. However, on the Web application, no empty row will display.

6.1.1.4.1 Properties for DropDownListBox control

Table 6.11:

Property	Support Level	Example Code
Accelerator	Supported	ddlb_1.Accelerator=67
AllowEdit	Supported	ddlb_1.AllowEdit = TRUE ddlb_1.AllowEdit = lb_value Note: The pbm_keydown event is unsupported if the AllowEdit property is set to true.
AutoHScroll	Supported	ddlb_1.AutoHScroll = TRUE
BringToTop	Supported	ddlb_1.BringToTop = TRUE
BackColor	Supported	ddlb_1.BackColor = RGB(0, 0, 255)

Property	Support Level	Example Code
		ddlb_1.BackColor = ll_value
Border	Supported	ddlb_1.Border = TRUE ddlb_1.Border = lb_value
BorderStyle	Supported	ddlb_1.BorderStyle = StyleBox! ddlb_1.BorderStyle = StyleLowered! ddlb_1.BorderStyle = StyleRaised!
DragAuto	Supported	ddlb_1.DragAuto = TRUE
DragIcon	Supported	ddlb_1.DragIcon = 'c:\archive\arrow.ico'
Enabled	Supported	ddlb_1.Enabled=TRUE ddlb_1.Enabled = lb_value
FaceName	Supported	ddlb_1.FaceName = "" ddlb_1.FaceName = ls_value
FontCharSet	Supported	ddlb_1.FontCharSet = ANSI!
FontFamily	Supported	ddlb_1.FontFamily = Roman!
FontPitch	Supported	ddlb_1.FontPitch = Fixed!
Height	Supported	ddlb_1.Height = 750 ddlb_1.Height = li_value
HScrollBar	Supported	ddlb_1.HScrollBar = TRUE ddlb_1.HscrollBar = lb_value
Italic	Supported	ddlb_1.Italic=TRUE ddlb_1.Italic = lb_value
Item[ ]	Supported	ls_value = ddlb_1.Item[1]
Limit	Supported	ddlb_1.Limit = 256 ddlb_1.Limit = li_value
Pointer	Supported	ddlb_1.Pointer = 'Size!' ddlb_1.Pointer ='d:\archive\IBEAM.BMP.cur'
RightToLeft	Supported	ddlb_1.RightToLeft = TRUE
ShowList	Supported	ddlb_1.ShowList = TRUE
Sorted	Supported	ddlb_1.Sorted = TRUE ddlb_1.Sorted = lb_value
TabOrder	Supported	ddlb_1.TabOrder = 30
Tag	Supported	ddlb_1.Tag = ls_value
Text	Supported	ddlb_1.Text = ls_value
TextColor	Supported	ddlb_1.TextColor = ll_value ddlb_1.TextColor = RGB(192,192,192)

Property	Support Level	Example Code
TextSize	Supported	ddlb_1.TextSize = li_value
Underline	Supported	ddlb_1.Underline= TRUE ddlb_1.Underline = lb_value
Visible	Supported	ddlb_1.Visible = TRUE ddlb_1.Visible = lb_value
VScrollBar	Supported	ddlb_1.VScrollBar = TRUE ddlb_1.VscrollBar = lb_value
Weight	Supported	ddlb_1.Weight = 700 ddlb_1.Weight = li_value
Width	Supported	ddlb_1.Width = li_value
X	Supported	ddlb_1.X = li_value
Y	Supported	ddlb_1.Y = li_value
ClassDefinition	Unsupported	

6.1.1.4.2 Events for DropDownListBox control

Table 6.12:

Event	Support Level	Example Code
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DoubleClicked	Supported	DoubleClicked()
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LoseFocus	Supported	LoseFocus()
Modified	Supported	Modified() Note: it is unsupported to trigger the event by clicking the Enter key.
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
SelectionChanged	Supported	SelectionChanged(index) If the return value of the event is greater than 0, the Modified event will be triggered. Otherwise, continue processing.
Help	Unsupported	
Other	Unsupported	

6.1.1.4.3 Functions for DropDownListBox control

Table 6.13:

Function	Support Level	Example Code
AddItem	Supported	li_return = ddlb_1.AddItem(item)
ClassName	Supported	ls_return = ddlb_1.ClassName()
Clear	Supported	li_return = ddlb_1.Clear()
Copy	Supported	li_return = ddlb_1.Copy()
Cut	Supported	li_return = ddlb_1.Cut()
DeleteItem	Supported	li_return = ddlb_1.DeleteItem(index)
DirList	Supported	ddlb_1.DirList("C:\EMPLOYEE*.TXT", 0)
DirSelect	Supported	lb_dir = ddlb_1.DirSelect(ls_dirname)
Drag	Supported	ddlb_1.Drag(Cancel!)
FindItem	Supported	li_return = ddlb_1.FindItem('hello', 4) li_return = ddlb_1.FindItem(text, index)
GetParent	Supported	PowerObject lpo_return lpo_return = ddlb_1.GetParent()
Hide	Supported	li_return = ddlb_1.Hide()
InsertItem	Supported	ddlb_1.InsertItem('hello',3) li_return = ddlb_1.InsertItem(item, index)
Move	Supported	ddlb_1.Move(200,400) li_return = ddlb_1.Move(li_x,li_ y)
Paste	Supported	li_return = ddlb_1.Paste()
PointerX	Supported	li_return = ddlb_1.PointerX()
PointerY	Supported	li_return = ddlb_1.PointerY()
PostEvent	Supported	ddlb_1.PostEvent(Clicked!)
ReplaceText	Supported	ddlb_1.ReplaceText("60 Days")
Reset	Supported	li_return = ddlb_1.Reset()
Resize	Supported	ddlb_1.Resize(200,800) li_return = ddlb_1.Resize(li_x, li_y)
SelectItem	Supported	ddlb_1.SelectItem ('hello',3) li_return = ddlb_1.SelectItem(item, index)
SelectedLength	Supported	li_returnvalue = ddlb_1.SelectedLength()
SelectedStart	Supported	li_returnvalue = ddlb_1.SelectedStart()
SelectedText	Supported	li_returnvalue = ddlb_1.SelectedText()
SelectText	Supported	li_returnvalue = ddlb_1.SelectText(start, length)
SetFocus	Supported	li_return = ddlb_1.SetFocus()

Function	Support Level	Example Code
		Unsupported: SetFocus(ddlb_1)
SetPosition	Supported	ddlb_1.SetPosition(ToTop!) li_return = ddlb_1.SetPosition(position)
SetRedraw	Supported	li_return = ddlb_1.SetRedraw(FALSE)
Show	Supported	li_return = ddlb_1.Show()
Text	Supported	ls_return = ddlb_1.Text(index)
TotalItems	Supported	li_return = ddlb_1.TotalItems()
TriggerEvent	Supported	ddlb_1.TriggerEvent(Constructor!) li_return = ddlb_1.TriggerEvent(event)
TypeOf	Supported	if ddlb_1.typeof() = DropDownListBox! then messagebox("show"," DropDownListBox!") end if
GetContextService	Unsupported	
Position	Unsupported	
Print	Unsupported	

6.1.1.5 DropDownPictureListBox control

6.1.1.5.1 Properties for DropDownPictureListBox control

Table 6.14:

Property	Support Level	Example Code
Accelerator	Supported	ddplb_1.Accelerator = 67
AllowEdit	Supported	ddplb_1.AllowEdit = TRUE ddplb_1.AllowEdit = lb_value Note: The pbm_keydown event is unsupported if the AllowEdit property is set to true.
AutoHScroll	Supported	ddplb_1.AutoHScroll = TRUE
BackColor	Supported	ddplb_1.BackColor = ll_value ddplb_1.BackColor = RGB (0,0,255)
BringToTop	Supported	ddplb_1.BringToTop = TRUE
Border	Supported	ddplb_1.Border = TRUE ddplb_1.Border = lb_value
BorderStyle	Supported	ddplb_1.BorderStyle = StyleBox! ddplb_1.BorderStyle = StyleLowered! ddplb_1.BorderStyle = StyleRaised! Note: StyleShadowBox! is unsupported.

Property	Support Level	Example Code
DragAuto	Supported	ddplb_1.DragAuto = TRUE
DragIcon	Supported	ddplb_1.DragIcon = 'c:\archive\arrow.ico'
Enabled	Supported	ddplb_1.Enabled = TRUE ddplb_1.Enabled = lb_value
FaceName	Supported	ddplb_1.Facename = ls_value
FontCharSet	Supported	ddplb_1.FontCharSet = ANSI!
FontFamily	Supported	ddplb_1.FontFamily = Roman!
FontPitch	Supported	ddplb_1.FontPitch = Fixed!
Height	Supported	ddplb_1.Height = li_value
HScrollBar	Supported	ddplb_1.Hscrollbar = lb_value
Italic	Supported	ddplb_1.Italic = lb_value
Item[ ]	Supported	String a[] = ddplb_1.Item[]
ItemPictureIndex[ ]	Supported	ddplb_1.ItemPictureIndex[] = li_value[]
Limit	Supported	ddplb_1.Limit = 256 ddplb_1.Limit = li_value
PictureHeight	Supported	ddplb_1.PictureHeight = 16
PictureWidth	Supported	ddplb_1.PictureWidth = 16

PictureName[ ]	Supported	ls_value = ddplb_1.PictureName[1]
		Assigning values to part of an array and using the default values for the other part is not supported.
		For example:
		picturename[1] = "aaa.bmp"
		picturename[3] = "bbb.bmp"
		var pic1 = picturename[1]; var pic2 = picturename[3];
		A supported example is given as follows:
		picturename[1] = "aaa.bmp"
		picturename[2] = "bbb.bmp"
		picturename[3] = "ccc.bmp"
		var pic1 = picturename[1]; var pic2 = picturename[2]; var pic3 = picturename[3];

Pointer	Supported	ddplb_1.Pointer = "Beam!" dbplb_1.Pointer = 'd:\archive\IBEAM.BMP.cur'
RightToLeft	Supported	dbplb_1.RightToLeft = TRUE
ShowList	Supported	ddplb_1.ShowList = TRUE
Sorted	Supported	ddplb_1.Sorted = lb_value

Property	Support Level	Example Code
TabOrder	Supported	ddplb_1.TabOrder = 40
Tag	Supported	ddplb_1.Tag = ls_value
Text	Supported	ddplb_1.Text = ls_value
TextColor	Supported	ddplb_1.Textcolor = ll_value ddplb_1.Textcolor = RGB(192,192,192)
TextSize	Supported	ddplb_1.TextSize = li_value
Underline	Supported	ddplb_1.Underline = lb_value
Visible	Supported	ddplb_1.Visible = lb_value
VScrollBar	Supported	ddplb_1.VscrollBar = lb_value
Weight	Supported	ddplb_1.Weight = li_value
Width	Supported	ddplb_1.Width = li_value
X	Supported	ddplb_1.X = li_value
Y	Supported	ddplb_1.Y = li_value
ClassDefinition	Unsupported	
PictureMaskColor	Unsupported	

6.1.1.5.2 Events for DropDownPictureListBox control

Table 6.15:

Event	Support Level	Example Code
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DoubleClicked	Supported	DoubleClicked()
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LoseFocus	Supported	LoseFocus()
Modified	Supported	Modified()
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
SelectionChanged	Supported	SelectionChanged(index) //If the return value is greater than 0, the Modified event will be triggered. Otherwise, continue processing.
Help	Unsupported	
Other	Unsupported	

6.1.1.5.3 Functions for DropDownPictureListBox control

Table 6.16:

Function	Support Level	Example Code
AddItem	Supported	li_return = ddplb_1.AddItem(ls_name, li_pic)
AddPicture	Supported	li_return = ddplb_1.AddPicture(picturename)
ClassName	Supported	ls_return = ddplb_1.ClassName()
Clear	Supported	ls_return = ddplb_1.Clear()
Copy	Supported	ls_return = ddplb_1.Copy()
Cut	Supported	ls_return = ddplb_1.Cut()
DeleteItem	Supported	li_return = ddplb_1.DeleteItem(index)
DeletePicture	Supported	li_return =ddplb_1.DeletePicture(index)
DeletePictures	Supported	li_return =ddplb_1.DeletePictures()
DirList	Supported	ddplb_1.DirList("C:\EMPLOYEE*.TXT", 0) ddplb_1.DirList("C:\EMPLOYEE*.DOC", 1, st_path)
DirSelect	Supported	ddplb_1.DirSelect(ls_filename)
Drag	Supported	ddplb_1.Drag(End!)
FindItem	Supported	li_return = ddplb_1.FindItem(text, index)
GetParent	Supported	PowerObject lpo_return lpo_return = ddplb_1.GetParent()
Hide	Supported	li_return = ddplb_1.Hide()
InsertItem	Supported	ddplb_1.InsertItem("Run Application", 5)
Move	Supported	li_return = ddplb_1.Move(li_x,li_ y)
Paste	Supported	li_return = ddplb_1.Paste()
PointerX	Supported	li_return = ddplb_1.PointerX()
PointerY	Supported	li_return = ddplb_1.PointerY()
PostEvent	Supported	ddplb_1.PostEvent(Clicked!) lb_return = ddplb_1.PostEvent(event)
Reset	Supported	li_return = ddplb_1.Reset()
Resize	Supported	li_return = ddplb_1.Resize(li_width,li_height)
SelectItem	Supported	li_return = ddplb_1.SelectItem(item, index)
SelectedLength	Supported	li_returnvalue = ddplb_1.SelectedLength()
SelectedStart	Supported	li_returnvalue = ddplb_1.SelectedStart()
SelectedText	Supported	li_returnvalue = ddplb_1.SelectedText()
SelectText	Supported	li_returnvalue = ddplb_1.SelectText(start, length)
SetFocus	Supported	li_return = ddplb_1.SetFocus() Unsupported: SetFocus(ddplb_1)

Function	Support Level	Example Code
SetPosition	Supported	li_return = ddplb_1.SetPosition(position)
SetRedraw	Supported	li_return= ddplb_1.SetRedraw(FALSE)
Show	Supported	li_return = ddplb_1.Show()
Text	Supported	ls_return = ddplb_1.Text(index)
TotalItems	Supported	li_return = ddplb_1.TotalItems()
TriggerEvent	Supported	li_return = ddplb_1.TriggerEvent(event)
TypeOf	Supported	if ddplb_1.TypeOf() = DropDownPictureListBox! Then ls_return = 'DropDownPictureListBox!' else ls_return = 'isvalid' end if
GetContextService	Unsupported	
Position	Unsupported	
Print	Unsupported	

6.1.1.6 EditMask control

6.1.1.6.1 Properties for EditMask control

Table 6.17:

Property	Support Level	Example Code
Accelerator	Supported	em_1.Accelerator = 67
Alignment	Supported	em_1.Alignment = Center!
AutoSkip	Supported	em_1.AutoSkip = TRUE
BackColor	Supported	em_1.BackColor = ll_value em_1.BackColor = RGB(192,192,192)
Border	Supported	em_1.Border = TRUE em_1.Border = lb_value
BorderStyle	Supported	em_1.BorderStyle = StyleLowered! em_1.BorderStyle = StyleRaised! em_1.BorderStyle = StyleBox! Note: StyleShadowBox! is unsupported.
BringToTop	Supported	em_1.BringToTop = TRUE
DisplayData	Supported	
DisplayOnly	Supported	em_1.DisplayOnly = TRUE em_1.DisplayOnly = lb_value
DragAuto	Supported	em_1.DragAuto = TRUE
DropDownCalendar	Supported	

Property	Support Level	Example Code
DragIcon	Supported	em_1.DragIcon = 'c:\archive\arrow.ico"
Enabled	Supported	em_1.Enabled = TRUE em_1.Enabled = lb_value
FaceName	Supported	em_1.FaceName = ls_value
FontCharSet	Supported	em_1.FontCharSet = ANSI!
FontFamily	Supported	em_1.FontFamily = Roman!
FontPitch	Supported	em_1.FontPitch = Fixed!
Height	Supported	em_1.Height = li_value
HideSelection	Supported	em_1.HideSelection = FALSE
Increment	Supported	em_1.Increment = 5.0 em_1.Increment = ld_value
Italic	Supported	em_1.Italic = lb_value
Limit	Supported	em_1.Limit = 50 em_1.Limit = li_value Note: the Limit property cannot be dynamically changed.
Mask	Supported	em_1.MaskDataType =StringMask em_1.Mask = "^^^-!!!" Note: The Mask property for EditMask can only be set in the PowerBuilder Painter.
MaskDataType	Supported	em_1.MaskDataType = StringMask! em_1.MaskDataType = DateTimeMask! em_1.MaskDataType = NumericMask! Note: this property can be set only in painter.
MinMax	Supported	em_1.MinMax = ("100 ~~ 10000")
Pointer	Supported	em_1.Pointer = 'AppStarting!' em_1.Pointer = 'd:\archive\IBEAM.BMP.cur'
RightToLeft	Supported	em_1.RightToLeft = TRUE
Spin	Supported	em_1.Spin = TRUE em_1.Spin = FALSE
TabOrder	Supported	em_1.TabOrder = 50
Tag	Supported	em_1.Tag = ls_value
Text	Supported	em_1.Text = ls_value
TextColor	Supported	em_1.TextColor = ll_value em_1.TextColor = RGB(192,192,192)
TextCase	Supported	em_1.TextCase = AnyCase!

Property	Support Level	Example Code
		em_1.TextCase = Lower! em_1.TextCase = Upper!
TextSize	Supported	em_1.TextSize = li_value
Underline	Supported	em_1.Underline = TRUE em_1.Underline = lb_value
UseCodeTable	Supported	em_1.UseCodeTable = TRUE
Visible	Supported	em_1.Visible = lb_value
Weight	Supported	em_1.Weight = li_value
Width	Supported	em_1.Width = li_value
X	Supported	em_1.X = li_value
Y	Supported	em_1.Y = li_value
AutoHScroll	Supported	
AutoVScroll	Supported	
ClassDefinition	Unsupported	
HScrollBar	Supported	
IgnoreDefaultButton	Unsupported	
TabStop[ ]	Unsupported	
VScrollBar	Supported	

6.1.1.6.2 Events for EditMask control

Table 6.18:

Event	Support Level	Example Code
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LoseFocus	Supported	LoseFocus()
Modified	Supported	Modified()
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
Help	Unsupported	
Other	Unsupported	

6.1.1.6.3 Functions for EditMask control

Table 6.19:

Function	Support Level	Example Code
ClassName	Supported	ls_returnvalue = em_1.ClassName()
Clear	Supported	li_returnvalue = em_1.Clear()
Copy	Supported	li_returnvalue = em_1.Copy()
Cut	Supported	li_returnvalue = em_1.Cut()
Drag	Supported	em_1.Drag(Begin!)
GetData	Supported	li_returnvalue = em_1.GetData(datavariable)
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = em_1.GetParent()
Hide	Supported	li_returnvalue = em_1.Hide()
LineCount	Supported	li_returnvalue = em_1.LineCount()
LineLength	Supported	li_returnvalue = em_1.LineLength()
Move	Supported	li_returnvalue = em_1.Move(li_xpos, li_ypos)
Paste	Supported	li_returnvalue = em_1.Paste()
PointerX	Supported	li_returnvalue = em_1.PointerX()
PointerY	Supported	li_returnvalue = em_1.PointerY()
PostEvent	Supported	lb_returnvalue = em_1.PostEvent(event)
ReplaceText	Supported	li_returnvalue = em_1.ReplaceText(text)
Resize	Supported	li_returnvalue = em_1.Resize(li_width, li_height)
SelectedLength	Supported	li_returnvalue = em_1.SelectedLength() If the control is off screen, the execution of the function may have an incorrect return result.
SelectedLine	Supported	integer li_SL li_SL = em_1.SelectedLine()
SelectedStart	Supported	li_returnvalue = em_1.SelectedStart() If the control is off screen, the execution of the function may have an incorrect return result.
SelectedText	Supported	ls_returnvalue = em_1.SelectedText() If the control is off screen, the execution of the function may have an incorrect return result.
SelectText	Supported	ls_returnvalue = em_1.SelectText(start, length) If the control is off screen, the execution of the function may have an incorrect return result.
SetFocus	Supported	li_returnvalue = em_1.SetFocus() Unsupported: SetFocus(em_test)

Function	Support Level	Example Code
SetMask	Supported	li_returnvalue = em_1.SetMask(maskdatatype, mask)
SetPosition	Supported	li_returnvalue = em_1.SetPosition(position)
SetRedraw	Supported	li_returnvalue = em_1.SetRedraw (FALSE)
Show	Supported	li_returnvalue = em_1.Show()
TextLine	Supported	li_returnvalue = em_1.TextLine()
TriggerEvent	Supported	li_returnvalue = em_1.TriggerEvent()

TypeOf	Supported	If em_1.TypeOf() = editmask! Then
		ls_returnvalue = 'editmask!'
		else
		ls_returnvalue = 'invalid!'
		End if

CanUndo	Unsupported	
GetContextService	Unsupported	
Position	Unsupported	
Print	Unsupported	
Scroll	Unsupported	
Undo	Unsupported	

6.1.1.7 GroupBox control

6.1.1.7.1 Properties for GroupBox control

Table 6.20:

Property	Support Level	Example Code
BackColor	Supported	gb_1.BackColor = ll_value
BorderStyle	Supported	gb_1.Border = TRUE gb_1.BorderStyle = StyleLowered!
BringToTop	Supported	gb_1.BringToTop = TRUE
DragAuto	Supported	gb_1.DragAuto = TRUE
DragIcon	Supported	gb_1.DragIcon = 'c:\archive\arrow.ico"
Enabled	Supported	gb_1.Enabled = lb_value
FaceName	Supported	gb_1.FaceName = ls_value
FontCharSet	Supported	gb_1.FontCharSet = ANSI!
FontFamily	Supported	gb_1.FontFamily = Roman!
FontPitch	Supported	gb_1.FontPitch = Fixed!
Height	Supported	gb_1.Height = li_value
Italic	Supported	gb_1.Italic = lb_value

Property	Support Level	Example Code
Pointer	Supported	gb_1.Pointer = 'Beam!' gb_1.Pointer = 'd:\archive\IBEAM.BMP.cur'
RightToLeft	Supported	gb_1.RightToLeft = TRUE
TabOrder	Supported	gb_1.TabOrder = 50
Tag	Supported	gb_1.Tag = ls_value
Text	Supported	gb_1.Text = ls_value
TextColor	Supported	gb_1.TextColor = ll_value gb_1.TextColor = RGB(192,192,192)
TextSize	Supported	gb_1.TextSize = li_value
Underline	Supported	gb_1.Underline = lb_value
Visible	Supported	gb_1.Visible = lb_value
Weight	Supported	gb_1.Weight = li_value
Width	Supported	gb_1.Width = li_value
X	Supported	gb_1.X = li_value
Y	Supported	gb_1.Y = li_value
ClassDefinition	Unsupported	

6.1.1.7.2 Events for GroupBox control

Table 6.21:

Event	Support Level	Example Code
Constructor	Supported	Constructor( )
Destructor	Supported	Destructor( )
Help	Unsupported	
Other	Unsupported	

6.1.1.7.3 Functions for GroupBox control

Table 6.22:

Function	Support Level	Example Code
ClassName	Supported	ls_returnvalue = gb_1.ClassName()
Drag	Supported	gb_1.drap(Cancel!)
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = gb_1.GetParent()
Hide	Supported	li_returnvalue = gb_1.Hide()
Move	Supported	gb_1.Move(200,500) li_returnvalue = gb_1.Move(li_xpos, li_ypos)
PointerX	Supported	li_returnvalue = gb_1.PointerX()

Function	Support Level	Example Code
PointerY	Supported	li_returnvalue = gb_1.PointerY()
PostEvent	Supported	lb_returnvalue = gb_1.PostEvent(event)
Resize	Supported	li_returnvalue = gb_1.Resize(li_width,li_height)
SetFocus	Supported	li_returnvalue = gb_1.SetFocus( ) Unsupported: SetFocus(gb_text)
SetPosition	Supported	li_returnvalue = gb_1.SetPosition(position)
SetRedraw	Supported	li_value = gb_1.SetRedraw(TRUE)
Show	Supported	li_returnvalue = gb_1.Show()
TriggerEvent	Supported	li_returnvalue = gb_1.TriggerEvent(event)

TypeOf	Supported	if gb_1.TypeOf() = GroupBox! Then
		ls_returnvalue = "GroupBox!"
		else
		ls_returnvalue = "invalid"
		end if

GetContextService	Unsupported	
Print	Unsupported	

6.1.1.8 HProgressBar control

6.1.1.8.1 Properties for HProgressBar control

Table 6.23:

Property	Support Level	Example Code
BringToTop	Supported	hpb_1.BringToTop = TRUE
DragAuto	Supported	hpb_1.DragAuto = TRUE
DragIcon	Supported	hpb_1.DragIcon = 'Question!'
Height	Supported	hpb_1.Height = li_value
MaxPosition	Supported	hpb_1.MaxPosition = 120
MinPosition	Supported	hpb_1.MinPosition = 20
Pointer	Supported	hpb_1.Pointer = 'Beam!' hpb_1.Pointer ='d:\archive\IBEAM.BMP.cur'
Position	Supported	hpb_1.Position = 50
SetStep	Supported	hpb_1.setstep = 20
SmoothScroll	Supported	hpb_1.SmoothScroll = True
TabOrder	Supported	hpb_1.TabOrder = 20
Tag	Supported	hpb_1.Tag = "HProgressBar control"
Visible	Supported	hpb_1.Visible = TRUE

Property	Support Level	Example Code
Width	Supported	hpb_1.Width = 700
X	Supported	hpb_1.X = 280
Y	Supported	hpb_1.Y = 1280
ClassDefinition	Unsupported	

6.1.1.8.2 Events for HProgressBar control

Table 6.24:

Event	Support Level	Example Code
Clicked	Supported	Clicked()
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DoubleClicked	Supported	DoubleClicked()
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LoseFocus	Supported	LoseFocus()
RightClicked	Supported	RightClicked()
Help	Unsupported	
Other	Unsupported	

6.1.1.8.3 Functions for HProgressBar control

Table 6.25:

Function	Support Level	Example Code
ClassName	Supported	ls_classnm = hpb_1.classname()
Drag	Supported	li_return = hpb_1.drag(Begin!)
GetParent	Supported	lpo_return = hpb_1.GetParent()
Hide	Supported	li_return = hpb_1.Hide()
Move	Supported	li_return = hpb_1.Move(li_x,li_y)
OffsetPos	Supported	li_return = hpb_1.OffsetPos(10)
PointerX	Supported	li_return = hpb_1.PointerX()
PointerY	Supported	li_return = hpb_1.PointerY()
PostEvent	Supported	hpb_1.PostEvent(Clicked!)
Resize	Supported	li_return = hpb_1.Resize(li_x, li_y)
SetFocus	Supported	li_return = hpb_1.SetFocus()

Function	Support Level	Example Code
SetPosition	Supported	li_return = hpb_1.SetPosition(ToTop!)
SetRange	Supported	li_return = hpb_1.SetRange(1, 10)
SetRedraw	Supported	li_return = hpb_1.SetRedraw(FALSE)
Show	Supported	li_return = hpb_1.Show()
StepIt	Supported	li_return = hpb_1.StepIt( )
TriggerEvent	Supported	li_return = hpb_1.TriggerEvent(Clicked!)
TypeOf	Supported	type_obj = hpb_1.typeof()
GetContextService	Unsupported	
Print	Unsupported	

6.1.1.9 HScrollBar control

6.1.1.9.1 Properties for HScrollBar control

Table 6.26:

Property	Support Level	Example Code
BringToTop	Supported	hsb_1.BringToTop = TRUE
DragAuto	Supported	hsb_1.DragAuto = TRUE
DragIcon	Supported	hsb_1.DragIcon = 'Question!'
Height	Supported	hsb_1.Height = li_value
MaxPosition	Supported	hsb_1.MaxPosition = 120
MinPosition	Supported	hsb_1.MinPosition = 20
Pointer	Supported	hsb_1.Pointer = 'Beam!' hsb_1.Pointer ='d:\archive\IBEAM.BMP.cur'
Position	Supported	hsb_1.Position = 50
StdHeight	Supported	hsb_1.StdHeight = True
TabOrder	Supported	hsb_1.TabOrder = 20
Tag	Supported	hsb_1.Tag = "HScrollBar control"
Visible	Supported	hsb_1.Visible = TRUE
Width	Supported	hsb_1.Width = 700
X	Supported	hsb_1.X = 280
Y	Supported	hsb_1.Y = 1280
ClassDefinition	Unsupported	

6.1.1.9.2 Events for HScrollBar control

Table 6.27:

Event	Support Level	Example Code
Constructor	Supported	Constructor()

Event	Support Level	Example Code
Destructor	Supported	Destructor()
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LineLeft	Supported	LineLeft()
LineRight	Supported	LineRight() Note: User defined events in the LineLeft or LineRight events for the HScrollBar control may be triggered more than once.
LoseFocus	Supported	LoseFocus()
Moved	Supported	Moved(scrollpos)
PageLeft	Supported	PageLeft()
PageRight	Supported	PageRight()
RButtonDown	Supported	RButtonDown()
Help	Unsupported	
Other	Unsupported	

6.1.1.9.3 Functions for HScrollBar control

Table 6.28:

Function	Support Level	Example Code
ClassName	Supported	ls_classnm = hsb_1.classname()
Drag	Supported	li_return = hsb_1.drag(Begin!)
GetParent	Supported	lpo_return = hsb_1.GetParent()
Hide	Supported	li_return = hsb_1.Hide()
Move	Supported	li_return = hsb_1.Move(li_x,li_y)
PointerX	Supported	li_return = hsb_1.PointerX()
PointerY	Supported	li_return = hsb_1.PointerY()
PostEvent	Supported	hsb_1.PostEvent(Clicked!)
Resize	Supported	li_return = hsb_1.Resize(li_x, li_y)
SetFocus	Supported	li_return = hsb_1.SetFocus()
SetPosition	Supported	li_return = hsb_1.SetPosition(ToTop!)
SetRedraw	Supported	li_return = hsb_1.SetRedraw(FALSE)
Show	Supported	li_return = hsb_1.Show()
TriggerEvent	Supported	li_return = hsb_1.TriggerEvent(Clicked!)

Function	Support Level	Example Code
TypeOf	Supported	type_obj = hsb_1.typeof()
GetContextService	Unsupported	
Print	Unsupported	

6.1.1.10 HTrackBar control

6.1.1.10.1 Properties for HTrackBar control

Table 6.29:

Property	Support Level	Example Code
BringToTop	Supported	htb_1.BringToTop = TRUE
DragAuto	Supported	htb_1.DragAuto = TRUE
DragIcon	Supported	htb_1.DragIcon = 'Question!'
Height	Supported	htb_1.Height = li_value
LineSize	Supported	htb_1.LineSize = li_value
MaxPosition	Supported	htb_1.MaxPosition = 120
MinPosition	Supported	htb_1.MinPosition = 20
PageSize	Supported	htb_1.PageSize = li_value
Pointer	Supported	htb_1.Pointer = 'Beam!'
Position	Supported	htb_1.Position = 50
Slider	Supported	htb_1.Slider = TRUE
SliderSize	Supported	htb_1.SliderSize = li_value
TabOrder	Supported	htb_1.TabOrder = 20
Tag	Supported	htb_1.Tag = ls_tag
TickFrequency	Supported	htb_1.TickFrequency = li_value
TickMarks	Supported	htb_1.TickMarks = HTicksOnRight!
Visible	Supported	htb_1.Visible = TRUE
Width	Supported	htb_1.Width = li_value
X	Supported	htb_1.X = li_x
Y	Supported	htb_1.Y = li_y
ClassDefinition	Unsupported	

6.1.1.10.2 Events for HTrackBar control

Table 6.30:

Event	Support Level	Example Code
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()

Event	Support Level	Example Code
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LineDown	Supported	LineDown()
LineUp	Supported	LineUp()
LoseFocus	Supported	LoseFocus()
Moved	Supported	Moved(scrollpos)
PageDown	Supported	PageDown()
PageUp	Supported	PageUp()
RButtonDown	Supported	RButtonDown()
Help	Unsupported	
Other	Unsupported	

6.1.1.10.3 Functions for HTrackBar control

Table 6.31:

Function	Support Level	Example Code
ClassName	Supported	ls_classnm = htb_1.classname()
Drag	Supported	li_return = htb_1.drag(Begin!)
GetParent	Supported	lpo_return = htb_1.GetParent()
Hide	Supported	li_return = htb_1.Hide()
Move	Supported	li_return = htb_1.Move(li_x,li_y)
PointerX	Supported	li_return = htb_1.PointerX()
PointerY	Supported	li_return = htb_1.PointerY()
PostEvent	Supported	htb_1.PostEvent(Clicked!)
Resize	Supported	li_return = htb_1.Resize(li_x, li_y)
SelectionRange	Supported	li_return = htb_1.SelectionRange(30, 70)
SetFocus	Supported	li_return = htb_1.SetFocus()
SetPosition	Supported	li_return = htb_1.SetPosition(ToTop!)
SetRedraw	Supported	li_return = htb_1.SetRedraw(FALSE)
Show	Supported	li_return = htb_1.Show()
TriggerEvent	Supported	li_return = htb_1.TriggerEvent(Clicked!)
TypeOf	Supported	type_obj = htb_1.typeof()
GetContextService	Unsupported	
Print	Unsupported	

6.1.1.11 Line control

6.1.1.11.1 Properties for Line control

Table 6.32:

Property	Support Level	Example Code
BeginX	Supported	ln_1.BeginX = li_value
BeginY	Supported	ln_1.BeginY = li_value
EndX	Supported	ln_1.EndX = li_value
EndY	Supported	ln_1.EndY = li_value
LineColor	Supported	ln_1.LineColor = ll_value ln_1.LineColor = RGB(192,192,192)
LineStyle	Supported	ln_1.LineStyle = Continuous! ln_1.LineStyle = Dash! ln_1.LineStyle = DashDot! ln_1.LineStyle = DashDotDot! ln_1.LineStyle = Dot! ln_1.LineStyle = Transparent!
LineThickness	Supported	ln_1.LineThickness = li_value
Tag	Supported	ln_1.Tag = ls_value
Visible	Supported	ln_1.Visible = TRUE ln_1.Visible = lb_value
ClassDefinition	Unsupported	

6.1.1.11.2 Events for Line control

Table 6.33:

Event	Support Level	Example Code
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()

6.1.1.11.3 Functions for Line control

Table 6.34:

Function	Support Level	Example Code
ClassName	Supported	ls_returnvalue = ln_1.ClassName()
GetParent	Supported	li_returnvalue = ln_1.GetParent( )
Hide	Supported	li_returnvalue = ln_1.Hide( )
Move	Supported	li_returnvalue = ln_1.Move(li_xpos, li_ypos)
Resize	Supported	li_returnvalue = ln_1.Resize(100, 150)
Show	Supported	li_returnvalue = ln_1.Show()
TypeOf	Supported	if ln_1.typeof() = Line! Then

Function	Support Level	Example Code

		ls_returnvalue = 'Line!'
		else
		ls_returnvalue = 'Invalid'
		end if

GetContextService	Unsupported	

6.1.1.12 ListBox control

6.1.1.12.1 Properties for ListBox control

Table 6.35:

Property	Support Level	Example Code
Accelerator	Supported	lb_1.Accelerator=67
BackColor	Supported	lb_1.BackColor = ll_value lb_1.BackColor = RGB(192,192,192)
Border	Supported	lb_1.Border = lb_value
BringToTop	Supported	lb_1.BringToTop = TRUE
BorderStyle	Supported	lb_1.BorderStyle = StyleLowered! lb_1.BorderStyle = StyleRaised! lb_1.BorderStyle = StyleBox! Note: StyleShadowBox! is unsupported.
DisableNoScroll	Supported	lb_1.DisableNoScroll = FALSE lb_1.DisableNoScroll = lb_value
DragAuto	Supported	lb_1.DragAuto = TRUE
DragIcon	Supported	lb_1.DragIcon = 'c:\examples\arrow.ico' lb_1.DragIcon = 'Question!'
Enabled	Supported	lb_1.Enabled = lb_value
ExtendedSelect	Supported	lb_1.ExtendedSelect = lb_value
FaceName	Supported	lb_1.FaceName = ls_value
FontCharSet	Supported	lb_1.FontCharSet = ANSI!
FontFamily	Supported	lb_1.FontFamily = Roman!
FontPitch	Supported	lb_1.FontPitch = Fixed!
Height	Supported	lb_1.Height = li_value
HScrollBar	Supported	lb_1.HScrollBar = TRUE lb_1.HscrollBar = lb_value
Italic	Supported	lb_1.Italic = lb_value
Item[ ]	Supported	ls_value = lb_1.Item[1]

Property	Support Level	Example Code
MultiSelect	Supported	lb_1.MultiSelect = lb_value
Pointer	Supported	lb_1.Pointer = 'SizeNS!' lb_1.Pointer='d:\archive\IBEAM.BMP.cur'
RightToLeft	Supported	lb_1.RightToLeft = TRUE
Sorted	Supported	lb_1.Sorted = TRUE
TabOrder	Supported	lb_1.TabOrder = 60
Tag	Supported	lb_1.Tag = ls_value
TextColor	Supported	lb_1.TextColor = ll_value lb_1.TextColor = RGB(192,192,192)
TextSize	Supported	lb_1.TextSize = li_value
Underline	Supported	lb_1.Underline = lb_value
Visible	Supported	lb_1.Visible = lb_value
VScrollBar	Supported	lb_1.VscrollBar = lb_value
Weight	Supported	lb_1.Weight = li_value
Width	Supported	lb_1.Width = li_value
X	Supported	lb_1.X = li_value
Y	Supported	lb_1.Y = li_value
ClassDefinition	Unsupported	
TabStop[ ]	Unsupported	

6.1.1.12.2 Events for ListBox control

Table 6.36:

Event	Support Level	Example Code
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DoubleClicked	Supported	DoubleClicked()
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LoseFocus	Supported	LoseFocus()
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
SelectionChanged	Supported	SelectionChanged(index)
Help	Unsupported	
Other	Unsupported	

6.1.1.12.3 Functions for ListBox control

Table 6.37:

Function	Support Level	Example Code
AddItem	Supported	li_returnvalue = lb_1.AddItem(item)
ClassName	Supported	ls_ returnvalue = lb_1.ClassName()
DeleteItem	Supported	li_returnvalue = lb_1.DeleteItem(index)
DirList	Supported	lb_1.DirList("C:\EMPLOYEE*.TXT", 0)
DirSelect	Supported	lb_1.DirSelect(ls_filename)
Drag	Supported	lb_1.Drag(End!)
Find Item	Supported	li_returnvalue = lb_1.FindItem(text, index)
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = lb_1.GetParent()
Hide	Supported	li_returnvalue = lb_1.Hide()
InsertItem	Supported	li_returnvalue = lb_1.InsertItem(ls_x, li_y)
Move	Supported	li_returnvalue = string(lb_1.Move(li_x,li_y))
PointerX	Supported	li_returnvalue = lb_1.PointerX()
PointerY	Supported	li_returnvalue = lb_1.PointerY()
PostEvent	Supported	lb_returnvalue = string(lb_1.PostEvent(Clicked!))
Reset	Supported	li_returnvalue = lb_1.Reset()
Resize	Supported	li_returnvalue = lb_1.Resize(li_x,li_y)
SelectedIndex	Supported	li_returnvalue = lb_1.SelectedIndex()
SelectedItem	Supported	ls_returnvalue = lb_1.SelectedItem()
SelectItem	Supported	li_returnvalue = lb_1.SelectItem (item, index)
SetFocus	Supported	li_returnvalue = lb_1.SetFocus() Unsupported: SetFocus(lb_test)
SetPosition	Supported	li_returnvalue = lb_1.SetPosition(position)
SetRedraw	Supported	li_value = lb_1.SetRedraw(TRUE)
SetState	Supported	li_returnvalue = lb_1.SetState(index, lb_y)
SetTop	Supported	li_returnvalue = lb_1.SetTop(index)
Show	Supported	li_returnvalue = lb_1.Show()
State	Supported	li_returnvalue = lb_1.State(index)
Text	Supported	ls_returnvalue = lb_1.Text (index)
Top	Supported	li_returnvalue = lb_1.Top()
TotalItems	Supported	li_returnvalue = lb_1.TotalItems()
TotalSelected	Supported	li_returnvalue = lb_1.TotalSelected()
TriggerEvent	Supported	li_returnvalue = lb_1.TriggerEvent(event)
TypeOf	Supported	if lb_1.typeof() = listbox! Then

Function	Support Level	Example Code
		ls_returnvalue ='listbox!' else ls_returnvalue ='isvalid' end if
GetContextService	Unsupported	
Print	Unsupported	

6.1.1.13 ListView control

Important Requirements

In the ListView control, selecting multiple items at one time is unsupported.

6.1.1.13.1 Properties for ListView control

Table 6.38:

Property	Support Level	Example Code
Accelerator	Supported	lv_1.Accelerator=67
AutoArrange	Supported	lv_1.AutoArrange = TRUE
BackColor	Supported	lv_1.BackColor = ll_value lv_1.BackColor = RGB(192,192,192)
Border	Supported	lv_1.Border = lb_value
BorderStyle	Supported	lv_1.BorderStyle = StyleBox! lv_1.BorderStyle = StyleLowered! lv_1.BorderStyle = StyleRaised! Note: StyleShadowBox! is not supported.
BringToTop	Supported	lv_1.BringToTop = TRUE
ButtonHeader	Supported	lv_1.ButtonHeader = TRUE
CheckBoxes	Supported	lv_1.CheckBoxes = TRUE
DeleteItems	Supported	lv_1.DeleteItems = lb_value
DragAuto	Supported	lv_1.DragAuto = TRUE
DragIcon	Supported	lv_1.DragIcon = 'c:\archive\arrow.ico"
EditLabels	Supported	lv_1.EditLabels = lb_value
Enabled	Supported	lv_1.Enabled = lb_value
ExtendedSelect	Supported	lv_1.ExtendedSelect = TRUE
FaceName	Supported	lv_1.FaceName =ls_value
FixedLocations	Supported	lv_1.FixedLocations = FALSE
FontCharSet	Supported	lv_1.FontCharSet = ANSI!
FontFamily	Supported	lv_1.FontFamily = Roman!
FontPitch	Supported	lv_1.FontPitch = Fixed!

Property	Support Level	Example Code
FullRowSelect	Supported	lv_1.FullRowSelect = TRUE
GridLines	Supported	lv_1.GridLines = TRUE
HeaderDragDrop	Supported	lv_1.HeaderDragDrop = TRUE
Height	Supported	lv_1.Height = li_value
HideSelection	Supported	lv_1.HideSelection = FALSE
Italic	Supported	lv_1.Italic = lb_value
Item[ ]	Supported	lv_1.item[intx] = ls_value
ItemPicture Index[ ]	Supported	lv_1.ItemPictureIndex [li_x] = ls_value
LabelWrap	Supported	lv_1.LabelWrap = TRUE
LargePicture Height	Supported	lv_1.LargePictureHeight = li_value Note: This value cannot be set to 0 or negative.
LargePicture MaskColor	Supported	lv_1.LargePictureMaskColor = RGB(255, 255, 0)
LargePicture Name[ ]	Supported	lv_1.LargePictureName [li_x] = ls_value
LargePicture Width	Supported	lv_1.LargePictureWidth = li_value Note: This value cannot be set to 0 or negative.
OneClickActivate	Supported	lv_1.OneClickActivate = TRUE
Pointer	Supported	lv_1.Pointer = 'SizeNS!' lv_1.Pointer = 'd:\archive\IBEAM.BMP.cur'
Scrolling	Supported	lv_1.Scrolling = TRUE
ShowHeader	Supported	lv_1.ShowHeader = TRUE
SmallPicture Height	Supported	lv_1.SmallPictureHeight = li_value Note: This value cannot be set to 0 or negative.
SmallPicture MaskColor	Supported	lv_1.SmallPictureMaskColor = RGB(255, 255, 0)
SmallPictureName[ ]	Supported	lv_1.SmallPictureName [li_x] = ls_value
SmallPicture Width	Supported	lv_1.SmallPictureWidth = li_value Note: This value cannot be set to 0 or negative.
SortType	Supported	lv_1.SortType = Unsorted!
StatePicture MaskColor	Supported	lv_1.StatePictureMaskColor = RGB(255, 255, 0)
StatePicture Name[ ]	Supported	lv_1.StatePictureName [li_x] = ls_value
TabOrder	Supported	lv_1.TabOrder = 70

Property	Support Level	Example Code
Tag	Supported	lv_1.Tag = ls_value
TextColor	Supported	lv_1.TextColor = ll_value lv_1.TextColor = RGB(192,192,192)
TextSize	Supported	lv_1.TextSize = li_value
TrackSelect	Supported	lv_1.TrackSelect = TRUE
TwoClickActivate	Supported	lv_1.TwoClickActivate = TRUE
Underline	Supported	lv_1.Underline = lb_value
UnderlineCold	Supported	lv_1.UnderlineCold = TRUE
UnderlineHot	Supported	lv_1.UnderlineHot = TRUE
View	Supported	lv_1.View = ListViewLargeIcon!
Visible	Supported	lv_1.Visible = lb_value
Weight	Supported	lv_1.Weight = li_value
Width	Supported	lv_1.Width = li_value
X	Supported	lv_1.X = li_value
Y	Supported	lv_1.Y =li_value
ClassDefinition	Unsupported	
LayoutRTL	Unsupported	
StatePictureHeight	Unsupported	
StatePictureWidth	Unsupported	

6.1.1.13.2 Events for ListView control

Table 6.39:

Event	Support Level	Example Code
BeginDrag	Supported	BeginDrag(index)
BeginLabelEdit	Supported	BeginLabelEdit(index)
BeginRightDrag	Supported	BeginRightDrag(index)
Clicked	Supported	Clicked(index)
ColumnClick	Supported	ColumnClick(column)
Constructor	Supported	Constructor()
DeleteAllItems	Supported	DeleteAllItems()
DeleteItem	Supported	DeleteItem(index)
Destructor	Supported	Destructor()
DoubleClicked	Supported	DoubleClicked(index)
DragDrop	Supported	DragDrop(source,index)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)

Event	Support Level	Example Code
DragWithin	Supported	DragWithin(source,index)
EndLabelEdit	Supported	EndLabelEdit(index,newlabel)
GetFocus	Supported	GetFocus()
InsertItem	Supported	InsertIem(index)
ItemActivate	Supported	ItemActivate(index, focuschanged, hasfocus, selectionchange, selected, otherchange)
ItemChanged	Supported	ItemChanged(index, focuschanged, hasfocus, selectionchange, selected, otherchange)
ItemChanging	Supported	ItemChanging(index, focuschange, hasfocus, selectionchange, selected, otherchange)
Key	Supported	Key(key,keyflags)
LoseFocus	Supported	LoseFocus()
RightClicked	Supported	RightClicked(index)
RightDouble Clicked	Supported	RightDoubleClicked(index)
Help	Unsupported	
Other	Unsupported	
Sort	Unsupported	

6.1.1.13.3 Functions for ListView control

Table 6.40:

Function	Support Level	Example Code
AddColumn	Supported	li_value = lv_1.AddColumn(ls_x,Center!,li_y)
AddItem	Supported	li_value = lv_1.AddItem(ls_x,li_y)
AddLargePicture	Supported	li_value = lv_1.AddLargePicture(ls_prop)
AddSmallPicture	Supported	li_value = lv_1.AddSmallPicture(ls_prop)
AddStatePicture	Supported	li_value = lv_1.AddstatePicture(ls_prop)
Arrange	Supported	li_value = lv_1.Arrange()
ClassName	Supported	ls_value = lv_1.ClassName( )
DeleteColumn	Supported	li_value = lv_1.DeleteColumn(1)
DeleteColumns	Supported	li_value = lv_1.DeleteColumns( )
DeleteItem	Supported	li_value = lv_1.DeleteItem(li_x)
DeleteItems	Supported	li_value = lv_1.DeleteItems( )
DeleteLarge Picture	Supported	li_value = lv_1.DeleteLargePicture(li_x)
DeleteLarge Pictures	Supported	li_value = lv_1.DeleteLargePictures( )

Function	Support Level	Example Code
DeleteSmallPicture	Supported	li_value = lv_1.DeleteSmallPicture(li_x)
DeleteSmall Pictures	Supported	li_value = lv_1.DeleteSmallPictures( )
DeleteState Picture	Supported	li_value = lv_1.DeleteStatePicture(li_x)
DeleteState Pictures	Supported	li_value = lv_1.DeleteStatePictures( )
Drag	Supported	lv_1.Drag(Begin!)
EditLabel	Supported	li_value= lv_1.EditLabel()
FindItem	Supported	li_Value = lv_1.FindItem (startindex, label, partial, wrap) li_value = lv_1.FindItem (startindex, direction, focused, selected, cuthighlighted, drophighlighted)
GetColumn	Supported	li_strValue = lv_1.GetColumn(index, label, alignment, width)
GetItem	Supported	li_value = lv_1.GetItem(li_x,lv1)
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = lv_1.GetParent()
Hide	Supported	li_value = lv_1.Hide( )
InsertColumn	Supported	li_value = lv_1.InsertColumn(li_x,ls_x,Center!,li_y)
InsertItem	Supported	li_value = lv_1.InsertItem(2, "Insert String")
Move	Supported	li_value = lv_1.Move(li_x,li_y)
PointerX	Supported	li_value = lv_1.PointerX()
PointerY	Supported	li_value = lv_1.PointerY()
PostEvent	Supported	lv_1.PostEvent(Clicked!)
Resize	Supported	li_value = lv_1.Resize(li_x,li_y)
SelectedIndex	Supported	li_value = lv_1.SelectedIndex( )
SetColumn	Supported	li_value = lv_1.SetColumn(li_x,ls_x,Center!,li_y)
SetFocus	Supported	li_value = lv_1.SetFocus( ) Unsupported: SetFocus(lv_test)
SetItem	Supported	li_value = lv_1.SetItem(li_x[],li_y[],lv1)
SetPosition	Supported	li_value = lv_1.SetPosition(Behind!, lv_three)
SetRedraw	Supported	li_value = lv_1.SetRedraw(TRUE)
Show	Supported	li_value = lv_1.Show( )
Sort	Supported	li_value = lv_1.Sort(Ascending!) li_value = lv_1.Sort(Descending!) li_value = lv_1.Sort(Unsorted!)

Function	Support Level	Example Code
		Note: userdefinesort! is not supported.
TotalColumns	Supported	li_cols = lv_1.TotalColumns( )
TotalItems	Supported	li_value = lv_1.TotalItems( )
TriggerEvent	Supported	li_value = lv_1.TriggerEvent("Clicked")

TypeOf	Supported	if lv_1.TypeOf( ) = ListView! Then
		ls_value = "ListView!"
		else
		ls_value = "invalid"
		end if

SetOverlayPicture	Supported	lv_1.SetOverlayPicture(1, index)
TotalSelected	Supported	
GetContextService	Unsupported	
GetOrigin	Unsupported	
Print	Unsupported	

6.1.1.14 MonthCalendar control

6.1.1.14.1 Properties for MonthCalendar control

Table 6.41:

Property	Support Level	Example Code
Accelerator	Supported	mc_1.Accelerator=67
AccessibleDescription	Supported	
AccessibleName	Supported	
AccessibleRole	Supported	
AutoSize	Supported	mc_1.AutoSize = false
BackColor	Supported	mc_1.BackColor = RGB(0, 0, 255) mc_1.BackColor = ll_value
Border	Supported	mc_1.Border = TRUE mc_1.Border = lb_value
BorderStyle	Supported	mc_1.BorderStyle = StyleBox! mc_1.BorderStyle = StyleLowered! mc_1.BorderStyle = StyleRaised! Note: StyleShadowBox! is unsupported.
BringToTop	Supported	mc_1.BringToTop = TRUE
DragAuto	Supported	mc_1.DragAuto = TRUE

Property	Support Level	Example Code
		Note: On the Web, dragging a DatePicker control will not trigger the DropDown event if the DragAuto property is set to True, which means there is no Month Calender displayed on the Web when you Drag the DatePicker with setting the DragAuto property to True.
DragIcon	Supported	mc_1.DragIcon = 'arrow.ico'
Enabled	Supported	mc_1.Enabled=TRUE mc_1.Enabled = lb_value
FaceName	Supported	mc_1.FaceName = "" mc_1.FaceName = ls_value
FirstDayOfWeek	Supported	mc_1.FirstDayOfWeek = Monday!
FontCharSet	Supported	mc_1.FontCharSet = ANSI!
FontFamily	Supported	mc_1.FontFamily = Roman!
FontPitch	Supported	mc_1.FontPitch = Fixed!
Height	Supported	mc_1.Height = 750 mc_1.Height = li_value
Italic	Supported	mc_1.Italic=TRUE mc_1.Italic = lb_value
MaxSelectCount	Supported	mc_1.MaxSelectCount = 7
MonthBackColor	Supported	mc_1.MonthBackColor = RGB(255, 255, 0
Pointer	Supported	mc_1.Pointer = 'Size!'
ScrollRate	Supported	mc_1.ScrollRate = 3
TabOrder	Supported	mc_1.TabOrder = 30
Tag	Supported	mc_1.Tag = ls_value
TextColor	Supported	mc_1.TextColor = ll_value
TextSize	Supported	mc_1.TextSize = li_value
TitleBackColor	Supported	mc_1.TitleBackColor = RGB(128, 255, 128)
TitleTextColor	Supported	mc_1.TitleTextColor = RGB(0,0,255)
TodayCircle	Supported	mc_1.TodaySection = false
TodaySection	Supported	mc_1.TodaySection = false
TrailingTextColor	Supported	mc_1.TrailingTextColor = 134217731
Underline	Supported	mc_1.Underline = lb_value
Visible	Supported	mc_1.Visible = True
WeekNumbers	Supported	mc_1.WeekNumbers = true
Weight	Supported	mc_1.Weight = 700 mc_1.Weight = li_value
Width	Supported	mc_1.Width = li_value

Property	Support Level	Example Code
		Note: 1. If the Width value is set to -32768, obtaining the width property returns -32768 on the Web, however it returns 0 in PowerBuilder. 2. If the Width value is set to 32768, obtaining the width property returns 0 on the Web, however it returns -32768 in PowerBuilder.
X	Supported	mc_1.X = li_value Note: 1. If the X value is set to -32768, obtaining the X property returns -32768 on the Web, however it returns 0 in PowerBuilder. 2. If the Width value is set to 32768, obtaining the width property returns 0 on the Web, however it returns -32768 in PowerBuilder.
Y	Supported	mc_1.Y = li_value Note: 1. If the Y value is set to -32768, obtaining the Y property returns -32768 on the Web, however it returns 0 in PowerBuilder. 2. If the Width value is set to 32768, obtaining the width property returns 0 on the Web, however it returns -32768 in PowerBuilder.
ClassDefinition	Unsupported	
RightToLeft	Unsupported	

6.1.1.14.2 Events for MonthCalendar control

Table 6.42:

Event	Support Level	Example Code
Clicked	Supported	Clicked()
Constructor	Supported	Constructor()
DateChanged	Supported	DateChanged() Note: SetSelectedDate and SetSelectedRange trigger a DateChanged event twice on the Web.
Destructor	Supported	Destructor()
DoubleClicked	Supported	Doubleclicked(flags,xpos,ypos)
DragDrop	Supported	DragDrop(source)

Event	Support Level	Example Code
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
GetFocus	Unsupported	
LoseFocus	Unsupported	
Help	Unsupported	
UserString	Unsupported	
Other	Unsupported	

6.1.1.14.3 Functions for MonthCalendar Control

Table 6.43:

Function	Support Level	Example Code
ClassName	Supported	ls_value = mc_1.ClassName()
ClearBoldDates	Supported	li_return = mc_1.ClearBoldDates()
Drag	Supported	mc_1.Drag(Begin!) Note: If you use Drag(Begin!), single mouse clicking will trigger DragEnter and DragDrop event in sequence on the Web. This is different from the event sequences in PowerBuilder.
GetDateLimits	Supported	li_return = mc_1.GetDateLimits (mindate, maxdate)
GetDisplayRange	Supported	li_return = mc_1.GetDisplayRange (startdate, enddate,PartlyDisplayed!)
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = mc_1.GetParent()
GetSelectedDate	Supported	li_return = mc_1.GetSelectedDate(seldate) Notes: 1. The function returns 0 on the Web if there are several dates selected or the specified date is an invalid value. 2. The function will trigger a DateChanged event twice on the Web.
GetSelectedRange	Supported	li_return = mc_1.GetSelectedRange(startdate, enddate) Note: The function will trigger a DateChanged event twice on the Web.
GetToday	Supported	currentdate = mc_1.GetToday()
Hide	Supported	li_return = mc_1.Hide()

Function	Support Level	Example Code
Move	Supported	li_return = mc_1.Move(li_x,li_y)
PointerX	Supported	li_value = mc_1.PointerX()
PointerY	Supported	li_value = mc_1.PointerY()
PostEvent	Supported	mc_1.PostEvent(Constructor!)
Resize	Supported	li_return = mc_1.Resize(li_width,li_height)
SetBoldDate	Supported	li_return = mc_1.SetBoldDate(d, true)
SetDateLimits	Supported	li_return = mc_1.SetDateLimits(mindate, maxdate)
SetFocus	Supported	li_return = mc_1.SetFocus()
SetPosition	Supported	li_return = mc_1.SetPosition(position)
SetRedraw	Supported	li_return= mc_1.SetRedraw(FALSE)
SetSelectedDate	Supported	li_return = mc_1.SetSelectedDate(seldate) Notes: If input an invalid date, the function returns 0 on the Web but -1 in PowerBuilder.
SetSelectedRange	Supported	li_return = mc_1.SetSelectedRange(startdate, enddate)
SetToday	Supported	li_return = mc_1.SetToday(currentdate)
Show	Supported	li_return = mc_1.Show()
TriggerEvent	Supported	mc_1.TriggerEvent(Constructor!)

TypeOf	Supported	if mc_1.TypeOf() = MonthCalendar! Then
		ls_return = 'MonthCalendar!'
		else
		ls_return = 'isvalid'
		end if

GetContextService	Unsupported	
Print	Unsupported	

6.1.1.15 MultiLineEdit control

6.1.1.15.1 Properties for MultiLineEdit control

Table 6.44:

Property	Support Level	Example Code
Accelerator	Supported	mle_1.Accelerator = 67
Alignment	Supported	mle_1.Alignment = Center!
AutoHScroll	Supported	mle_1.AutoHScroll = TRUE
AutoVScroll	Supported	mle_1.AutoVscroll = TRUE
BackColor	Supported	mle_1.BackColor = ll_value mle_1.BackColor = RGB(192,192,192)

Property	Support Level	Example Code
Border	Supported	mle_1.Border = lb_value
BorderStyle	Supported	mle_1.BorderStyle = StyleBox! mle_1.BorderStyle = StyleLowered! mle_1.BorderStyle = StyleRaised! Note: StyleShadowBox! is unsupported.
BringToTop	Supported	mle_1.BringToTop = TRUE
DisplayOnly	Supported	mle_1.DisplayOnly =lb_value
DragAuto	Supported	mle_1.DragAuto = TRUE
DragIcon	Supported	mle_1.DragIcon = 'c:\archive\arrow.ico''
Enabled	Supported	mle_1.Enabled = lb_valueM Note: when the user moves the pointer over a MultiLineEdit control that has set the Enabled property to false, the pointer on the Web will be different from that in the PowerBuilder application.
FaceName	Supported	mle_1.FaceName = ls_value
FontCharSet	Supported	mle_1.FontCharSet = ANSI!
FontFamily	Supported	mle_1.FontFamily = Roman!
FontPitch	Supported	mle_1.FontPitch = Fixed!
Height	Supported	mle_1.Height = li_value
HideSelection	Supported	mle_1.HideSelection = FALSE
HScrollBar	Supported	mle_1.HScrollBar = TRUE
IgnoreDefault Button	Supported	mle_1.IgnoreDefaultButton = TRUE
Italic	Supported	mle_1.Italic = lb_value
Limit	Supported	mle_1.Limit = li_value
Pointer	Supported	mle_1.pointer= 'beam!' mle_1.pointer ='d:\archive\IBEAM.BMP.cur'
RightToLeft	Supported	mle_1.RightToLeft = TRUE
TabOrder	Supported	mle_1.TabOrder = 80
Tag	Supported	mle_1.Tag = ls_value
Text	Supported	mle_1.Text = ls_value It is unsupported to set the value of this property to null.
TextCase	Supported	mle_1.TextCase = AnyCase! mle_1.TextCase = Lower! mle_1.TextCase = Upper!

Property	Support Level	Example Code
TextColor	Supported	mle_1.TextColor = ll_value
TextSize	Supported	mle_1.TextSize = li_value
Underline	Supported	mle_1.Underline = lb_value
Visible	Supported	mle_1.Visible = lb_value
VScrollBar	Supported	mle_1.VScrollBar = TRUE
Weight	Supported	mle_1.Weight = li_value
Width	Supported	mle_1.Width = li_value
X	Supported	mle_1.X = li_value
Y	Supported	mle_1.Y = li_value
ClassDefinition	Unsupported	
TabStop[ ]	Unsupported	

6.1.1.15.2 Events for MultiLineEdit control

Table 6.45:

Event	Support Level	Example Code
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LoseFocus	Supported	LoseFocus()
Modified	Supported	Modified() A Modified event is followed by a LoseFocus event.
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
Help	Unsupported	
Other	Unsupported	

6.1.1.15.3 Functions for MultiLineEdit control

Table 6.46:

Function	Support Level	Example Code
ClassName	Supported	ls_value = mle_1.ClassName()
Clear	Supported	ls_value = mle_1.Clear() Note: the function executes successfully only when the control has focus.
Copy	Supported	ls_value = mle_1.Copy()

Function	Support Level	Example Code
Cut	Supported	ls_value = mle_1.Cut()
Drag	Supported	mle_1.Drag(Begin!)
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = mle_1.GetParent()
Hide	Supported	ls_value = mle_1.Hide()
LineCount	Supported	li_value = mle_1.LineCount()
LineLength	Supported	li_value = mle_1.LineLength()
Move	Supported	ls_value = mle_1.Move(li_x,li_y)
Paste	Supported	ls_value = mle_1.Paste()
PointerX	Supported	ls_value = mle_1.PointerX()
PointerY	Supported	ls_value = mle_1.PointerY()
Position	Supported	mle_1.Position()
PostEvent	Supported	ls_value = mle_1.PostEvent(Constructor!)
ReplaceText	Supported	ls_strexevalue = mle_1.ReplaceText(ls_x)
Resize	Supported	ls_value = mle_1.Resize(li_x,li_y)
Scroll	Supported	mle_1.Scroll(4)
SelectedLength	Supported	ls_value = mle_1.SelectedLength()
SelectedLine	Supported	integer li_SL li_SL = mle_1.SelectedLine()
SelectedStart	Supported	ls_value = mle_1.SelectedStart()
SelectedText	Supported	ls_value = mle_1.SelectedText()
SelectText	Supported	li_value = mle_1.SelectText(li_x,li_y)
SetFocus	Supported	ls_value = mle_1.SetFocus() Unsupported: SetFocus(mle_1)
SetPosition	Supported	mle_1.SetPosition(Behind!, mle_2) ls_value = mle_1.SetPosition(ToTop!) ls_value = mle_1.SetPosition(ToBottom!)
SetRedraw	Supported	li_return= mle_1.SetRedraw(FALSE)
Show	Supported	ls_value = mle_1.Show()
TextLine	Supported	ls_value = mle_1.TextLine()
TriggerEvent	Supported	ls_value = mle_1.TriggerEvent(Constructor!)

TypeOf	Supported	if mle_1.typeof() = MultiLineEdit! Then
		ls_value = 'MultiLineEdit!'
		else
		ls_value = 'Invalid' end if

Function	Support Level	Example Code
Undo	Supported	li_value = mle_1.Undo()
CanUndo	Unsupported	
GetContextService	Unsupported	
Print	Unsupported	

6.1.1.16 OLEControl control

Important Requirements

The shortcut key for Ocx is unsupported.

Using parentheses when calling an OLEObject method is required.

6.1.1.16.1 Properties for OLEControl control

Table 6.47:

Property	Support Level	Example Code
Activation	Supported	ole_1.Activation = ActivateOnDoubleClick! Note: activateOnGetfocus! is unsupported.
BackColor	Supported	ole_1.BackColor = ll_value ole_1.BackColor = RGB(192,192,192)
BinaryIndex	Supported	
BinaryKey	Supported	
Border	Supported	ole_1.Border = TRUE ole_1.Border = lb_value
BorderStyle	Supported	ole_1.BorderStyle = StyleBox! ole_1.BorderStyle = StyleLowered! Note: StyleShadowBox! is unsupported.
BringToTop	Supported	ole_1.BringToTop = TRUE
DisplayName	Supported	ole_1.DisplayName = "My project"
DisplayType	Supported	ole_1.DisplayType = DisplayAsIcon! Note: displayasactivexdocument! is unsupported
DocFileName	Supported	ls_ret = ole_1.docfilename
DragAuto	Supported	ole_1.DragAuto = TRUE
DragIcon	Supported	ole_1.DragIcon = 'Question!'
Enabled	Supported	ole_1.Enabled=TRUE ole_1.Enabled = lb_value
FocusRectangle	Supported	ole_1.FocusRectangle = TRUE ole_1.FocusRectangle = lb_value

Property	Support Level	Example Code
Height	Supported	ole_1.Height = 750 ole_1.Height = li_value
LinkItem	Supported	ls_ret = iole.LinkItem
Object	Supported	OLEObject obj = ole_1.object.activesheet.cells Int x = ole_1.object.month
ObjectData	Supported	blob lblob_data = ole_1.objectdata
Pointer	Supported	ole_1.Pointer = 'SizeWE!' ole_1.Pointer='d:\archive\IBEAM.BMP.cur'
SizeMode	Supported	ole_1.SizeMode = Clip!
TabOrder	Supported	ole_1.TabOrder = 40 ole_1.TabOrder = li_order
Tag	Supported	ole_1.Tag = ls_value
Visible	Supported	ole_1.Visible = lb_value
Width	Supported	ole_1.Width = li_value
X	Supported	ole_1.X = li_value
Y	Supported	ole_1.Y = li_value
ClassDefinition	Unsupported	
ClassLongName	Unsupported	
ClassShortName	Unsupported	
ContentsAllowed	Unsupported	
IsDragTarget	Unsupported	
LinkUpdate Options	Unsupported	
ParentStorage	Unsupported	

6.1.1.16.2 Events for OLEControl control

Table 6.48:

Event	Support Level	Example Code
Clicked	Supported	Clicked()
Constructor	Supported	Constructor()
DataChange	Supported	DataChange()
Destructor	Supported	Destructor()
DoubleClicked	Supported	DoubleClicked()
GetFocus	Supported	GetFocus()
LoseFocus	Supported	LoseFocus()
PropertyChanged	Supported	PropertyChanged(propertyname)

Event	Support Level	Example Code
PropertyRequestEdit	Supported	PropertyRequestEdit(propertyname,cancelchange)
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
SaveObject	Supported	
Close	Unsupported	
DragDrop	Unsupported	
DragEnter	Unsupported	
DragLeave	Unsupported	
DragWithin	Unsupported	
Error	Unsupported	
ExternalException	Unsupported	
Help	Unsupported	
Other	Unsupported	
Rename	Unsupported	
Save	Unsupported	
ViewChange	Unsupported	

6.1.1.16.3 Functions for OLEControl control

Table 6.49:

Function	Support Level	Example Code
Activate	Supported	integer li_return li_return = ole_1.Activate(OffSite!) Note: the argument InPlace! is unsupported
ClassName	Supported	ls_return = ole_1.ClassName()
Clear	Supported	li_return = ole_1.Clear()
DoVerb	Supported	li_return = ole_1.DoVerb(7)
GetData	Supported	li_return = ole_1.GetData(ClipFormatText!, ls_oledata)
GetNativePointer	Supported	UnsignedLong lul_oleptr li_return = ole_1.GetNativePointer(lul_oleptr)
GetParent	Supported	PowerObject lpo_parent lpo_parent = ole_1.GetParent()
Hide	Supported	ole_1.Hide()
InsertClass	Supported	li_return = ole_1.InsertClass("excel.sheet")
InsertFile	Supported	li_return = ole_1.InsertFile("c:\xls\expense.xls")
InsertObject	Supported	li_return = ole_1.InsertObject()

Function	Support Level	Example Code
		Items listed in the Web Insert Object dialog box contain and only contain all the registered OLE applications in the client. Items listed in PowerBuilder Insert object dialog box are defined in PowerBuilder.
LinkTo	Supported	li_return = ole_1.LinkTo("c:\xls\expense.xls", "R1C1:R5C5")
Move	Supported	li_return = ole_1.Move(P_Train1.X + P_Train1.Width, P_Train1.Y)
Open	Supported	li_return = ole_1.Open("c:\ole2\mystuff.ole")
PointerX	Supported	li_return = ole_1.PointX()
PointerY	Supported	li_return = ole_1.PointY()
PostEvent	Supported	ole_1.PostEvent("DataChange")
ReleaseNative Pointer	Supported	li_return = ole_1.ReleaseNativePointer(lul_oleptr)
Resize	Supported	li_return = ole_1.Resize(500, 1000)
Save	Supported	li_return = ole_1.Save()
SaveAs	Supported	li_return = ole_1.SaveAs("c:\ole\expense.ole")
SetData	Supported	li_return = ole_1.SetData(ClipFormatText!, ls_oledata)
SetFocus	Supported	li_return = ole_1.SetFocus()
SetPosition	Supported	ole_1.SetPosition(ToTop!) ole_1.SetPosition(Behind!, cb_exec)
SetRedraw	Supported	ole_1.SetRedraw(true)
Show	Supported	ole_1.Show()
TriggerEvent	Supported	li_return = ole_1.TriggerEvent("DataChange")
TypeOf	Supported	if ole_1.typeof()=olecontrol! then ls_return="olecontrol!"
Copy	Unsupported	
Cut	Unsupported	
Drag	Unsupported	
GetContextService	Unsupported	
Paste	Unsupported	
PasteLink	Unsupported	
PasteSpecial	Unsupported	
Print	Unsupported	
SelectObject	Unsupported	

Function	Support Level	Example Code
UpdateLinks Dialog	Unsupported	

6.1.1.17 OLECustomControl control

Important Requirements

The shortcut key for Ocx is unsupported.

Using parentheses when calling an OLEObject method is required.

6.1.1.17.1 Properties for OLECustomControl control

Table 6.50:

Property	Support Level	Example Code
BackColor	Supported	ole_1.BackColor = ll_value ole_1.BackColor = RGB(192,192,192)
BinaryIndex	Supported	
BinaryKey	Supported	
Border	Supported	ole_1.Border = TRUE ole_1.Border = lb_value
BorderStyle	Supported	ole_1.BorderStyle = StyleBox! ole_1.BorderStyle = StyleLowered! Note: StyleShadowBox! is unsupported.
BringToTop	Supported	ole_1.BringToTop = TRUE
DisplayName	Supported	ole_1.DisplayName = "My project"
DragAuto	Supported	ole_1.DragAuto = TRUE
DragIcon	Supported	ole_1.DragIcon = 'Question!'
Enabled	Supported	ole_1.Enabled =TRUE ole_1.Enabled = lb_value
FocusRectangle	Supported	ole_1.FocusRectangle = TRUE ole_1.FocusRectangle = lb_value
Object	Supported	OLEObject obj = ole_1.object.activesheet.cells Int x = ole_1.object.month
Pointer	Supported	ole_1.Pointer = 'SizeNWSE!' ole_1.Pointer='d:\archive\IBEAM.BMP.cur'
TabOrder	Supported	ole_1.TabOrder = 40 ole_1.TabOrder = li_order
Tag	Supported	ole_1.Tag = ls_value
Visible	Supported	ole_1.Visible = lb_value

Property	Support Level	Example Code
Width	Supported	ole_1.Width = li_value
X	Supported	ole_1.X = li_value
Y	Supported	ole_1.Y = li_value
Alignment	Unsupported	
Cancel	Unsupported	
ClassDefinition	Unsupported	
ClassLongName	Unsupported	
ClassShortName	Unsupported	
Default	Unsupported	
FaceName	Unsupported	
FontCharSet	Unsupported	
FontFamily	Unsupported	
FontPitch	Unsupported	
IsDragTarget	Unsupported	
Italic	Unsupported	
TextColor	Unsupported	
TextSize	Unsupported	
Underline	Unsupported	
Weight	Unsupported	

6.1.1.17.2 Events for OLECustomControl control

Table 6.51:

Event	Support Level	Example Code
Clicked	Supported	Clicked()
Constructor	Supported	Constructor()
DataChange	Supported	DataChange()
Destructor	Supported	Destructor()
DoubleClicked	Supported	DoubleClicked()
GetFocus	Supported	GetFocus()
LoseFocus	Supported	LoseFocus()
PropertyChanged	Supported	PropertyChanged(propertyname)
PropertyRequestEdit	Supported	PropertyRequestEdit(propertyname,cancelchange)
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
DragDrop	Unsupported	
DragEnter	Unsupported	

Event	Support Level	Example Code
DragLeave	Unsupported	
DragWithin	Unsupported	
Error	Unsupported	
ExternalException	Unsupported	
Help	Unsupported	
Other	Unsupported	

6.1.1.17.3 Functions for OLECustomControl control

Table 6.52:

Function	Support Level	Example Code
ClassName	Supported	ls_return = ole_1.ClassName( )
GetData	Supported	li_return = ole_1.GetData(ClipFormatText!, ls_oledata)
GetNativePointer	Supported	UnsignedLong lul_oleptr li_return = ole_1.GetNativePointer(lul_oleptr)
GetParent	Supported	PowerObject lpo_parent lpo_parent = ole_1.GetParent()
Hide	Supported	li_return = ole_1.Hide()
Move	Supported	li_return = ole_1.Move(500 , 1000)
PointerX	Supported	li_return = ole_1.PointX()
PointerY	Supported	li_return = ole_1.PointY( )
PostEvent	Supported	ole_1.PostEvent("DoubleClicked")
ReleaseNative Pointer	Supported	li_return = ole_1.ReleaseNativePointer(lul_oleptr)
Resize	Supported	li_return = ole_1.Resize(500, 1000)
SetData	Supported	li_return = ole_1.SetData(ClipFormatText!, ls_oledata)
SetFocus	Supported	li_return = ole_1.SetFocus( )
SetPosition	Supported	ole_1.SetPosition(ToTop!) ole_1.SetPosition(Behind!, cb_exec)
SetRedraw	Supported	ole_1.SetRedraw(true)
Show	Supported	li_return = ole_1.Show( )
TriggerEvent	Supported	li_return = ole_1.TriggerEvent("DoubleClicked")

TypeOf	Supported	if ole_1.typeof( )=olecustomcontrol! then
		ls_return="olecustomcontrol!"
		end if

Function	Support Level	Example Code
Drag	Unsupported	
GetContextService	Unsupported	
Print	Unsupported	

6.1.1.18 Oval control

6.1.1.18.1 Properties for Oval control

Table 6.53:

Property	Support Level	Example Code
FillColor	Supported	o_1.FillColor=RGB(255,255,0)
FillPattern	Supported	o_1.FillPattern = Diamond!
Height	Supported	o_1.Height = 750
LineColor	Supported	o_1.LineColor = RGB(255,0,0)
LineStyle	Supported	o_1.LineStyle = Dash!
LineThickness	Supported	
Tag	Supported	o_1.SetMicroHelp(This.Tag)
Visible	Supported	o_1.Visible = TRUE
Width	Supported	o_1.Width = 750
X	Supported	o_1.X = 215
Y	Supported	o_1.Y = 215
ClassDefinition	Unsupported	

6.1.1.18.2 Events for Oval control

Table 6.54:

Event	Support Level	Example Code
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()

6.1.1.18.3 Functions for Oval control

Table 6.55:

Function	Support Level	Example Code
ClassName	Supported	ls_value = o_1.ClassName()
GetParent	Supported	lo_value = o_1.GetParent()
Hide	Supported	o_1.Hide()
Move	Supported	o_1.Move(150, 200)
PostEvent	Supported	o_1.PostEvent(Clicked!) Note: The PostEvent function returns 1 if it is successful.

Function	Support Level	Example Code
Resize	Supported	o_1.Resize(100, 150)
Show	Supported	o_1.Show()
TriggerEvent	Supported	o_1.TriggerEvent(Clicked!) Note: in an Web application the TriggerEvent function returns 1 if it is successful.

TypeOf	Supported	if o_1.typeof()=Oval! then
		ls_return="Oval!"
		end if

GetContextService	Unsupported	

6.1.1.19 Picture control

Important Requirements

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
- Ico files with .ICO extensions
- Cursor files with .CUR extensions
- PNG files with .PNG extensions

Pictures in an application

- The name of a picture file cannot be a string that ends with '!'.
- The picture directory can be an absolute path or relative path.

  For example:

  this.picturename = "c:\app\about.bmp"

  or this.picturename = " image\about.bmp"
- If a picture fails to display in the Web application, copy the picture directly to the *Web Root Path /images* directory. *Web Root Path* stands for the folder in the Web server that is configured in Appeon Developer Config for storing the Web application files.

6.1.1.19.1 Properties for Picture control

Table 6.56:

Property	Support Level	Example Code
Border	Supported	p_1.Border = lb_value
BringToTop	Supported	p_1.BringToTop = TRUE

Property	Support Level	Example Code
BorderStyle	Supported	p_1.BorderStyle = StyleBox! p_1.BorderStyle = StyleLowered! p_1.BorderStyle = StyleRaised! Note: StyleShadowBox! is unsupported.
DragAuto	Supported	p_1.DragAuto = TRUE
DragIcon	Supported	p_1.DragIcon = 'c:\examples\arrow.ico'
Enabled	Supported	p_1.Enabled = lb_value
FocusRectangle	Supported	p_1.FocusRectangle = TRUE
Height	Supported	p_1.Height = li_value
Invert	Supported	p_1.Invert = FALSE
OriginalSize	Supported	p_1.OriginalSize = lb_value
PictureName	Supported	p_1.PictureName = ls_value Note: It is unsupported to dynamically change this property if the initial value is a GIF file.
Pointer	Supported	p_1.pointer= ‘size!’ p_1.pointer= "'d:\archive\IBEAM.BMP.cur'
PowerTipText	Supported	p_1.PowerTipText = "Cancel the operation"
TabOrder	Supported	p_1.TabOrder = 100
Tag	Supported	p_1.Tag = ls_value
Visible	Supported	p_1.Visible = lb_value
Width	Supported	p_1.Width = li_value
X	Supported	p_1.X = li_value
Y	Supported	p_1.Y = li_value
ClassDefinition	Unsupported	
Map3DColors	Unsupported	

6.1.1.19.2 Events for Picture control

Table 6.57:

Event	Support Level	Example Code
Clicked	Supported	Clicked()
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DoubleClicked	Supported	DoubleClicked()
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)

Event	Support Level	Example Code
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LoseFocus	Supported	LoseFocus()
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
Help	Unsupported	
Other	Unsupported	

6.1.1.19.3 Functions for Picture control

Table 6.58:

Function	Support Level	Example Code
ClassName	Supported	ls_value = p_1.ClassName( )
Drag	Supported	p_1.Drag(Begin!)
Draw	Supported	li_value = p_1.Draw(li_x,li_y) Note: run the Draw function once the picture has been resized. After conversion, the size of the modified picture will be the same as the most recent state. In PowerBuilder, the size of the drawn picture will be the same as the original one.
GetParent	Supported	PowerObject lpo_value lpo_value = p_1.GetParent( )
Hide	Supported	li_value = p_1.Hide( )
Move	Supported	li_value = p_1.Move(li_x,li_y)
PointerX	Supported	li_value = p_1.PointerX()
PointerY	Supported	li_value = p_1.PointerY()
PostEvent	Supported	lb_value = p_1.PostEvent(Clicked!)
Resize	Supported	li_value = p_1.Resize(li_x,li_y)
SetFocus	Supported	li_value = p_1.SetFocus( ) Unsupported: SetFocus(p_1)
SetPicture	Supported	p_1.SetPicture(Emp_pic)
SetPosition	Supported	li_value = p_1.SetPosition( )
SetRedraw	Supported	li_value = p_1.SetRedraw(FALSE)
Show	Supported	li_value = p_1.Show( )
TriggerEvent	Supported	li_value = p_1.TriggerEvent(Clicked!)

TypeOf	Supported	if p_1.TypeOf( ) = Picture! Then
		ls_value = "Picture!"
		else
		ls_value = "invalid"

Function	Support Level	Example Code
		end if
GetContextService	Unsupported	
Print	Unsupported	

6.1.1.20 PictureButton control

Important Requirements

It is unsupported to use GIF files in PictureButton controls.

6.1.1.20.1 Properties for PictureButton control

Table 6.59:

Property	Support Level	Example Code
BackColor	Supported	pb_1.BackColor = 8421376
BringToTop	Supported	pb_1.BringToTop = TRUE
Cancel	Supported	pb_1.Cancel = TRUE
Default	Supported	pb_1.Default = TRUE
DisabledName	Supported	pb_1.DisabledName = "d:\pbhelp\controls.bmp"
DragAuto	Supported	pb_1.DragAuto = TRUE
DragIcon	Supported	pb_1.DragIcon = 'c:\archive\arrow.ico'
Enabled	Supported	pb_1.Enabled = TRUE
FaceName	Supported	pb_1.FaceName = ls_value
FontCharSet	Supported	pb_1.FontCharSet = ANSI!
FontFamily	Supported	pb_1.FontFamily = Roman!
FontPitch	Supported	pb_1.FontPitch = Fixed!
Height	Supported	pb_1.Height = li_value
HTextAlign	Supported	pb_1.HtextAlign = "Left!" //Specifies how the text in the control is aligned. Values are: Center! Justify! Left! Right!
Italic	Supported	pb_1.Italic = lb_value
OriginalSize	Supported	pb_1.OriginalSize = FALSE //Specifies whether the width and height properties of a bitmap image (picture) are set to their original values.
PictureName	Supported	pb_1.PictureName = ls_value
Pointer	Supported	pb_1.pointer = 'Arrow!' pb_1.pointer ='d:\archive\IBEAM.BMP.cur'
PowerTipText	Supported	pb_1.PowerTipText = "This button opens a new form"
TabOrder	Supported	pb_1.TabOrder = 10

Property	Support Level	Example Code
Tag	Supported	pb_1.Tag = ls_value
Text	Supported	pb_1.Text = ls_value
TextSize	Supported	pb_1.TextSize = li_value
Underline	Supported	pb_1.Underline = lb_value
Visible	Supported	pb_1.Visible = lb_value
VTextAlign	Supported	pb_1.VTextAlign= Top! //Specifies how the text in the control is aligned. Values are: Bottom!, MultiLine!, Top!, and VCenter! All these values, except for MultiLine!, assume that there is only one line of text.
Weight	Supported	pb_1.Weight = li_value
Width	Supported	pb_1.Width = li_value
X	Supported	pb_1.X = li_value
Y	Supported	pb_1.Y = li_value
ClassDefinition	Unsupported	
Map3DColors	Unsupported	

6.1.1.20.2 Events for PictureButton control

Table 6.60:

Event	Support Level	Example Code
Clicked	Supported	Clicked()
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LoseFocus	Supported	LoseFocus()
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
Help	Unsupported	
Other	Unsupported	

6.1.1.20.3 Functions for PictureButton control

Table 6.61:

Function	Support Level	Example Code
ClassName	Supported	ls_returnvalue = pb_1.ClassName()

Function	Support Level	Example Code
Drag	Supported	pb_1.Drag (Cancel!)
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = pb_1.GetParent()
Hide	Supported	li_returnvalue = pb_1.Hide()
Move	Supported	li_returnvalue = pb_1.Move(li_xpos, li_ypos)
PointerX	Supported	lb_returnvalue = pb_1.PointerX()
PointerY	Supported	lb_returnvalue = pb_1.PointerY()
PostEvent	Supported	lb_returnvalue = pb_1.PostEvent(event)
Resize	Supported	li_returnvalue = pb_1.Resize(li_width, li_height)
SetFocus	Supported	li_returnvalue = pb_1.SetFocus() Unsupported: SetFocus(pb_1)
SetPosition	Supported	li_returnvalue = pb_1.SetPosition(position)
SetRedraw	Supported	li_returnvalue = pb_1.SetRedraw (TRUE)
Show	Supported	li_returnvalue = pb_1.Show()
TriggerEvent	Supported	li_returnvalue = pb_1.TriggerEvent(event)

TypeOf	Supported	if pb_1.TypeOf() = PictureButton! Then
		ls_returnvalue = 'PictureButton!'
		end if

GetContextService	Unsupported	
Print	Unsupported	

6.1.1.21 PictureHyperLink control

Important Requirements

The PictureHyperLink controls can contain images in the following formats:

- Bitmaps, with .BMP extensions
- GIF or animated GIF files, with the .GIF extensions
- JPEG files, with .JPEG or .JPG extensions
- Ico files with .ICO extensions
- Cursor files with .CUR extensions

6.1.1.21.1 Properties for PictureHyperLink control

Table 6.62:

Property	Support Level	Example Code
Border	Supported	phl_1.Border = lb_value
BorderStyle	Supported	phl_1.BorderStyle = StyleBox!

Property	Support Level	Example Code
		phl_1.BorderStyle = StyleLowered! phl_1.BorderStyle = StyleRaised! Note: StyleShadowBox! is unsupported.
BringToTop	Supported	phl_1.BringToTop = TRUE
DragAuto	Supported	
DragIcon	Supported	
Enabled	Supported	phl_1.Enabled = lb_value
FocusRectangle	Supported	phl_1.FocusRectangle = TRUE
Height	Supported	phl_1.Height = li_value
OriginalSize	Supported	phl_1.OriginalSize = lb_value
PictureName	Supported	phl_1.PictureName = ls_value
Pointer	Supported	phl_1= 'Size!' phl_1='d:\archive\IBEAM.BMP.cur'
PowerTipText	Supported	phl_1.PowerTipText = "This button opens a new form"
TabOrder	Supported	phl_1.TabOrder = 100
Tag	Supported	phl_1.Tag = ls_value
URL	Supported	phl_1.URL = "http://www.appeon.com/"
Visible	Supported	phl_1.Visible = lb_value
Width	Supported	phl_1.Width = li_value
X	Supported	phl_1.X = li_value
Y	Supported	phl_1.Y = li_value
ClassDefinition	Unsupported	
Invert	Unsupported	
Map3DColors	Unsupported	

6.1.1.21.2 Events for PictureHyperLink control

Table 6.63:

Event	Support Level	Example Code
Clicked	Supported	Clicked()
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DoubleClicked	Supported	DoubleClicked()
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)

Event	Support Level	Example Code
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LoseFocus	Supported	LoseFocus()
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
Help	Unsupported	
Other	Unsupported	

6.1.1.21.3 Functions for PictureHyperLink control

Table 6.64:

Function	Support Level	Example Code
ClassName	Supported	ls_value = phl_1.ClassName( )
Draw	Supported	li_value = phl_1.Draw(li_x,li_y)
Drag	Supported	phl_1.Drag(Begin!)
GetParent	Supported	PowerObject lpo_value lpo_value = phl_1.GetParent()
Hide	Supported	li_value = phl_1.Hide( )
Move	Supported	li_value = phl_1.Move(li_x,li_y)
PointerX	Supported	li_value = phl_1.PointerX()
PointerY	Supported	li_value = phl_1.PointerY()
PostEvent	Supported	lb_value = phl_1.PostEvent(Clicked!)
Resize	Supported	li_value = phl_1.Resize(li_x,li_y)
SetFocus	Supported	li_value = phl_1.SetFocus( ) Unsupported: SetFocus(phl_1)
SetPicture	Supported	phl_1.SetPicture(Emp_pic)
SetPosition	Supported	li_value = phl_1.SetPosition( )
SetRedraw	Supported	li_returnvalue = phl_1.SetRedraw (FALSE)
Show	Supported	li_value = phl_1.Show( )
TriggerEvent	Supported	li_value = phl_1.TriggerEvent(Clicked!)

TypeOf	Supported	if phl_1.TypeOf( ) = PictureHyperLink! Then
		ls_value = "PictureHyperLink!"
		else
		ls_value = "invalid"
		end if

GetContextService	Unsupported	
Print	Unsupported	

6.1.1.22 PictureListBox control

Important Requirements

Setting a PictureListBox as a Dropdown PictureListBox is not supported.

6.1.1.22.1 Properties for PictureListBox control

Table 6.65:

Property	Support Level	Example Code
Accelerator	Supported	plb_1.Accelerator=67
BackColor	Supported	plb_1.BackColor = ll_value plb_1.BackColor = RGB(192,192,192)
Border	Supported	plb_1.Border = lb_value
BorderStyle	Supported	plb_1.BorderStyle = StyleLowered! plb_1.BorderStyle = StyleRaised! plb_1.BorderStyle = StyleBox! Note: StyleShadowBox! is not supported.
BringToTop	Supported	plb_1.BringToTop = TRUE
DisableNoScroll	Supported	plb_1.DisableNoScroll = lb_value
DragAuto	Supported	plb_1.DragAuto = TRUE
DragIcon	Supported	plb_1.DragIcon ='c:\archive\arrow.ico'
Enabled	Supported	plb_1.Enabled = lb_value
ExtendedSelect	Supported	plb_1.ExtendedSelect = lb_value
FaceName	Supported	plb_1.FaceName = ls_value
FontCharSet	Supported	plb_1.FontCharSet = ANSI!
FontFamily	Supported	plb_1.FontFamily = Roman!
FontPitch	Supported	plb_1.FontPitch = Fixed!
Height	Supported	plb_1.Height = li_value
HScrollBar	Supported	plb_1.Hscrollbar = lb_value
Italic	Supported	plb_1.Italic = lb_value
Item[ ]	Supported	plb_1.Item[] = ls_value
ItemPicture Index[ ]	Supported	plb_1.ItemPictureIndex[ ] = li_value[]
MultiSelect	Supported	plb_1.MultiSelect = lb_value
PictureHeight	Supported	plb_1.PictureHeight = 16
PictureWidth	Supported	plb_1.PictureWidth = 16
PictureName[ ]	Supported	ls_result[] = plb_1.PictureName[li_x[]] Assigning values to part of an array and using the default values for the other part is not supported.

Property	Support Level	Example Code
		For example: *picturename[1] = "aaa.bmp"* picturename[3] = "bbb.bmp" *var pic1 = picutrename[1];* *var pic2 = picturename[3];* A supported example is given as follows: *picturename[1] = "aaa.bmp"* *picturename[2] = "bbb.bmp"* *picturename[3] = "ccc.bmp"* *var pic1 = picutrename[1];* *var pic2 = picturename[2];* *var pic3 = picturename[3];*
Pointer	Supported	plb_1.pointer = 'Icon!' plb_1.pointer ='d:\archive\IBEAM.BMP'
RightToLeft	Supported	plb_1.RightToLeft = TRUE
Sorted	Supported	plb_1.Sorted =lb_value
TabOrder	Supported	plb_1.TabOrder = 110
Tag	Supported	plb_1.Tag = ls_value
TextColor	Supported	plb_1.Textcolor = ll_value
TextSize	Supported	plb_1.TextSize = li_value
Underline	Supported	plb_1.Underline = lb_value
Visible	Supported	plb_1.Visible = lb_value
VScrollBar	Supported	plb_1.VScrollBar = lb_value
Weight	Supported	plb_1.Weight = li_value
Width	Supported	plb_1.Width = li_value
X	Supported	plb_1.X = li_value
Y	Supported	plb_1.Y = li_value
PictureMaskColor	Supported	plb_1.PictureMaskColor= ll_value
ClassDefinition	Unsupported	
ShowList	Unsupported	
TabStop[ ]	Unsupported	

6.1.1.22.2 Events for PictureListBox control

Table 6.66:

Event	Support Level	Example Code
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DoubleClicked	Supported	DoubleClicked()
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LoseFocus	Supported	LoseFocus()
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
SelectionChanged	Supported	SelectionChanged(index)
Help	Unsupported	
Other	Unsupported	

6.1.1.22.3 Functions for PictureListBox control

Table 6.67:

Function	Support Level	Example Code
AddItem	Supported	li_position = plb_1.AddItem(ls_name, li_pic)
AddPicture	Supported	li_pic = plb_1.AddPicture("c:\pics\cardinal.bmp")
ClassName	Supported	ls_value = plb_1.ClassName()
DeleteItem	Supported	plb_1.DeleteItem(li_value)
DeletePicture	Supported	plb_1.DeletePicture (6)
DeletePictures	Supported	plb_1.DeletePictures ()
DirList	Supported	plb_1.DirList("C:\EMPLOYEE*.TXT", 0)
DirSelect	Supported	plb_1.DirSelect(ls_filename)
Drag	Supported	plb_1.Drag(End!)
FindItem	Supported	li_index = plb_1.FindItem ("Delete", 1)
GetParent	Supported	PowerObject lpo_value lpo_value = plb_1.GetParent()
Hide	Supported	li_value = plb_1.Hide()
InsertItem	Supported	plb_1.InsertItem("Run Application", 5) plb_1.InsertItem("Run Application", 5, 4)
Move	Supported	li_value = plb_1.Move(li_x[],li_y[])
PointerX	Supported	li_value = plb_1.PointerX()

Function	Support Level	Example Code
PointerY	Supported	li_value = plb_1.PointerY()
PostEvent	Supported	lb_value = plb_1.PostEvent(Clicked!)
Reset	Supported	li_value = plb_1.Reset()
Resize	Supported	li_value = plb_1.Resize(li_x[],li_y[])
SelectedIndex	Supported	li_index = plb_1.SelectedIndex()
SelectedItem	Supported	li_item = plb_1.SelectedItem()
SelectItem	Supported	li_value = plb_1.SelectItem (li_x[],li_y[])
SetFocus	Supported	li_value = plb_1.SetFocus() Unsupported: SetFocus(plb_1)
SetPosition	Supported	li_returnvalue = plb_1.SetPosition(position)
SetRedraw	Supported	li_returnvalue = plb_1.SetRedraw (FALSE)
SetState	Supported	plb_1.SetState(6, TRUE)
SetTop	Supported	plb_1.SetTop(6)
Show	Supported	li_value = plb_1.Show()
State	Supported	li_item = plb_1.State(3)
Text	Supported	ls_return = plb_1.Text(2)
Top	Supported	li_num = plb_1.Top()
TotalItems	Supported	li_value= plb_1.TotalItems()
TotalSelected	Supported	integer SelectedTotal SelectedTotal = plb_1.TotalSelected()
TriggerEvent	Supported	li_value = plb_1.TriggerEvent(Constructor!)

TypeOf	Supported	if plb_1.typeof()=picturelistbox! Then
		ls_value='picturelistbox!'
		else
		ls_value='isvalid'
		end if

Clear	Unsupported	
Copy	Unsupported	
Cut	Unsupported	
GetContextService	Unsupported	
Paste	Unsupported	
Position	Unsupported	
Print	Unsupported	

6.1.1.23 RadioButton control

6.1.1.23.1 Properties for RadioButton control

Table 6.68:

Property	Support Level	Example Code
Automatic	Supported	rb_1.Automatic = TRUE
BackColor	Supported	rb_1.BackColor = ll_value rb_1.BackColor = RGB(192,192,192)
BorderStyle	Supported	rb_1.BorderStyle = StyleLowered! rb_1.BorderStyle = StyleRaised!
BringToTop	Supported	rb_1.BringToTop = TRUE
Checked	Supported	rb_1.Checked = lb_value
DragAuto	Supported	rb_1.DragAuto = TRUE
DragIcon	Supported	rb_1.DragIcon ='c:\archive\arrow.ico'
Enabled	Supported	rb_1.Enabled = lb_value
FaceName	Supported	rb_1.FaceName = ls_value
FontCharSet	Supported	rb_1.FontCharSet = ANSI!
FontFamily	Supported	rb_1.FontFamily = Roman!
FontPitch	Supported	rb_1.FontPitch = Fixed!
Height	Supported	rb_1.Height = li_value
LeftText	Supported	rb_1.LeftText = TRUE
Italic	Supported	rb_1.Italic = lb_value
Pointer	Supported	rb_1.Pointer ='AppStarting!' rb_1.Pointer='d:\archive\IBEAM.BMP'
RightToLeft	Supported	rb_1.RightToLeft = TRUE
TabOrder	Supported	rb_1.TabOrder = 120
Tag	Supported	rb_1.Tag = ls_value
Text	Supported	rb_1.Text = ls_value
TextColor	Supported	rb_1.TextColor = ll_value
TextSize	Supported	rb_1.TextSize = li_value
Underline	Supported	rb_1.Underline = lb_value
Visible	Supported	rb_1.Visible = lb_value
Weight	Supported	rb_1.Weight = li_value
Width	Supported	rb_1.Width = li_value
X	Supported	rb_1.X = li_value
Y	Supported	rb_1.Y = li_value
ClassDefinition	Unsupported	

6.1.1.23.2 Events for RadioButton control

Table 6.69:

Event	Support Level	Example Code
Clicked	Supported	Clicked()
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LoseFocus	Supported	LoseFocus()
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
Help	Unsupported	
Other	Unsupported	

6.1.1.23.3 Functions for RadioButton control

Table 6.70:

Function	Support Level	Example Code
ClassName	Supported	ls_value = rb_1.ClassName( )
Drag	Supported	rb_1.Drag(Begin!)
GetParent	Supported	PowerObject lpo_value lpo_value = rb_1.GetParent( )
Hide	Supported	li_value = rb_1.Hide( )
Move	Supported	li_value = rb_1.Move(li_x,li_y)
PointerX	Supported	li_vaule = rb_1.PointerX()
PointerY	Supported	li_value = rb_1.PointerY()
PostEvent	Supported	lb_value = rb_1.PostEvent(Clicked!)
Resize	Supported	li_value = rb_1.Resize(li_x,li_y)
SetFocus	Supported	li_value = rb_1.SetFocus( ) Unsupported: SetFocus(rb_1)
SetPosition	Supported	li_value = rb_1.SetPosition(ToTop!) li_value = rb_1.SetPosition(ToBottom!)
SetRedraw	Supported	li_returnvalue = rb_1.SetRedraw (FALSE)
Show	Supported	li_value = rb_1.Show( )
TriggerEvent	Supported	li_value = rb_1.TriggerEvent(Clicked!)
TypeOf	Supported	if rb_1.TypeOf( ) = RadioButton! Then

Function	Support Level	Example Code
		ls_value = "RadioButton!" end if
GetContextService	Unsupported	
Print	Unsupported	

6.1.1.24 Rectangle control

6.1.1.24.1 Properties for Rectangle control

Table 6.71:

Property	Support Level	Example Code
FillColor	Supported	r_1.FillColor = ll_value
FillPattern	Supported	r_1.FillPattern = Diamond!
Height	Supported	r_1.Height = li_value
LineColor	Supported	r_1.LineColor = ll_value
LineStyle	Supported	r_1.LineStyle = Continuous! r_1.LineStyle = Dash! r_1.LineStyle = DashDot! r_1.LineStyle = DashDotDot! r_1.LineStyle = Dot! r_1.LineStyle = Transparent!
LineThickness	Supported	r_1.LineThickness =li_value Note: If LineThickness is greater than one pixel (about four PowerBuilder units), the LineStyle property is forced to Continuous!
Tag	Supported	r_1.Tag = ls_value
Visible	Supported	r_1.Visible = lb_value
Width	Supported	r_1.Width = li_value
X	Supported	r_1.X = li_value
Y	Supported	r_1.Y = li_value
ClassDefinition	Unsupported	

6.1.1.24.2 Events for Rectangle control

Table 6.72:

Event	Support Level	Example Code
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()

6.1.1.24.3 Functions for Rectangle control

Table 6.73:

Function	Support Level	Example Code
ClassName	Supported	ls_value = r_1.ClassName( )
GetParent	Supported	PowerObject lpo_value lpo_value = r_1.GetParent( )
Hide	Supported	li_value = r_1.Hide( )
Move	Supported	li_value = r_1.Move(li_x,li_y)
PostEvent	Supported	lb_value = r_1.PostEvent(Clicked!) Note: The PostEvent function returns 1 if it is successful.
Resize	Supported	lb_value = r_1.Resize(Clicked!)
Show	Supported	li_value = r_1.Show( )
TriggerEvent	Supported	li_value = r_1.TriggerEvent(Clicked!) Note: The TriggerEvent function returns 1 if it is successful.

TypeOf	Supported	if r_1.TypeOf( ) = Rectangle! Then
		li_value = "Rectangle!"
		end if

GetContextService	Unsupported	

6.1.1.25 RichTextEdit control

Differences

1. On the Web, the printed result of the RichText may be different from what you see in the RichText control. This is because the printed result will be reformatted according to the size of the paper.
2. On the Web, an input filed is selected when you click on it, you can then delete or replace it by using the keyboard. This is different from that in PowerBuilder.
3. Because of the following differences, the return value on the Web and in PowerBuilder is different when using the functions that relates to the calculation of length. For example, SelectText, SelectedText, SelectAllText, CopyRTF, PasteRTF, Position, Clear, LineCount, TextLine, Copy, Cut, Position, Paste and SelectLength.
   - A picture will be counted as one character on the Web.
   - A new line character will be counted as one character on the Web.
   - A line ending character will be counted as one character.
   - An input field will be counted as separated characters, therefore the return value will be the length of the content in the input fields or the content of the input field.

4. Appeon adopts Windows RTF specification, which differs from PowerBuilder RTF specification, so the displaying result on the Web may differ from that in PowerBuilder. For example: Copying and pasting the content to the RichText control, the displaying result on the Web is different from that in PowerBuilder; The content of RichTextEdit control saved on the Web differs from that in PowerBuilder. However the pure text displayed is almost the same.

5. A menu does not pop up when right clicking a picture in the RichTextEdit control on the Web. This is different from that in PowerBuilder.

6. When a part of text in an InputField is selected, pressing Tab control replaces the selected text with a white space on the Web while a white space is inserted before the selected text in PowerBuilder.

7. When using CTRL + V to paste a picture to the RichTextEdit control, a shortcut icon will be shown on Web, and when clicked, the picture will display, while pasted in PowerBuilder, the picture will directly display in the RichTextEdit control.

Unsupported

1. After you double click the ruler bar in the RichTextEdit control, Paragraph window will be displayed. On the Web, the Standard tab item in the Paragraph window is unsupported.

2. End identifier is unsupported.

6.1.1.25.1 Properties for RichTextEdit control

Table 6.74:

Property	Support Level	Example Code
Accelerator	Supported	rte_1.Accelerator=67 rte_1.Accelerator = ASC("M")
BackColor	Supported	rte_1.BackColor = RGB(255, 255, 0)
Border	Supported	rte_1.Border = TRUE
BorderStyle	Supported	rte_1.BorderStyle = StyleLowered!
BottomMargin	Supported	rte_1.BottomMargin = 1
BringToTop	Supported	rte_1.BringToTop = TRUE
DisplayOnly	Supported	rte_1.DisplayOnly = TRUE
DocumentName	Supported	rte_1.DocumentName = "Report 1"
DragAuto	Supported	rte_1.DragAuto = TRUE
DragIcon	Supported	rte_1.DragIcon = 'Question!'
Enabled	Supported	rte_1.Enabled = TRUE
HeaderFooter	Supported	
Height	Supported	rte_1.Height = 750
HScrollBar	Supported	rte_1.HScrollBar = TRUE

Property	Support Level	Example Code
		Note: On the Web, the horizontal scrollbar is always displayed even if the property is set to FALSE, However, the scroll box of the scrollbar is displayed only when all of the data cannot be displayed at one time
InputFieldNamesVisible	Supported	rte_1.InputFieldNamesVisible = FALSE
InputFieldsVisible	Supported	rte_1.InputFieldsVisible = TRUE
LeftMargin	Supported	rte_1.LeftMargin = 1
Modified	Supported	
PopMenu	Supported	rte_1.PopMenu = TRUE
Resizable	Supported	rte_1.Resizable = TRUE Note: After setting this property to True, setting Border properties will not be effective.
RightMargin	Supported	rte_1.RightMargin = 1
RulerBar	Supported	rte_1.RulerBar = TRUE
TabBar	Supported	rte_1.TabBar = TRUE
TabOrder	Supported	rte_1.TabOrder = 10
Tag	Supported	rte_1.SetMicroHelp(This.Tag)
ToolBar	Supported	rte_1.ToolBar = TRUE
TopMargin	Supported	rte_1.TopMargin = 1
Visible	Supported	rte_1.Visible = TRUE
VScrollBar	Supported	rte_1.VScrollBar = TRUE Note: On the Web, the vertical scrollbar is always displayed even if the property is set to FALSE. However, the scroll box of the scrollbar is displayed only when the contents of the object are outside the borders.
Width	Supported	rte_1.Width = 750
WordWrap	Supported	rte_1.WordWrap = TRUE Note: 1. When this property is changed from TRUE to FALSE, the content in the RichTextEdit control will be rearranged on the Web. This is different from that in PowerBuilder. 2. If this property is set to False, you cannot execute SetAlignment to set the alignment for the selected paragraphs on the Web and the text will not

Property	Support Level	Example Code
		automatically wrap to the next line even if the line reaches the margin. These are different from that in PowerBuilder. 3. Alignment buttons on the Web are only effective if the WordWrap property is set to True. This is different from that in PowerBuilder.
X	Supported	rte_1.X = 215
Y	Supported	rte_1.Y = 215
ClassDefinition	Unsupported	
ImeMode	Unsupported	
InputFieldBack Color	Unsupported	
Pointer	Unsupported	
PicturesAsFrame	Unsupported	
ReturnsVisible	Unsupported	
SpacesVisible	Unsupported	
TabsVisible	Unsupported	
UndoDepth	Unsupported	

6.1.1.25.2 Events for RichTextEdit control

Table 6.75:

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor
DoubleClicked	Supported	DoubleClicked
GetFocus	Supported	rte_1 = GetFocus()
Key	Supported	IF key = KeyF1! THEN Beep(1) Note: Some keystrokes take no effect in formatting the document in RichText DataWindow. For example, pressing Esc after updating data in an input field restores the data in PowerBuilder but takes no effect on the Web.
LoseFocus	Supported	
Modified	Supported	
MouseDown	Supported	
MouseMove	Supported	
MouseUp	Supported	
RButtonDown	Supported	

Event	Support Level	Example Code
RButtonUp	Supported	
DragDrop	Supported	
DragEnter	Supported	
DragLeave	Supported	
DragWithin	Supported	
FileExists	Unsupported	
Help	Unsupported	
InputFieldSelected	Unsupported	
Other	Unsupported	
PictureSelected	Unsupported	
PrintFooter	Unsupported	
PrintHeader	Unsupported	

6.1.1.25.3 Functions for RichTextEdit control

Table 6.76:

Function	Support Level	Example Code
ClassName	Supported	ls_value = rte_1.ClassName(gd_double)
Clear	Supported	rte_1.Clear() Note: Executing this function on the Web clears the selected text as well as the format from the control. However, in PowerBuilder, it only clears the selected text.
Copy	Supported	li_value = rte_1.Copy()
CopyRTF	Supported	ls_value = rte_1.CopyRTF()
Cut	Supported	rte_1.Cut()
DataSource	Supported	rte_1.DataSource(dw_1)
Drag	Supported	rte_1.Drag(Begin!)
Find	Supported	li_value = rte_1.Find(sle_search.Text, TRUE, & TRUE, FALSE, TRUE)
FindNext	Supported	rte_1.FindNext()
GetAlignment	Supported	l_align = rte_1.GetAlignment()
GetParagraph Setting	Supported	ll_value = rte_1.GetParagraphSetting(Indent!))
GetParent	Supported	luo_value = rte_1.GetParent()
GetSpacing	Supported	l_spacing = rte_1.GetSpacing()
GetTextColor	Supported	ll_color = rte_1.GetTextColor()
GetTextStyle	Supported	lb_value = rte_1.GetTextStyle(Bold!)

Function	Support Level	Example Code
Hide	Supported	rte_1.Hide()
InputField ChangeData	Supported	li_value = rte_1.InputFieldChangeData ("instrument", & lb_instruments.SelectedItem())
InputField CurrentName	Supported	ls_value = rte_1.InputFieldCurrentName()
InputField DeleteCurrent	Supported	li_value = rte_1.InputFieldDeleteCurrent() The input filed that is being edited can be deleted by executing the InputFieldDeleteCurrent function on the Web. But it cannot be deleted in PowerBuilder in that way.
InputField GetData	Supported	ls_value = rte_1.InputFieldGetData(empname)
InputFieldInsert	Supported	ls_value = String(rte_1.InputFieldInsert("lastname")) Note: On the Web, when you click on an input field and then execute InputFieldInsert, the input field is replaced by the new one. In PowerBuilder, the new input field is inserted and the old one will still exist.
InputFieldLocate	Supported	ls_value = rte_1.InputFieldLocate(Last!, "address")
InsertDocument	Supported	li_value = rte_1.InsertDocument("c:\pb\test .rtf", & TRUE, FileTypeRichText!)
InsertPicture	Supported	li_value = rte_1.InsertPicture("c:\windows\earth.bmp")
IsPreview	Supported	lb_value = rte_1.IsPreview()
LineCount	Supported	li_value = rte_1.LineCount()
LineLength	Supported	li_value = rte_1.LineLength()
Move	Supported	rte_1.Move(150, 200)
PageCount	Supported	ls_value = String(rte_1.PageCount())
Paste	Supported	rte_1.Paste() Note: Double-byte characters inserted by executing this function can not be displayed properly.
PasteRTF	Supported	string ls_richtext rte_1.PasteRTF(ls_richtext, Header!)
PointerX	Supported	li_value = rte_1.PointerX()
PointerY	Supported	li_value = rte_1.PointerY()
Position	Supported	l_band = rte_1.Position(ll_startline, ll_startchar, & ll_endline, ll_endchar)
PostEvent	Supported	rte_1.PostEvent(Clicked!)
Preview	Supported	rte_1.Preview(TRUE) Note: When executing this function to display the RichTextEdit control, the content will be rearranged

Function	Support Level	Example Code
		according to the size of the page on the Web. This is different from that in PowerBuilder.
Print	Supported	rte_1.Print(1, "1-5", FALSE, TRUE)
ReplaceText	Supported	rte_1.ReplaceText("Draft") Note: If the text contains double-byte characters, the return value of this function is different from in PowerBuilder.
Resize	Supported	rte_1.Resize(100, 150)
SaveDocument	Supported	rte_1.SaveDocument("c:\test.rtf", FileTypeRichText!) Note: 1. encoding argument is unsupported. 2. The values FileTypePDF!, FileTypeDoc!, FileTypeHTML! of the filetype argument are unsupported.
Scroll	Supported	rte_1.Scroll(4)
ScrollNextPage	Supported	rte_1.ScrollNextPage()
ScrollNextRow	Supported	rte_1.ScrollNextRow()
ScrollPriorPage	Supported	rte_1.ScrollPriorPage()
ScrollPriorRow	Supported	rte_1.ScrollPriorRow()
ScrollToRow	Supported	rte_1.ScrollToRow(25)
SelectedLength	Supported	ll_length = rte_1.SelectedLength()
SelectedLine	Supported	li_value = rte_1.SelectedLine()
SelectedPage	Supported	li_value = rte_1.SelectedPage()
SelectedStart	Supported	li_value = rte_1.SelectedStart()
SelectedText	Supported	ls_value = rte_1.SelectedText() Notes: 1. Executing the this function on the Web will select text and pictures , however the pictures will not appear to be selected. 2. It returns a string containing the selected text when selecting text in the header or footer bands on the Web. This is different from in PowerBuilder.
SelectText	Supported	rte_1.SelectText(1,1, 3,4) Note: 1. When the focus is on an input field, executing the SelectText function will select the input field

Function	Support Level	Example Code
		however the input field will not appear to be selected on the Web. 2. After executing the SelectText function on the Web, if the focus is not at the area where the arguments specify, the focus will be automatically placed at that area. This is different from that in PowerBuilder. 3. If the ReadOnly is set to True, this function still takes effect on the Web.
SelectTextAll	Supported	rte_1.SelectTextAll() Note: If the focus is in the header or footer band, executing "SelectTextAll (detail!)" on the Web returns -1 and no text will be selected. However, in PowerBuilder, the text in the band, which the focus is in, will be selected.
SelectTextLine	Supported	rte_1.SelectTextLine()
SelectTextWord	Supported	rte_1.SelectTextWord()
SetAlignment	Supported	li_value = rte_1.SetAlignment(Right!)
SetFocus	Supported	rte_1.SetFocus()
SetParagraph Setting	Supported	ll_indent = rte_1.SetParagraphSetting(Indent!, 250)
SetPosition	Supported	rte_1.SetPosition(ToTop!)
SetRedraw	Supported	rte_1.SetRedraw(FALSE)
SetSpacing	Supported	rte_1.SetSpacing(Spacing2!)
SetTextColor	Supported	rte_1.SetTextColor(RGB(100, 0, 0))
SetTextStyle	Supported	rte_1.SetTextStyle(TRUE, FALSE, FALSE, & FALSE, TRUE, FALSE) Note: If the RichTextEdit control retrieves the focus by executing the SetFocus or SetText, executing the SetTextStyle function will affect the formatting of the following inputted text on the Web. This is different from that in PowerBuilder.
Show	Supported	rte_1.Show()
ShowHeadFoot	Supported	rte_1.ShowHeadFoot(TRUE)
TextLine	Supported	ls_value = rte_1.TextLine()
TriggerEvent	Supported	rte_1.TriggerEvent(Clicked!)
TypeOf	Supported	rte_1.Typeof()
CanUndo	Unsupported	

Function	Support Level	Example Code
GetContextService	Unsupported	
SelectedColumn	Unsupported	
Undo	Unsupported	

6.1.1.26 RoundRectangle control

6.1.1.26.1 Properties for RoundRectangle control

Table 6.77:

Property	Support Level	Example Code
CornerHeight	Supported	rr_1.CornerHeight = 10
CornerWidth	Supported	rr_1.CornerWidth = 10
FillColor	Supported	rr_1.FillColor=RGB(255,255,0)
FillPattern	Supported	rr_1.FillPattern = Diamond!
Height	Supported	rr_1.Height = 750
LineColor	Supported	rr_1.LineColor = RGB(255,0,0)
LineStyle	Supported	rr_1.LineStyle = Dash!
LineThickness	Supported	rr_1.LineThickness =li_value
Tag	Supported	rr_1.Tag = ls_value
Visible	Supported	rr_1.Visible = True
Width	Supported	rr_1.Width = 750
X	Supported	rr_1.X = 215
Y	Supported	rr_1.Y = 215
ClassDefinition	Unsupported	

6.1.1.26.2 Events for RoundRectangle control

Table 6.78:

Event	Support Level	Example Code
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()

6.1.1.26.3 Functions for RoundRectangle control

Table 6.79:

Function	Support Level	Example Code
ClassName	Supported	ls_value = rr_1.ClassName( )

Function	Support Level	Example Code
GetParent	Supported	PowerObject lpo_value lpo_value = rr_1.GetParent( )
Hide	Supported	li_value = rr_1.Hide( )
Move	Supported	li_value = rr_1.Move(li_x,li_y)
PostEvent	Supported	lb_value = rr_1.PostEvent(Clicked!) Note: The PostEvent function returns 1 if it is successful.
Resize	Supported	lb_value = rr_1.Resize(Clicked!)
Show	Supported	li_value = rr_1.Show( )
TriggerEvent	Supported	li_value = rr_1.TriggerEvent(Clicked!) Note: The TriggerEvent function returns 1 if it is successful.

TypeOf	Supported	if rr_1.TypeOf( ) = RoundRectangle! Then
		li_value = "RoundRectangle!"
		end if

GetContextService	Unsupported	

6.1.1.27 SingleLineEdit control

6.1.1.27.1 Properties for SingleLineEdit control

Table 6.80:

Property	Support Level	Example Code
Accelerator	Supported	sle_1.Accelerato r= 67
AutoHScroll	Supported	sle_1.AutoHScroll = TRUE
BackColor	Supported	sle_1.BackColor = ll_value
Border	Supported	sle_1.Border = lb_value
BorderStyle	Supported	sle_1.BorderStyle = StyleBox! sle_1.BorderStyle = StyleLowered! sle_1.BorderStyle = StyleRaised! Note: StyleShadowBox! is unsupported.
BringToTop	Supported	sle_1.BringToTop = TRUE
DisplayOnly	Supported	sle_1.DisplayOnly = lb_value
DragAuto	Supported	sle_1.DragAuto = TRUE
DragIcon	Supported	sle_1.DragIcon ='c:\archive\arrow.ico'
Enabled	Supported	sle_1.Enabled = lb_value
FaceName	Supported	sle_1.FaceName = ls_value

Property	Support Level	Example Code
FontCharSet	Supported	sle_1.FontCharSet = ANSI!
FontFamily	Supported	sle_1.FontFamily = Roman!
FontPitch	Supported	sle_1.FontPitch = Fixed!
Height	Supported	sle_1.Height = li_value
HideSelection	Supported	sle_1.HideSelection = FALSE
Italic	Supported	sle_1.Italic = lb_value
Limit	Supported	sle_1.Limit = li_value
Password	Supported	sle_1.Password = lb_value
Pointer	Supported	sle_1.pointer= 'Arrow!' sle_1.pointer= 'd:\archive\IBEAM.BMP.cur'
RightToLeft	Supported	sle_1.RightToLeft = TRUE
TabOrder	Supported	sle_1.TabOrder = 130
Tag	Supported	sle_1.Tag = ls_value
Text	Supported	sle_1.Text = ls_value
TextCase	Supported	sle_1.TextCase = AnyCase! sle_1.TextCase = Lower! sle_1.TextCase = Upper!
TextColor	Supported	sle_1.TextColor = ll_value
TextSize	Supported	sle_1.TextSize = li_value
Underline	Supported	sle_1.Underline = lb_value
Visible	Supported	sle_1.Visible = lb_value
Weight	Supported	sle_1.Weight = li_value
Width	Supported	sle_1.Width = li_value
X	Supported	sle_1.X = li_value
Y	Supported	sle_1.Y = li_value
ClassDefinition	Unsupported	

6.1.1.27.2 Events for SingleLineEdit control

Table 6.81:

Event	Support Level	Example Code
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)

Event	Support Level	Example Code
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LoseFocus	Supported	LoseFocus()
Modified	Supported	Modified()
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
Help	Unsupported	
Other	Unsupported	

6.1.1.27.3 Functions for SingleLineEdit control

Table 6.82:

Function	Support Level	Example Code
ClassName	Supported	ls_returnvalue = sle_1.ClassName()
Clear	Supported	li_returnvalue = sle_1.Clear()
Copy	Supported	li_returnvalue = sle_1.Copy()
Cut	Supported	li_returnvalue = sle_1.Cut()
Drag	Supported	sle_1Drag(Cancel!)
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = sle_1.GetParent()
Hide	Supported	li_returnvalue = sle_1.Hide()
Move	Supported	li_returnvalue = sle_1.Move(li_x,li_y)
Paste	Supported	li_returnvalue = sle_1.Paste()
PointerX	Supported	li_returnvalue = sle_1.PointerX()
PointerY	Supported	li_returnvalue = sle_1.PointerY()
Position	Supported	sle_1.Position()
PostEvent	Supported	lb_returnvalue = sle_1.PostEvent(event)
ReplaceText	Supported	li_returnvalue = sle_1.ReplaceText(text)
Resize	Supported	li_returnvalue = sle_1.Resize(li_x,li_y)
SelectedLength	Supported	li_returnvalue = sle_1.SelectedLength() If the control is off screen, the execution of the function may have an incorrect return result.
SelectedStart	Supported	li_returnvalue = sle_1.SelectedStart() If the control is off screen, the execution of the function may have an incorrect return result.
SelectedText	Supported	ls_returnvalue = sle_1.SelectedText() If the control is off screen, the execution of the function may have an incorrect return result.

Function	Support Level	Example Code
SelectText	Supported	li_returnvalue = sle_1.SelectText(start, length) If the control is off screen, the execution of the function may have an incorrect return result.
SetFocus	Supported	li_returnvalue = sle_1.SetFocus() Unsupported: SetFocus(sle_1)
SetPosition	Supported	li_returnvalue = sle_1.SetPosition(position)
SetRedraw	Supported	li_return= sle_1.SetRedraw(FALSE)
Show	Supported	li_returnvalue = sle_1.Show()
TriggerEvent	Supported	li_returnvalue = sle_1.TriggerEvent(event)

TypeOf	Supported	if sle_1.TypeOf() = SingleLineEdit! Then
		ls_returnvalue = "SingleLineEdit!"
		end if

Undo	Supported	li_returnvalue = sle_1.Undo()
CanUndo	Unsupported	
GetContextService	Unsupported	
Print	Unsupported	

6.1.1.28 StaticHyperLink control

6.1.1.28.1 Properties for StaticHyperLink control

Table 6.83:

Property	Support Level	Example Code
Alignment	Supported	shl_1.Alignment = ls_value
BackColor	Supported	shl_1.BackColor = ll_value shl_1.BackColor = RGB(192,192,192)
Border	Supported	shl_1.Border = lb_value
BorderColor	Supported	shl_1.BorderColor = ll_value
BorderStyle	Supported	shl_1.BorderStyle = StyleBox! shl_1.BorderStyle = StyleLowered! shl_1.BorderStyle = StyleRaised! Note: StyleShadowBox! is unsupported.
BringToTop	Supported	shl_1.BringToTop = TRUE
DisabledLook	Supported	
DragAuto	Supported	shl_1.DragAuto = TRUE
DragIcon	Supported	shl_1.DragIcon = 'c:\archive\arrow.ico'

Property	Support Level	Example Code
Enabled	Supported	shl_1.Enabled = lb_value Note: in PowerBuilder, setting the Enabled property of StaticHyperLink to TRUE or FALSE has no effect. The text always displays as it is.
FaceName	Supported	shl_1.FaceName = ls_value
FillPattern	Supported	shl_1.FillPattern = Diamond!
FocusRectangle	Supported	shl_1.FocusRectangle = TRUE
FontCharSet	Supported	shl_1.FontCharSet = ANSI!
FontFamily	Supported	shl_1.FontFamily = Roman!
FontPitch	Supported	shl_1.FontPitch = Fixed!
Height	Supported	shl_1.Height = li_value
Italic	Supported	shl_1.Italic = lb_value
Pointer	Supported	shl_1.Pointer = 'HourGlass!' shl_1.Pointer='d:\archive\IBEAM.BMP.cur'
RightToLeft	Supported	shl_1.RightToLeft = TRUE
TabOrder	Supported	shl_1.TabOrder = 140
Tag	Supported	shl_1.Tag = ls_value
Text	Supported	shl_1.Text = ls_value
TextColor	Supported	shl_1.TextColor = ll_value
TextSize	Supported	shl_1.TextSize = li_value
Underline	Supported	shl_1.Underline = lb_value
URL	Supported	shl_1.URL = "http://www.appeon.com"
Visible	Supported	shl_1.Visible = lb_value
Weight	Supported	shl_1.Weight = li_value
Width	Supported	shl_1.Width = li_value
X	Supported	shl_1.X = li_value
Y	Supported	shl_1.Y = li_value
ClassDefinition	Unsupported	

6.1.1.28.2 Events for StaticHyperLink control

Table 6.84:

Event	Support Level	Example Code
Clicked	Supported	Clicked()
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DoubleClicked	Supported	DoubleClicked()

Event	Support Level	Example Code
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LoseFocus	Supported	LoseFocus()
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
Help	Unsupported	
Other	Unsupported	

6.1.1.28.3 Functions for StaticHyperLink control

Table 6.85:

Function	Support Level	Example Code
ClassName	Supported	ls_returnvalue = shl_1.ClassName()
Drag	Supported	shl_1.Drag(End!)
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = shl_1.GetParent()
Hide	Supported	li_returnvalue = shl_1.Hide()
Move	Supported	li_returnvalue = shl_1.Move(li_x,li_y)
PointerX	Supported	li_returnvalue = shl_1.PointerX()
PointerY	Supported	li_returnvalue = shl_1.PointerY()
PostEvent	Supported	lb_returnvalue = shl_1.PostEvent(event)
Resize	Supported	li_returnvalue = shl_1.Resize(li_x,li_y)
SetFocus	Supported	li_returnvalue = shl_1.SetFocus() Unsupported: SetFocus(shl_1)
SetPosition	Supported	li_returnvalue = shl_1.SetPosition(position)
SetRedraw	Supported	shl_1.SetRedraw(FALSE)
Show	Supported	li_returnvalue = shl_1.Show()
TriggerEvent	Supported	li_returnvalue = shl_1.TriggerEvent(event)
TypeOf	Supported	if shl_1.typeof() = StaticHyperLink! then ls_returnvalue = 'StaticHyperLink!' end if
GetContextService	Unsupported	
Print	Unsupported	

6.1.1.29 StaticText control

6.1.1.29.1 Properties for StaticText control

Table 6.86:

Property	Support Level	Example Code
Alignment	Supported	st_1.Alignment = ls_value
BackColor	Supported	st_1.BackColor = ll_value st_1.BackColor = RGB(192,192,192)
Border	Supported	st_1.Border = lb_value
BorderColor	Supported	st_1.BorderColor = ll_value
BorderStyle	Supported	st_1.BorderStyle = StyleBox! st_1.BorderStyle = StyleLowered! st_1.BorderStyle = StyleRaised! Note: StyleShadowBox! is unsupported.
BringToTop	Supported	st_1.BringToTop = TRUE
DisabledLook	Supported	
DragAuto	Supported	st_1.DragAuto = TRUE
DragIcon	Supported	st_1.DragIcon = 'c:\archive\arrow.ico'
Enabled	Supported	st_1.Enabled = lb_value
FaceName	Supported	st_1.FaceName = ls_value
FillPattern	Supported	st_1.FillPattern = Diamond!
FocusRectangle	Supported	st_1.FocusRectangle = TRUE
FontCharSet	Supported	st_1.FontCharSet = ANSI!
FontFamily	Supported	st_1.FontFamily = Roman!
FontPitch	Supported	st_1.FontPitch = Fixed!
Height	Supported	st_1.Height = li_value
Italic	Supported	st_1.Italic = lb_value
Pointer	Supported	st_1.Pointer = 'AppStarting!' st_1.Pointer = 'd:\archive\IBEAM.BMP.cur'
RightToLeft	Supported	st_1.RightToLeft = TRUE
TabOrder	Supported	st_1.TabOrder = 140
Tag	Supported	st_1.Tag = ls_value
Text	Supported	st_1.Text = ls_value
TextColor	Supported	st_1.TextColor = ll_value
TextSize	Supported	st_1.TextSize = li_value
Underline	Supported	st_1.Underline = lb_value
Visible	Supported	st_1.Visible = lb_value

Property	Support Level	Example Code
Weight	Supported	st_1.Weight = li_value
Width	Supported	st_1.Width = li_value
X	Supported	st_1.X = li_value
Y	Supported	st_1.Y = li_value
ClassDefinition	Unsupported	

6.1.1.29.2 Events for StaticText control

Table 6.87:

Event	Support Level	Example Code
Clicked	Supported	Clicked()
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DoubleClicked	Supported	DoubleClicked()
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LoseFocus	Supported	LoseFocus()
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
Help	Unsupported	
Other	Unsupported	

6.1.1.29.3 Functions for StaticText control

Table 6.88:

Function	Support Level	Example Code
ClassName	Supported	ls_returnvalue = st_1.ClassName()
Drag	Supported	st_1.Drag(Cancel!)
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = st_1.GetParent()
Hide	Supported	li_returnvalue = st_1.Hide()
Move	Supported	li_returnvalue = st_1.Move(li_x,li_y)
PointerX	Supported	li_returnvalue = st_1.PointerX()
PointerY	Supported	li_returnvalue = st_1.PointerY()
PostEvent	Supported	lb_returnvalue = st_1.PostEvent(event)
Resize	Supported	li_returnvalue = st_1.Resize(li_x,li_y)

Function	Support Level	Example Code
SetFocus	Supported	li_returnvalue = st_1.SetFocus() Unsupported: SetFocus(st_1)
SetPosition	Supported	li_returnvalue = st_1.SetPosition(position)
Show	Supported	li_returnvalue = st_1.Show()
SetRedraw	Supported	li_returnvalue = st_1.SetRedraw (FALSE)
TriggerEvent	Supported	li_returnvalue = st_1.TriggerEvent(event)

TypeOf	Supported	if st_1.typeof() = StaticText! Then
		ls_returnvalue = 'StaticText!'
		end if

GetContextService	Unsupported	
Print	Unsupported	

6.1.1.30 Tab control

Differences

- If the MultiLine property of a Tab control is set to True, and the width of the control is exactly the same as the width of all the tabs in it, the tabs may be displayed in two lines on the Web although one line in PowerBuilder. To avoid the difference, specify a width for the Tab control that is bigger than the width of all the tabs, and do more testing till the tabs on the Web can be displayed in the same way as in PowerBuilder.
- In the following scenario, setting the RaggedRight to TRUE will not take effect:
  1. The MultiLine property is disabled.
  2. The MultiLine property is enabled but all tabs can be located in a row.
- In PowerBuilder, if the BoldSelectText property is enabled, only the label of the selected tab page will be displayed in bold type. However, on the Web, the labels of all tabs appear in bold type. Remember to adjust the width of each tab page so that the text can be displayed within the label when enabling this property for the Web.
- In the following scenario, the width of tabs on the Web may be different from in PowerBuilder. When the MultiLine property is enabled and the RaggedRight property is disabled, the largest width of a tab is over W/N (W stands for the total width of all tabs, N stands for the total number of all tabs.)

6.1.1.30.1 Properties for Tab control

Table 6.89:

Property	Support Level	Example Code
Alignment	Supported	tab_1.Alignment = 'Center!'
BackColor	Supported	tab_1.BackColor = ll_value tab_1.BackColor = RGB(192,192,192)

Property	Support Level	Example Code
BoldSelectedText	Supported	tab_1.BoldSelectedText = TRUE
BringToTop	Supported	tab_1.BringToTop = TRUE
Control[]	Supported	Userobject luo_1 luo_1 = tab_1.Control[1] Note: this property cannot be dynamically changed.
CreateOnDemand	Supported	tab_1.CreateOnDemand = TRUE
DragAuto	Supported	tab_1.DragAuto = True
DragIcon	Supported	tab_1.DragIcon ='c:\archive\arrow.ico'
Enabled	Supported	tab_1.Enabled = FALSE
FaceName	Supported	tab_1.FaceName = ls_value
FixedWidth	Supported	tab_1.FixedWidth = lb_value
FocusOnButton Down	Supported	tab_1.FocusOnButtonDown = TRUE
FontCharSet	Supported	tab_1.FontCharSet = ANSI!
FontFamily	Supported	tab_1.FontFamily = Roman!
FontPitch	Supported	tab_1.FontPitch = Fixed!
Height	Supported	tab_1.Height = li_value
Italic	Supported	tab_1.Italic = lb_value
MultiLine	Supported	tab_1.Multiline = TRUE
PictureOnRight	Supported	tab_1.PictureOnRight = TRUE
Pointer	Supported	tab_1.Pointer='SizeNWSE!' tab_1.Pointer='d:\archive\IBEAM.BMP.cur'
PowerTips	Supported	tab_1.PowerTips = TRUE
RaggedRight	Supported	tab_1.RaggedRight=TRUE
SelectedTab	Supported	tab_1.SelectedTab = li_value
ShowPicture	Supported	tab_1.ShowPicture = TRUE
ShowText	Supported	tab_1.ShowText = lb_value
TabOrder	Supported	tab_1.TabOrder = 150
TabPosition	Supported	tab_1.TabPosition = TabsOnTop! The TabPosition property is partially supported: • Supported values of the TabPosition property: TabsOnBottom!,TabsOnTop! • Unsupported values of the TabPosition property: TabsOnBottomAndTop!, TabsOnLeftAndRight!, TabsOnRightAndLeft!,

Property	Support Level	Example Code
		TabsOnTopAndBottom!,TabsOnLeft!, TabsOnRight!
Tag	Supported	tab_1.Tag = ls_value
TextSize	Supported	tab_1.TextSize = li_value
Underline	Supported	tab_1.Underline = lb_value
Visible	Supported	tab_1.Visible = lb_value
Weight	Supported	tab_1.Weight = li_value
Width	Supported	tab_1.Width = li_value
X	Supported	tab_1.X = li_value
Y	Supported	tab_1.Y = li_value
ClassDefinition	Unsupported	
PerpendicularText	Unsupported	

6.1.1.30.2 Events for Tab control

Table 6.90:

Event	Support Level	Example Code
Clicked	Supported	Clicked(index)
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DoubleClicked	Supported	DoubleClicked()
DragDrop	Supported	DragDrop(source,index)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source,index)
GetFocus	Supported	GetFocus()
Key	Supported	Key(key,keyflags)
LoseFocus	Supported	LoseFocus()
3Clicked	Supported	RightClicked(index)
RightDouble Clicked	Supported	RightDoubleClicked(index)
SelectionChanged	Supported	SelectionChanged(oldindex,newindex)
Selection Changing	Supported	SelectionChanging(oldindex,newindex)
Help	Unsupported	
Other	Unsupported	

6.1.1.30.3 Functions for Tab control

Table 6.91:

Function	Support Level	Example Code
ClassName	Supported	ls_returnvalue = tab_1.ClassName()
CloseTab	Supported	li_returnvalue = tab_1.CloseTab()
Drag	Supported	li_returnvalue = tab_1.Drag()
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = tab_1.GetParent()
Hide	Supported	li_returnvalue = tab_1.Hide()
Move	Supported	li_returnvalue = tab_1.Move(li_x,li_y)
MoveTab	Supported	li_returnvalue = tab_1.MoveTab(li_x,li_y)
OpenTab	Supported	li_returnvalue = tab_1.OpenTab()
OpenTabWith Parm	Supported	li_returnvalue = tab_1.OpenTabWithParm()
PointerX	Supported	li_returnvalue = tab_1.PointerX()
PointerY	Supported	li_returnvalue = tab_1.PointerY()
PostEvent	Supported	li_returnvalue = tab_1.PostEvent(Clicked!)
Resize	Supported	li_returnvalue = tab_1.Resize(li_x,li_y)
SelectTab	Supported	li_returnvalue = tab_1.SelectTab(tabidentifier)
SetFocus	Supported	li_returnvalue = tab_1.SetFocus() Unsupported: SetFocus(tab_1)
SetPosition	Supported	li_returnvalue = tab_1.SetPosition(position)
SetRedraw	Supported	li_return= tab_1.SetRedraw(FALSE)
Show	Supported	li_returnvalue = tab_1.Show()
TabTriggerEvent	Supported	li_returnvalue = tab_address.TabTriggerEvent("ue_display")
TriggerEvent	Supported	li_returnvalue = tab_1.TriggerEvent(event)

TypeOf	Supported	if tab_1.typeof() = Tab! then
		ls_returnvalue = 'Tab!'
		endif

GetContextService	Unsupported	
Print	Unsupported	
TabPostEvent	Unsupported	

6.1.1.31 TreeView control

Differences

The length of the converted TreeView labels has minor differences to those in PowerBuilder.

6.1.1.31.1 Properties for TreeView control

Table 6.92:

Property	Support Level	Example Code
Accelerator	Supported	tv_1.Accelerator = 67
BackColor	Supported	tv_1.BackColor = ll_value tv_1.BackColor = RGB(192,192,192)
Border	Supported	tv_1.Border = lb_value
BorderStyle	Supported	tv_1.BorderStyle=StyleRaised! tv_1.BorderStyle=StyleBox! tv_1.BorderStyle=StyleLowered! Note: StyleShadowBox! is not supported.
BringToTop	Supported	tv_1.BringToTop = TRUE
CheckBoxes	Supported	tv_1.CheckBoxes = lb_value
DeleteItems	Supported	tv_1.DeleteItems = lb_value
DisableDragDrop	Supported	tv_1.DisableDragDrop = TRUE
DragAuto	Supported	tv_1.DragAuto = TRUE
DragIcon	Supported	tv_1.DragIcon='c:\archive\arrow.ico'
EditLabels	Supported	tv_1.EditLabels = lb_value
Enabled	Supported	tv_1.Enabled = lb_value
FontCharSet	Supported	tv_1.FontCharSet = ANSI!
FontFamily	Supported	tv_1.FontFamily = Roman!
FontPitch	Supported	tv_1.FontPitch = Fixed!
FaceName	Supported	tv_1.FaceName = ls_value
FullRowSelect	Supported	tv_1.FullRowSelect = TRUE
HasButtons	Supported	tv_1.HasButtons = lb_value
HasLines	Supported	tv_1.HasLines = lb_value
Height	Supported	tv_1.Height = li_value
HideSelection	Supported	tv_1.HideSelection = FALSE
Indent	Supported	tv_1.Indent = 100
Italic	Supported	tv_1.Italic = lb_value
LinesAtRoot	Supported	tv_1.LinesAtRoot = lb_value
PictureHeight	Supported	tv_1.PictureHeight = 16
PictureMaskColor	Supported	tv_1.PictureMaskColor = RGB(255, 255, 0)
PictureName	Supported	tv_1.PictureName = ls_value Assigning values to part of an array and using the default values for the other part is not supported. For example:

Property	Support Level	Example Code

		`picturename[1] = "aaa.bmp"`
		`picturename[3] = "bbb.bmp"`
		`var pic1 = picutrename[1];`
		`var pic2 = picturename[3];`
		
		A supported example is given as follows:
		
		`picturename[1] = "aaa.bmp"`
		`picturename[2] = "bbb.bmp"`
		`picturename[3] = "ccc.bmp" var`
		`pic1 = picutrename[1];`
		`var pic2 = picturename[2];`
		`var pic3 = picturename[3];`

PictureWidth	Supported	tv_1.PictureWidth = 16
Pointer	Supported	tv_1.Pointer ='UPArrow!' tv_1.Pointer ='d:\archive\IBEAM.BMP.cur'
SingleExpand	Supported	tv_1.SingleExpand = TRUE
StatePicture Height	Supported	tv_1.StatePictureHeight = 16
StatePicture MaskColor	Supported	tv_1.StatePictureMaskColor = RGB(255, 255, 0)

StatePicture Name[ ]	Supported	ls_value[] = tv_1.StatePictureName[]
		
		Assigning values to part of an array and using the default values for the other part is not supported.
		
		For example:
		
		`statepicturename[1] = "aaa.bmp"`
		`statepicturename[3] = "bbb.bmp"`
		`var pic1 = statepicutrename[1];`
		`var pic2 = statepicturename[3];`
		
		A supported example is given as follows:
		
		`statepicturename[1] = "aaa.bmp"`
		`statepicturename[2] = "bbb.bmp"`
		`statepicturename[3] = "ccc.bmp"`
		`var pic1 = statepicutrename[1];`
		`var pic2 = statepicturename[2];`
		`varpic3 = statepicturename[3];`

StatePictureWidth	Supported	tv_1.StatePictureWidth = 16
TabOrder	Supported	tv_1.TabOrder = 160
Tag	Supported	tv_1.Tag = ls_value
TextColor	Supported	tv_1.TextColor = ll_value
TextSize	Supported	tv_1.TextSize = li_value
ToolTips	Supported	tv_1.ToolTips = TRUE
TrackSelect	Supported	tv_1.TrackSelect = TRUE
Underline	Supported	tv_1.Underline = lb_value

Property	Support Level	Example Code
Visible	Supported	tv_1.Visible = lb_value
Weight	Supported	tv_1.Weight = li_value
Width	Supported	tv_1.Width = li_value
X	Supported	tv_1.X = li_value
Y	Supported	tv_1.Y = li_value
ClassDefinition	Unsupported	
SortType	Unsupported	

6.1.1.31.2 Events for TreeView control

Table 6.93:

Event	Support Level	Example Code
BeginDrag	Supported	BeginDrag(handle)
BeginLabelEdit	Supported	BeginLabelEdit(handle)
BeginRightDrag	Supported	BeginRightDrag(handle)
Clicked	Supported	Clicked(handle)
Constructor	Supported	Constructor()
DeleteItem	Supported	DeleteItem(handle)
Destructor	Supported	Destructor()
DoubleClicked	Supported	DoubleClicked(handle)
DragDrop	Supported	DragDrop(source,handle)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source,handle)
EndLabelEdit	Supported	EndLabelEdit(handle,newtext)
GetFocus	Supported	GetFocus()
Key	Supported	Key(key,keyflags)
ItemCollapsed	Supported	ItemCollapsed(handle)
ItemCollapsing	Supported	ItemCollapsing(handle)
ItemExpanded	Supported	ItemExpanded(handle)
ItemExpanding	Supported	ItemExpanding(handle)
ItemPopulate	Supported	ItemPopulate(handle)
LoseFocus	Supported	LoseFocus()
RightClicked	Supported	RightClicked(handle)
RightDouble Clicked	Supported	RightDoubleClicked(handle)
SelectionChanged	Supported	SelectionChanged(oldhandle,newhandle)

Event	Support Level	Example Code
Selection Changing	Supported	SelectionChanging(oldhandle,newhandle)
Help	Unsupported	
Other	Unsupported	
Sort	Unsupported	

6.1.1.31.3 Functions for TreeView control

Table 6.94:

Function	Support Level	Example Code
AddPicture	Supported	li_returnvalue = tv_1.AddPicture(picturename)
AddStatePicture	Supported	li_returnvalue = tv_1.AddStatePicture(picturename)
ClassName	Supported	ls_returnvalue = tv_1.ClassName()
CollapseItem	Supported	li_returnvalue = tv_1.CollapseItem(itemhandle)
DeleteItem	Supported	li_returnvalue = tv_1.DeleteItem(itemhandle)
DeletePicture	Supported	li_returnvalue = tv_1.DeletePicture(index)
DeletePictures	Supported	li_returnvalue = tv_1.DeletePictures()
DeleteState Picture	Supported	li_returnvalue = tv_1.DeleteStatePicture(index)
DeleteStatePictures	Supported	li_returnvalue = tv_1.DeleteStatePictures()
Drag	Supported	tv_1.Drag(End!)
EditLabel	Supported	li_returnvalue = tv_1.EditLabel(itemhandle)
ExpandAll	Supported	li_returnvalue = tv_1.ExpandAll(itemhandle)
ExpandItem	Supported	li_returnvalue = tv_1.ExpandItem(itemhandle)
FindItem	Supported	ll_returnvalue = tv_1.FindItem(navigationcode, itemhandle) Note: the argument *navigationcode* can only use the following values: RootTreeItem!, NextTreeItem!, PreviousTreeItem!, ParentTreeItem!, ChildTreeItem!, FirstVisibleTreeItem!, NextVisibleTreeItem!, PreviousVisibleTreeItem!, CurrentTreeItem!. The *navigationcode* value cannot be DropHighlightTreeItem!.
GetItem	Supported	li_returnvalue = tv_1.GetItem(itemhandle,tvi_ret)
GetItemAtPointer	Supported	li_index = tv_1.GetItematPointer( )
GetParent	Supported	lobj_returnvalue = tv_1.GetParent( )
Hide	Supported	li_returnvalue = tv_1.Hide( )
InsertItem	Supported	ll_returnvalue = tv_1.InsertItem(handleparent,handleafter,item)

Function	Support Level	Example Code
		ll_returnvalue = tv_1.InsertItem(handleparent,handleafter,label,pictureindex)
InsertItemFirst	Supported	ll_returnvalue = tv_1.InsertItemFirst(handleparent,item) ll_returnvalue = tv_1.InsertItemFirst(handleparent,label,pictureindex)
InsertItemLast	Supported	ll_returnvalue = tv_1.InsertItemLast(handleparent,item) ll_returnvalue = tv_1.InsertItemLast(handleparent, label, pictureindex)
InsertItemSort	Supported	tv_1.InsertItemSort(ll_roothandle, l_tvi)
Move	Supported	li_returnvalue = tv_1.Move(li_xpos,li_ypos)
PointerX	Supported	li_returnvalue = tv_1.PointerX()
PointerY	Supported	li_returnvalue = tv_1.PointerY()
PostEvent	Supported	lb_returnvalue = tv_1.PostEvent(event)
Resize	Supported	li_returnvalue = tv_1.Resize(li_width,li_height)
SelectItem	Supported	ll_returnvalue = tv_1.SelectItem(itemhandle)
SetFirstVisible	Supported	li_tvret = tv_1.SetFirstVisible(ll_tvi)
SetFocus	Supported	li_returnvalue = tv_1.SetFocus() Unsupported: SetFocus(tv_1)
SetItem	Supported	li_returnvalue = tv_1.SetItem(itemhandle,item)
SetLevelPictures	Supported	tv_1.SetLevelPictures(3, li_level, li_level, & li_level, li_level)
SetOverlayPicture	Supported	tv_1.SetOverlayPicture(1, index)
SetPosition	Supported	li_returnvalue = tv_1.SetPosition(position)
SetRedraw	Supported	li_return= tv_1.SetRedraw(True)
Show	Supported	li_returnvalue = tv_1.Show()
Sort	Supported	li_returnvalue = tv_1.Sort()
SortAll	Supported	tv_1.SortAll(ll_tvi , Ascending!)
TriggerEvent	Supported	li_returnvalue = tv_1.TriggerEvent(event)

TypeOf	Supported	if tv_1.TypeOf() = TreeView! Then
		ls_returnvalue = 'TreeView!'
		end if

GetContextService	Unsupported	
Print	Unsupported	
SetDropHighlight	Unsupported	

6.1.1.32 VProgressBar control

6.1.1.32.1 Properties for VProgressBar control

Table 6.95:

Property	Support Level	Example Code
BringToTop	Supported	vpb_1.BringToTop = TRUE
DragAuto	Supported	vpb_1.DragAuto = TRUE
DragIcon	Supported	vpb_1.DragIcon = 'Question!'
Height	Supported	vpb_1.Height = li_value
MaxPosition	Supported	vpb_1.MaxPosition = 120
MinPosition	Supported	vpb_1.MinPosition = 20
Pointer	Supported	vpb_1.Pointer = 'Beam!' vpb_1.Pointer ='d:\archive\IBEAM.BMP.cur'
Position	Supported	vpb_1.Position = 50
StdStep	Supported	vpb_1.setstep = 20
SmoothScroll	Supported	vpb_1.SmoothScroll = True
TabOrder	Supported	vpb_1.TabOrder = 20
Tag	Supported	vpb_1.Tag = "VProgressBar control"
Visible	Supported	vpb_1.Visible = TRUE
Width	Supported	vpb_1.Width = 700
X	Supported	vpb_1.X = 280
Y	Supported	vpb_1.Y = 1280
ClassDefinition	Unsupported	

6.1.1.32.2 Events for VProgressBar control

Table 6.96:

Event	Support Level	Example Code
Clicked	Supported	Clicked(flags,xpos,ypos)
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DoubleClicked	Supported	DoubleClicked(flags,xpos,ypos)
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()

Event	Support Level	Example Code
LoseFocus	Supported	LoseFocus()
RButtonDown	Supported	RButtonDown(flags,xpos,ypos)
Help	Unsupported	
Other	Unsupported	

6.1.1.32.3 Functions for VProgressBar control

Table 6.97:

Function	Support Level	Example Code
ClassName	Supported	ls_classnm = vpb_1.classname()
Drag	Supported	li_return = vpb_1.drag(Begin!)
GetParent	Supported	lpo_return = vpb_1.GetParent()
Hide	Supported	li_return = vpb_1.Hide()
Move	Supported	li_return = vpb_1.Move(li_x,li_y)
OffsetPos	Supported	li_return = vpb_1.OffsetPos(10)
PointerX	Supported	li_return = vpb_1.PointerX()
PointerY	Supported	li_return = vpb_1.PointerY()
PostEvent	Supported	vpb_1.PostEvent(Clicked!)
Resize	Supported	li_return = vpb_1.Resize(li_x, li_y)
SetFocus	Supported	li_return = vpb_1.SetFocus()
SetPosition	Supported	li_return = vpb_1.SetPosition(ToTop!)
SetRange	Supported	li_return = vpb_1.SetRange(1, 10)
SetRedraw	Supported	li_return = vpb_1.SetRedraw(FALSE)
Show	Supported	li_return = vpb_1.Show()
StepIt	Supported	li_return = vpb_1.StepIt( )
TriggerEvent	Supported	li_return = vpb_1.TriggerEvent(Clicked!)
TypeOf	Supported	type_obj = vpb_1.typeof( )
GetContextService	Unsupported	
Print	Unsupported	

6.1.1.33 VScrollBar control

6.1.1.33.1 Properties for VScrollBar control

Table 6.98:

Property	Support Level	Example Code
BringToTop	Supported	vsb_1.BringToTop = TRUE

Property	Support Level	Example Code
DragAuto	Supported	vsb_1.DragAuto = TRUE
DragIcon	Supported	vsb_1.DragIcon = 'Question!'
Height	Supported	vsb_1.Height = li_value
MaxPosition	Supported	vsb_1.MaxPosition = 120
MinPosition	Supported	vsb_1.MinPosition = 20
Pointer	Supported	vsb_1.Pointer = 'Beam!' vsb_1.Pointer ='d:\archive\IBEAM.BMP.cur'
Position	Supported	vsb_1.Position = 50
StdWidth	Supported	vsb_1.StdHeight = True
TabOrder	Supported	vsb_1.TabOrder = 20
Tag	Supported	vsb_1.Tag = "VScrollBar control"
Visible	Supported	vsb_1.Visible = TRUE
Width	Supported	vsb_1.Width = 700
X	Supported	vsb_1.X = 280
Y	Supported	vsb_1.Y = 1280
ClassDefinition	Unsupported	

6.1.1.33.2 Events for VScrollBar control

Table 6.99:

Event	Support Level	Example Code
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LineDown	Supported	LineDown()
LineUp	Supported	LineUp()
LoseFocus	Supported	LoseFocus()
Moved	Supported	Moved(scrollpos)
PageDown	Supported	PageDown()
PageUp	Supported	PageUp()
RButtonDown	Supported	RButtonDown(flags,xpos,ypos)
Help	Unsupported	

Event	Support Level	Example Code
Other	Unsupported	

6.1.1.33.3 Functions for VScrollBar control

Table 6.100:

Function	Support Level	Example Code
ClassName	Supported	ls_classnm = vsb_1.classname()
Drag	Supported	li_return = vsb_1.drag(Begin!)
GetParent	Supported	lpo_return = vsb_1.GetParent()
Hide	Supported	li_return = vsb_1.Hide()
Move	Supported	li_return = vsb_1.Move(li_x,li_y)
PointerX	Supported	li_return = vsb_1.PointerX()
PointerY	Supported	li_return = vsb_1.PointerY()
PostEvent	Supported	vsb_1.PostEvent(Clicked!)
Resize	Supported	li_return = vsb_1.Resize(li_x, li_y)
SetFocus	Supported	li_return = vsb_1.SetFocus()
SetPosition	Supported	li_return = vsb_1.SetPosition(ToTop!)
SetRedraw	Supported	li_return = vsb_1.SetRedraw(FALSE)
Show	Supported	li_return = vsb_1.Show()
TriggerEvent	Supported	li_return = vsb_1.TriggerEvent(Clicked!)
TypeOf	Supported	type_obj = vsb_1.typeof()
GetContextService	Unsupported	
Print	Unsupported	

6.1.1.34 VTrackBar control

6.1.1.34.1 Properties for VTrackBar control

Table 6.101:

Property	Support Level	Example Code
BringToTop	Supported	vtb_1.BringToTop = TRUE
DragAuto	Supported	vtb_1.DragAuto = TRUE
DragIcon	Supported	vtb_1.DragIcon = 'Question!'
Height	Supported	vtb_1.Height = li_value
LineSize	Supported	vtb_1.LineSize = li_value
MaxPosition	Supported	vtb_1.MaxPosition = 120
MinPosition	Supported	vtb_1.MinPosition = 20

Property	Support Level	Example Code
PageSize	Supported	vtb_1.PageSize = li_value
Pointer	Supported	vtb_1.Pointer = 'Beam!' vtb_1.Pointer ='d:\archive\IBEAM.BMP.cur'
Position	Supported	vtb_1.Position = 50
Slider	Supported	vtb_1.Slider = TRUE
SliderSize	Supported	vtb_1.SliderSize = li_value
TabOrder	Supported	vtb_1.TabOrder = 20
Tag	Supported	vtb_1.Tag = ls_tag
TickFrequency	Supported	vtb_1.TickFrequency = li_value
TickMarks	Supported	vtb_1.TickMarks = VTicksOnBottom!
Visible	Supported	vtb_1.Visible = TRUE
Width	Supported	vtb_1.Width = li_value
X	Supported	vtb_1.X = li_x
Y	Supported	vtb_1.Y = li_y
ClassDefinition	Unsupported	

6.1.1.34.2 Events for VTrackBar control

Table 6.102:

Event	Support Level	Example Code
Constructor	Supported	Constructor()
Destructor	Supported	Destructor()
DragDrop	Supported	DragDrop(source)
DragEnter	Supported	DragEnter(source)
DragLeave	Supported	DragLeave(source)
DragWithin	Supported	DragWithin(source)
GetFocus	Supported	GetFocus()
LineDown	Supported	LineDown()
LineUp	Supported	LineUp()
LoseFocus	Supported	LoseFocus()
Moved	Supported	Moved(scrollpos)
PageDown	Supported	PageDown()
PageUp	Supported	PageUp()
RButtonDown	Supported	RButtonDown(flags,xpos,ypos)
Help	Unsupported	
Other	Unsupported	

6.1.1.34.3 Functions for VTrackBar control

Table 6.103:

Function	Support Level	Example Code
ClassName	Supported	ls_classnm = vtb_1.classname()
Drag	Supported	li_return = vtb_1.drag(Begin!)
GetParent	Supported	lpo_return = vtb_1.GetParent()
Hide	Supported	li_return = vtb_1.Hide()
Move	Supported	li_return = vtb_1.Move(li_x,li_y)
PointerX	Supported	li_return = vtb_1.PointerX()
PointerY	Supported	li_return = vtb_1.PointerY()
PostEvent	Supported	vtb_1.PostEvent(Clicked!)
Resize	Supported	li_return = vtb_1.Resize(li_x, li_y)
SelectionRange	Supported	li_return = vtb_1.SelectionRange(30, 70)
SetFocus	Supported	li_return = vtb_1.SetFocus()
SetPosition	Supported	li_return = vtb_1.SetPosition(ToTop!)
SetRedraw	Supported	li_return = vtb_1.SetRedraw(FALSE)
Show	Supported	li_return = vtb_1.Show()
TriggerEvent	Supported	li_return = vtb_1.TriggerEvent(Clicked!)
TypeOf	Supported	type_obj = vtb_1.typeof()
GetContextService	Unsupported	
Print	Unsupported	

6.1.1.35 Window control

Important Requirements

- There are six window types: Child, Main, MDI, MDIHelp, Popup, and Response.
- Window object cannot be dynamically created (for example, using the CREATE statement); it must be defined in the PowerBuilder painter.
- Main Window can have an associated menu.
- User-defined window events and control events are supported.
- For more information relating to windows refer to the following links:

  Window types

  Window variables

  Opening and closing windows

  User operation in windows

6.1.1.35.1 Properties for Window control

Table 6.104:

Property	Support Level	Example Code
BackColor	Supported	w_1.BackColor = RGB(0, 0, 255)
Border	Supported	w_1.Border = TRUE
BringToTop	Supported	w_1.BringToTop = TRUE
Center	Supported	w_1.Center = TRUE Note: This property is unsupported for an MDI frame window.
Control [ ]	Supported	Control[] graphicobject lobj_control lobj_control = w_1.control[1] Note: This property cannot be dynamically changed.
ControlMenu	Supported	w_1.ControlMenu= TRUE
Enabled	Supported	w_1.Enabled = TRUE
Height	Supported	w_1.Height = 750 //Specifies the height of the window w_1 Note: The Height property is unsupported for an MDI frame window.
HScrollBar	Supported	w_1.HScrollBar = TRUE
Icon	Supported	PowerBuilder and Web behaves differently if no icon is specified for the window: • For PowerBuilder, the icon (if any) specified for the application will be displayed as the icon for the window; • For Web, the Windows default icon will be displayed as the icon for the window.
MaxBox	Supported	w_1.MaxBox= TRUE
MenuID	Supported	Menu menuvar = w_1.MenuID
MenuName	Supported	ls_value = w_1.MenuName
MinBox	Supported	w_1.MinBox = TRUE
Pointer	Supported	w_1.pointer = 'Icon!' w_1.pointer ='d:\archive\IBEAM.BMP.cur'
Resizable	Supported	w_1.Resizable = TRUE Notes: 1. For MDI frame window, the Resizable property will always be TRUE.

Property	Support Level	Example Code
		2. At the execution of the ArrangeSheet function, no matter whether the Resizable property of each open sheet is set to TRUE or FALSE, all open sheets will be tiled and resized so that they do not overlap. 3. After setting this property to True, setting Border properties will not be effective.
RightToLeft	Supported	w_1.RightToLeft = TRUE Note: If the RightToLeft property of an MDI Window is set to Yes, the content in the window except for the content in the status bar displays from right to left.
Tag	Supported	w_1.SetMicroHelp(This.Tag) //Set or get the Tag property
Title	Supported	w_1.Title = "Monthly Report"
TitleBar	Supported	w_1.TitleBar = TRUE
ToolbarVisible	Supported	w_1.ToolbarVisible = TRUE Note: The ToolBarVisible is unsupported for Popup and Main windows.
Visible	Supported	w_1.Visible = TRUE Note: This property is unsupported for sheets whose window type is Response, Popup or Main windows.
VScrollBar	Supported	w_1.VScrollBar = TRUE

Width	Supported	w_1.Width = 750 //Set or get the Width property
		Note: The Width property is unsupported for an MDI frame window.

WindowState	Supported	w_1.WindowState = Maximized! Note: On the Web, the initial window state for an MDI frame window is not based on the WindowState property. It will be decided by the Internet Explorer state.

WindowType	Supported	IF dw_1.WindowType=Main! THEN
		MessaMessageBox("Window Type", "This is a main window.")
		END IF

X	Supported	w_1.X = 215 //Set or get the X property Notes: The X property is not supported for the MDI frame window.
Y	Supported	w_1.Y = 215

Property	Support Level	Example Code
		//Set or get the Y property Notes: The Y property is not supported for the MDI frame window.
ClassDefinition	Unsupported	
ClientEdge	Unsupported	
ColumnsPerPage	Unsupported	
ContextHelp	Unsupported	
KeyboardIcon	Unsupported	
LinesPerPage	Unsupported	
PaletteWindow	Unsupported	
ToolbarAlignment	Unsupported	
ToolbarHeight	Unsupported	
ToolbarWidth	Unsupported	
ToolbarX	Unsupported	
ToolbarY	Unsupported	
UnitsPerColumn	Unsupported	
UnitsPerLine	Unsupported	

6.1.1.35.2 Events for Window control

Table 6.105:

Event	Support Level	Example Code
Activate	Supported	Activate Note: Activate and Deactivate events are unsupported for dialogue boxes.
Clicked	Supported	Clicked
Close	Supported	Close Note: the Close event is handled the same way as Post in Appeon. In the Web application, the return value of the Close event may be different from that in PowerBuilder.
CloseQuery	Supported	CloseQuery
Deactivate	Supported	Deactivate Notes: 1. With the Web applications, the Activate/Deactivate events will not be triggered if the user shift the focus from a window in the current application to a place outside the application.

Event	Support Level	Example Code
		2. Activate and Deactivate events are unsupported for dialogue boxes.
DoubleClicked	Supported	DoubleClicked
DragDrop	Supported	DragDrop
DragEnter	Supported	DragEnter
DragLeave	Supported	DragLeave
DragWithin	Supported	DragWithin
Hide	Supported	Hide
HotLinkAlarm	Supported	HotLinkAlarm
Key	Supported	Key
MouseDown	Supported	MouseDown
MouseMove	Supported	MouseMove
MouseUp	Supported	MouseUp
Open	Supported	Open
RButtonDown	Supported	RButtonDown
Resize	Supported	Resize
Show	Supported	Show
Timer	Supported	Timer
Help	Unsupported	
Other	Unsupported	
RemoteExec	Unsupported	
Remote HotLinkStart	Unsupported	
Remote HotLinkStop	Unsupported	
RemoteRequest	Unsupported	
RemoteSend	Unsupported	
SystemKey	Unsupported	
ToolbarMoved	Unsupported	

6.1.1.35.3 Functions for Window control

Table 6.106:

Function	Support Level	Example Code
ArrangeSheets	Supported	w_mdi.ArrangeSheets(arrangetype)
ChangeMenu	Supported	Parent.ChangeMenu(m_test_menu2) Notes:

Function	Support Level	Example Code
		1. It is unsupported to use a menu item as the *menuname* argument. 2. The menu you want to make the current menu and the current menu cannot be the same menu.
ClassName	Supported	ls_value = w_test.ClassName()
CloseChannel	Supported	w_test.CloseChannel()
CloseUserObject	Supported	w_test.CloseUserObject()
ExecRemote	Suppported	ExecRemote("[Save()]", "Excel", "REGION.XLS")
GetActiveSheet	Supported	w_test. GetActiveSheet()
GetDataDDE	Supported	w_test.GetDataDDE(Str20)
GetDataDDE Origin	Supported	string ls_name GetCommandDDEOrigin(ls_name)
GetFirstSheet	Supported	w_test. GetFirstSheet()
GetNextSheet	Supported	w_test. GetNextSheet()
GetRemote	Supported	GetRemote("[Save()]", "Excel", "REGION.XLS")
GetParent	Supported	w_test.GetParent()
Hide	Supported	Parent.Hide()
Move	Supported	w_test.Move(X,Y)
OpenChannel	Supported	long handle handle = OpenChannel("Excel", "REGION.XLS")
OpenUserObject	Supported	w_test.OpenUserObject() Note: The userobjecttype argument cannot be a system control.
OpenUserObject WithParm	Supported	w_test.OpenUserObjectWithParm()
ParentWindow	Supported	w_parent = w_test.parentwindow()
PointerX	Supported	li_dist = w_city.PointerX()
PointerY	Supported	li_dist = w_city.PointerY()
PostEvent	Supported	w_main.PostEvent(Clicked!)
Resize	Supported	w_test.Resize(300,250)

RespondRemote	Supported	IF GetDataDDE(Value) = 1 THEN
		RespondRemote(TRUE)
		END IF

SetFocus	Supported	w_test.SetFocus() Unsupported: SetFocus(w_test)
SetMicroHelp	Supported	w_test.SetMicroHelp ("Preview")

Function	Support Level	Example Code
SetPosition	Supported	w_test.SetPosition(position [, precedingwindow]) Note: The window with property TopMost may not always be the top one in Appeon.
SetRedraw	Supported	li_return= w_test.SetRedraw(FALSE)
SetRemote	Supported	SetRemote("R5C7", "4500", handle)
Show	Supported	Parent.Show()
StartHotLink	Supported	StartHotLink("Any", "MyPBApp", "Any")
StopHotLink	Supported	StopHotLink("Any", "MyPBApp", "Any")
TriggerEvent	Supported	w_test.TriggerEvent(close!)

TypeOf	Supported	if this.typeof()=window! then
		messagebox("show", "test typeof()")
		end if

WorkSpaceHeight	Supported	Height = W_employee.WorkSpaceHeight()
WorkSpaceWidth	Supported	Width = W_employee.WorkSpaceWidth()
WorkSpaceX	Supported	w_test.WorkSpaceX()
WorkSpaceY	Supported	w_test.WorkSpaceY()
GetCommand DDE	Unsupported	
GetCommand DDEOrigin	Unsupported	
GetContextService	Unsupported	
GetToolbar	Unsupported	
GetToolbarPos	Unsupported	
Print	Unsupported	
SetDataDDE	Unsupported	
SetToolbar	Unsupported	
SetToolbarPos	Unsupported	
StartServerDDE	Unsupported	
StopServerDDE	Unsupported	

6.1.1.35.4 Window types

The six window types, including Child, Main, MDI, MDIHelp, Popup, and Response, are supported well by Appeon with the following behavior differences.

- For a Web application, an MDI sheet will be always behind the windows that are opened with the function Open.
- For a Web application, events that should be triggered by clicking the Title bar of an MDI frame window will be triggered by clicking the Title bar of Internet Explorer.

- **Child:** In PowerBuilder, a Child window is always above its parent window. With Web applications, a Child window can be behind its parent window.

6.1.1.35.5 Window variables

- Window type arrays are supported, and window type variable that is a structure member is supported.
- The declaration and assignment of window variable are all supported.
- It is supported to dynamically create a window instance by using CREATE.

  For example, the following syntax is supported:

```
window lwin_onewindow
lwin_onewindow = create Window
lwin_onewindow.title = "This is a city processing window"
```

6.1.1.35.6 Opening and closing windows

Opening Windows

- Open(windowvar, windowtype{, parent})

  Refer to the Window Functions section of System Functions for the supported syntax for opening the windows.

  In the Web application, if window A is opened by the Open function whereas window B is opened by the OpenSheet function, window A will always display over window B.

- When a window is open, the Constructor events for the instance variables and controls are executed in the same order on the Web as in PowerBuilder:

  1. Executing Constructor events for all instance variables is prior to executing Constructor events for all controls.
  2. For the instance variables/controls, the events are triggered by the following order: outside -> inside and top -> bottom.
  3. If a control is a container (tab/tab page/visual user object), the Constructor event for the container is triggered first. Then all the controls in the container are subsequently triggered.

     Events triggered for the window object follows the order Open >> Resize >> Show >> Activate.

- The same rule applies in PowerBuilder: the front-to-back order of all the controls in the window is determined by the Control[] property of the window and the BringToTop property of each specific control.

Closing Windows

- Refer to the Window Functions section for the supported syntax for closing the windows.

- When a window is closed, the Destructor events of the variables and controls in the window are executed in the same order on the Web as in PowerBuilder:

  1. The Destructor events for the controls are triggered in the same order as the Constructor events.

  2. Executing Destructor events for all controls happens prior to executing Destructor events for all instance variables.

  3. The Destructor events for the instance variables are triggered in the opposite order as the Constructor events.

6.1.1.35.7 User operation in windows

Supported

The following major user operations are supported.

- Maximize
- Restore
- Minimize
- Single-click
- Double-click
- Drag
- Close
- Switching between controls in the window with the Tab key
- Switch between windows by pressing the Ctrl + Tab key

Unsupported

- It is unsupported to scroll a window using the PageUp/PageDown, upArrow/downArrow, or leftArrow/rightArrow keys
- It is unsupported to switch between controls in a window by pressing the Enter/UpArrow/DownArrow/LeftArrow/RightArrow key

6.1.2 Unsupported controls

The following are the controls that Appeon does not support:

- InkEdit control
- InkPicture control

- Animation control

6.2 System objects

Supported

- There can only be one Application object in an application.
- The following are the system objects that Appeon supports:

Table 6.107:

Application	DataStore	Connection	CORBAObject
DataWindowChild	DragObject	DWObject	DynamicStagingArea
DynamicDescriptionArea	Environment	GraphicObject	Graph
grAxis	grDispAttr	Inet	InternetResult
ListViewItem	MenuCascade	mailFileDescription	mailMessage
mailRecipient	mailSession	MDIClient	Menu
Message	NonVisualObject	OLEObject	PowerObject
Timing	Transaction Object	TreeViewItem	UserObject
WindowObject			

- DragObject, GraphicObject, PowerObject and WindowObject cannot be dynamically created (for example, by using the CREATE statement); they must be defined as static objects, created in PowerBuilder painter.

Unsupported

- It is strongly recommended that for a given application, no objects have the same name.
- The following are the system objects that Appeon does not support:

Table 6.108:

ADOResultSet	ArrayBounds	ClassDefinition	ClassDefinitionObject
ConnectionInfo	ConnectObject	ContextInformation	ContextKeyword
CORBABadTypeContext	CORBABadInvorder	CORBABadOperation	CORBABadParam
CORBABadTypeCode	CORBACommFailure	CORBACurrent	CORBADataConversion
CORBAFreeMem	CORBAImpLimit	CORBAInitialize	CORBAInternal
CORBAIntFrePos	CORBAInvalidTransaction	CORBAInvFlag	CORBAInvIdentInvOBJRef
CORBAMarshal	CORBANoImplement	CORBANoMemory	CORBANoPermission
CORBANoResources	CORBANoResponse	CORBAOBJAdapter	CORBAObjectNoTexist
CORBAPersistStore	CORBASystemException	CORBATransactionRequired	CORBATransactionRolledback

CORBATranslent	CORBAUnion	CORBAUnknown	CORBAUser Exception
CPlusPlus	DivideByZeroError	DWRuntimeError	Enumeration Definition
EnumerationItem Definition	Error	ErrorLogging	Exception
ExtObject	JaguarORB	NullObjectError	OLERuntimeError
OLEStorage	OLEStream	OLETxnObject	OMControl
OMCustomControl	OMEmbeddedControl	OMObject	OMStorage
OMStream	ORB	PBTocppObject	Pipeline
ProfileCall	ProfileClass	ProfileLine	ProfileRoutine
Profiling	RemoteObject	ResultSet	ResultSets
RuntimeError	ScriptDefinition	Service	SimpleTypeDefinition
SSLCallBack	SSLServiceProvider	SystemFunctions	Throwable
TraceActivityNode	TraceBeginEnd	TraceError	TraceESQL
TraceFile	TraceGarbageCollect	TraceLine	TraceObject
TraceRoutine	TraceTree	TraceTreeError	TraceTreeESQL
TraceTreeGarbageCollect	TraceTreeLine	TraceTreeNode	TraceTreeObject
TraceTreeRoutine	TraceTreeUser	TraceUser	TransactionServer
Transport	TypeDefinition	VariableCardinality Definition	VariableDefinition

6.2.1 Supported Objects

6.2.1.1 Application object

Important Requirements

- There can only be one Application object in an application.
- An application cannot be given the same name as a PowerBuilder control type. For example, naming an application "datawindow", "editmask" or "commandbutton" is not supported.
- An application name cannot be a reserved word in JavaScript. See Identifiers for more information.
- A commandline argument in the application Open event is supported.
- It is recommended that the use of Application object is limited to the following well-supported actions:

  1. Launching login screens.
  2. Opening and closing MDI, SDI, Main, Response and Popup windows.

3. Creating instances of one or more transaction objects.

Additional properties of Application object

- You can specify an icon for the application, but the icon will be displayed in PowerBuilder only and will not be displayed on Web.
- The type of SQLCA can be a transaction user object.
- The type of SQLSA can be a DynamicStagingArea user object.
- The type of SQLDA can be a DynamicDescriptionArea user object.
- The variable type MESSAGE can use the default value "message" or a user-defined message object, and you can also customize the Message object by defining a class user object inherited from the built-in Message object.
- The specification of Error will be ignored.

6.2.1.1.1 Properties for Application object

Table 6.109:

Property	Support Level	Example Code
AppName	Supported	String ls_AppName ls_AppName =app_1.AppName //Get the AppName property
ClassDefinition	Unsupported	
DDETimeOut	Unsupported	
DisplayName	Supported	String ls_DisplayName ls_DisplayName = app_1.DisplayName
DWMessageTitle	Unsupported	
FreeDBLibraries	Unsupported	
MicroHelpDefault	Supported	app_1.MicroHelpDefault = 'Ready'
RightToLeft	Supported	app_1.RightToLeft = TRUE
ToolbarFrameTitle	Supported	ls_value = app_1.ToolbarFrameTitle
ToolbarPopMenu Tex	Unsupported	
ToolbarSheetTitle	Supported	ls_value =app_1.ToolbarSheetTitle
ToolbarTips	Supported	lb_value = app_1.ToolbarTips
ToolbarText	Supported	lb_value = app_1.ToolbarText
ToolbarUser Control	Supported	

6.2.1.1.2 Events for Application object

Table 6.110:

Event	Support Level	Example Code
Close	Supported	Close
Idle	Supported	Idle
Open	Supported	Open Note: The *commandline* argument is supported. The Commandline argument cannot include continuous number signs ("#").
ConnectionBegin	Unsupported	
ConnectionEnd	Unsupported	
SystemError	Unsupported	

6.2.1.1.3 Functions for Application object

Table 6.111:

Function	Support Level	Example Code
ClassName	Supported	ls_class = l_app.ClassName()
GetParent	Supported	lobj_parent = l_app.GetParent()
PostEvent	Supported	l_app.PostEvent (Clicked!) l_app.PostEvent ("Clicked!")
TriggerEvent	Supported	l_app.TriggerEvent (Clicked!) l_app.TriggerEvent ("Clicked!")

TypeOf	Supported	string ls_applicationtype
		if l_app.TypeOf()=Application! Then
		ls_applicationtype = "Application!"
		end if

GetContextService	Unsupported	
SetLibraryList	Unsupported	
SetTransPool	Unsupported	
Note: in PowerBuilder 9 and above, GetLibrary and SetLibrary are global functions.		

6.2.1.2 Connection object

Important Requirements

Connection object can be created dynamically using the CREATE statement.

6.2.1.2.1 Properties for Connection object

Table 6.112:

Property	Support Level	Example Code
Application	Supported	ls_name = myconnect.application

Property	Support Level	Example Code
Driver	Supported	ls_driver = myconnect.Driver Myconnect.Driver = "jaguar"
ErrCode	Supported	ll_rc = myconnect.errcode
ErrText	Supported	ll_rc = myconnect.errtext
Location	Supported	ls_location = myconnect.location myconnect.location = "192.0.0.217:9000"
UserID	Supported	ls_userid = myconnect.UserID myconnect.UserID = "jagadmin"
Password	Supported	ls_pwd = myconnect.Password myconnect.Password = ""
ClassDefinition	Unsupported	
ConnectString	Unsupported	
Handle	Unsupported	
Options	Unsupported	
Trace	Unsupported	

6.2.1.2.2 Events for Connection object

Table 6.113:

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor
Error	Unsupported	

6.2.1.2.3 Functions for Connection object

Table 6.114:

Function	Support Level	Example Code
ClassName	Supported	ls_name = Myconnect.ClassName()
ConnectToServer	Supported	ll_rc = Myconnect.ConnectToServer() Note: The return value is always 0 on the Web.
CreateInstance	Supported	ll_rc = Myconnect.CreateInstance(ln_tools,"simpletest/nvo_tools")
DisconnectServer	Supported	ll_rc = Myconnect.disconnectserver() Note: The return value is always 0 on the Web.
GetParent	Supported	lobj_parent = Myconnect.GetParent()
PostEvent	Supported	lb_rc = Myconnect.PostEvent(event)

Function	Support Level	Example Code
TriggerEvent	Supported	li_rc = Myconnect.TriggerEvent(event)

TypeOf	Supported	IF Myconnect.TypeOf() = Connection! THEN
		ls_rc = "Connection"
		END IF

GetContextService	Unsupported	
GetServerInfo	Unsupported	
Lookup	Unsupported	
RemoteStop Connection	Unsupported	
RemoteStop Listening	Unsupported	

6.2.1.3 CORBAObject object

Important Requirements

The CORBAObject object can be created dynamically using the CREATE statement.

6.2.1.3.1 Properties for CORBAObject object

Table 6.115:

Property	Support Level	Example Code
ClassDefinition	Unsupported	

6.2.1.3.2 Events for CORBAObject object

Table 6.116:

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor

6.2.1.3.3 Functions for CORBAObject object

Table 6.117:

Function	Support Level	Example Code
ClassName	Supported	ls_name = my_corbaobj.ClassName()
GetParent	Supported	lobj_parent = my_corbaobj.GetParent()
PostEvent	Supported	lb_rc = my_corbaobj.PostEvent(event)
TriggerEvent	Supported	li_rc = my_corbaobj.TriggerEvent(event)
TypeOf	Supported	IF my_corbaobj.TypeOf() = CorbaObject! THEN ls_rc = "Connection"

Function	Support Level	Example Code
		END IF
_Is_A	Unsupported	
_Narrow	Unsupported	
GetContextService	Unsupported	

6.2.1.4 DynamicDescriptionArea object

6.2.1.4.1 Properties for DynamicDescriptionArea object

Table 6.118:

Property	Support Level	Example Code
NumInputs	Supported	Integer li_input li_input = SQLDA.NumInputs
NumOutputs	Supported	Integer li_output li_output = SQLDA.NumOutputs
InParmType[]	Supported	if SQLDA.InParmType[3] = TypeInteger! then ...
OutParmType[]	Supported	CHOOSE CASE SQLDA.OutParmType[li_Idx] On the Web, the value of this property may differ from that in PowerBuilder.
ClassDefinition	Unsupported	

6.2.1.4.2 Events for DynamicDescriptionArea object

Table 6.119:

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor

6.2.1.4.3 Functions for DynamicDescriptionArea object

Table 6.120:

Function	Support Level	Example Code
ClassName	Supported	ls_class = dsa_1.ClassName()
GetDynamicDate	Supported	ls_Value = String(SQLDA.GetDynamicDate(li_Idx))
GetDynamic DateTime	Supported	ls_Value = String(SQLDA.GetDynamicDateTime(li_Idx))
GetDynamic Number	Supported	ls_Value = String(SQLDA.GetDynamicNumber(li_Idx))
GetDynamic String	Supported	ls_Value = SQLDA.GetDynamicString(li_Idx)

Function	Support Level	Example Code
GetDynamicTime	Supported	ls_Value = String(SQLDA.GetDynamicTime(li_Idx))
GetParent	Supported	PowerObject lobj_parent_name lobj_parent_name = this.GetParent ()
PostEvent	Supported	this.PostEvent (Clicked!) this.PostEvent ("Clicked")
SetDynamicParm	Supported	SQLDA.SetDynamicParm(1,2004)
TriggerEvent	Supported	this.TriggerEvent (Clicked!) this.TriggerEvent ("Clicked")
TypeOf	Supported	SQLSA.TypeOf()
GetContextService	Unsupported	

6.2.1.5 DynamicStagingArea object

Important Requirements

The DynamicStagingArea object can be created dynamically using the CREATE statement.

6.2.1.5.1 Properties for DynamicStagingArea object

Table 6.121:

Property	Support Level	Example Code
ClassDefinition	Unsupported	

6.2.1.5.2 Events for DynamicStagingArea object

Table 6.122:

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor

6.2.1.5.3 Functions for DynamicStagingArea object

Table 6.123:

Function	Support Level	Example Code
ClassName	Supported	ls_class = dsa_1.ClassName()
Destructor	Supported	Destructor
Constructor	Supported	Constructor
GetParent	Supported	lobj_parent_name = this.GetParent ()
PostEvent	Supported	this.PostEvent (Clicked!) this.PostEvent ("Clicked!")

Function	Support Level	Example Code
TriggerEvent	Supported	this.TriggerEvent (Clicked!) this.TriggerEvent ("Clicked!")
TypeOf()	Supported	SQLSA.TypeOf()
GetContextService	Unsupported	

6.2.1.6 Environment object

6.2.1.6.1 Properties for Environment object

Table 6.124:

Property	Support Level	Example Code
CharSet	Supported	
CPUType	Supported	
Language	Supported	
MachineCode	Supported	
OSFixesRevision	Supported	
OSMajorRevision	Supported	
OSMinorRevision	Supported	
PBFixesRevision	Supported	
PBMajorRevision	Supported	
PBMinorRevision	Supported	
NumberOfColors	Supported	
ScreenHeight	Supported	
ScreenWidth	Supported	
OSType	Supported	
PBType	Supported	
Win16	Supported	
ClassDefinition	Unsupported	

6.2.1.6.2 Functions for Environment object

Table 6.125:

Function	Support Level	Example Code
ClassName	Supported	string varname varname = ClassName(gd_double)
TypeOf	Supported	dw_1.Typeof()
GetContextService	Unsupported	
GetParent	Unsupported	

6.2.1.7 Graph object

6.2.1.7.1 Properties for Graph object

Table 6.126:

Property	Support Level	Example Code
BackColor	Supported	gr_1.BackColor = RGB(255, 255, 0)
Border	Supported	gr_1.Border = TRUE
BorderStyle	Supported	gr_1.BorderStyle = StyleLowered!
BringToTop	Supported	gr_1.BringToTop = TRUE
Category	Supported	gr_1.Category.Label = "Types of Products"
CategorySort	Supported	gr_1.CategorySort = Unsorted! Note: If the values of Category are multibyte characters (for example, Chinese characters), they are sorted by internal statement number on the Web.
Depth	Supported	gr_1.Depth = 50
DragAuto	Supported	gr_1.DragAuto = TRUE
DragIcon	Supported	gr_1.DragIcon = 'Question!'
Elevation	Supported	gr_1.Elevation = 35
Enabled	Supported	gr_1.Enabled = TRUE
FocusRectangle	Supported	gr_1.FocusRectangle = TRUE
GraphType	Supported	gr_1.GraphType= AreaGraph! Note: The following graph types are unsupported: Area3D!, Bar3DGraph!, Col3DGraph!, Line3D!, Pie3D!
Height	Supported	gr_1.Height = 750
Legend	Supported	gr_1.Legend = AtBottom!
LegendDispAttr	Supported	
OverlapPercent	Supported	gr_1.OverlapPercent = 10
Perspective	Supported	gr_1.Perspective = 25
PieDispAttr	Supported	
Pointer	Supported	gr_1.Pointer = 'Beam!' gr_1.Pointer = 'd:\archive\IBEAM.BMP'
Rotation	Supported	gr_1.Rotation = -45
Series	Supported	gr_1.Series.Scaletype = Log10!
SeriesSort	Supported	gr_1.SeriesSort = Unsorted! Note: 1. The result of executing this property on the Web may differ from that in PowerBuilder. This is

Property	Support Level	Example Code
		because changing other properties may affect the SeriesSort property in PowerBuilder but it does not have such effect on the Web. 2. If the values of Category are multibyte characters (for example, Chinese characters), they are sorted by internal statement numbers on the Web.
ShadeColor	Supported	dw_1.Object.gr_1.ShadeColor = 16600000
Spacing	Supported	gr_1.Spacing = 120
TabOrder	Supported	gr_1.TabOrder = 10
Tag	Supported	gr_1.SetMicroHelp(This.Tag)
TextColor	Supported	gr_1.Series.DispAttr.TextColor = RGB(0,0,255)
Title	Supported	gr_1.TitleBar = TRUE gr_1.Title = "Monthly Report"
TitleDispAttr	Supported	
Values	Supported	ls_value = dw_1.Object.gr_1.Values
Visible	Supported	gr_1.Visible = TRUE
Width	Supported	gr_1.Width = 750
X	Supported	gr_1.X = 215
Y	Supported	gr_1.Y= 400
ClassDefinition	Unsupported	

6.2.1.7.2 Events for Graph object

Table 6.127:

Event	Support Level	Example Code
Clicked	Supported	Clicked
Constructor	Supported	Constructor
Destructor	Supported	Destructor
DoubleClicked	Supported	DoubleClicked
DragDrop	Supported	DragDrop
DragEnter	Supported	DragEnter
DragLeave	Supported	GragLeave
DragWithin	Supported	DragWithin
GetFocus	Supported	GetFocus
LoseFocus	Supported	LoseFocus
RButtonDown	Supported	RButtonDown
Help	Unsupported	
Other	Unsupported	

6.2.1.7.3 Functions for Graph object

Table 6.128:

Function	Support Level	Example Code
AddCategory	Supported	gr_1.AddCategory("PCs")
AddData	Supported	ll_value = gr_1.FindSeries("Costs") gr_1.AddData(ll_value, 12, 3)
AddSeries	Supported	li_value = gr_1.AddSeries("Costs")
CategoryCount	Supported	li_value = gr_1.CategoryCount()
CategoryName	Supported	ls_value = gr_1.CategoryName(5)
ClassName	Supported	ls_value= gr_1.ClassName(gd_double)
Clipboard	Supported	gr_1.Clipboard()
DataCount	Supported	ll_value = gr_1.DataCount("Costs")
DeleteCategory	Supported	gr_1.DeleteCategory(CategName)
DeleteData	Supported	gr_1.DeleteData(SeriesNbr, 7)
DeleteSeries	Supported	gr_1.DeleteSeries(ls_name)
Drag	Supported	gr_1.Drag(Begin!)
FindCategory	Supported	li_value=gr_1.FindCategory(sle_ctgry.Text)
FindSeries	Supported	li_value =gr_1.FindSeries(sle_series.Text)
GetData	Supported	data_value = gr_1.GetData(SeriesNbr, ItemNbr, xValue!)
GetDataPie Explode	Supported	gr_1.GetDataPieExplode(series, datapoint, percentage) Note: This function takes effect only in the pie graph. In other graphs the return value of GetDataPieExplode is -1.
GetDataStyle	Supported	ll_color = gr_emp_data.GetDataStyle(SeriesNbr, 6, Foreground!, color_nbr) On the Web, the GetDataStyle function returns 1 if the specified data point and its series contain data, otherwise it returns -1. This is different from that in PowerBuilder.
GetDataValue	Supported	rtn = gr_1.GetDataValue(SeriesNbr, ItemNbr, data_value)
GetParent	Supported	luo_value = gr_1.GetParent()
GetSeriesStyle	Supported	gr_1.GetSeriesStyle(SeriesName, & line_style, line_width)
Hide	Supported	gr_1.Hide()
ImportClipboard	Supported	gr_1.ImportClipboard() Notes:

Function	Support Level	Example Code
		1. The importtype argument of this function is unsupported. 2. When using this function, you should ensure that the data type imported matches the data type that you specified in a DataWindow control, DataStore object or graph control. Otherwise the output on the Web may be different from that in PowerBuilder. For example, if the date type specified in a DataWindow is string, when you import a file with date number (eg, 2006-08-01), on the Web the date data will be recognized as a string (2006), however in PowerBuilder, it will be recognized as a date data (2006-08-01).
ImportFile	Supported	gr_1.ImportFile("D:\EMPLOYEE.TXT", 2, 30, 3) Note: 1. The *filename* argument of ImportFile function must be a tab-separated file (TXT) or a comma-separated file (CSV). 2. The importtype argument of this function is unsupported. 3. When using this function, you should ensure that the data type imported matches the data type that you specified in a DataWindow control, DataStore object or graph control. Otherwise the output on the Web may be different from that in PowerBuilder. For example, if the date type specified in a DataWindow is string, when you import a file with date number (eg, 2006-08-01), on the Web the date data will be recognized as a string (2006), however in PowerBuilder, it will be recognized as a date data (2006-08-01).
ImportString	Supported	gr_1.ImportString(ls_Text, 2, 30, 3) Notes: 1. The importtype argument of this function is unsupported. 2. When using this function, you should ensure that the data type imported matches the data type that you specified in a DataWindow control, DataStore object or graph control. Otherwise the output on the Web may be different from that

Function	Support Level	Example Code
		in PowerBuilder. For example, if the date type specified in a DataWindow is string, when you import a file with date number (eg, 2006-08-01), on the Web the date data will be recognized as a string (2006), however in PowerBuilder, it will be recognized as a date data (2006-08-01).
InsertCategory	Supported	gr_1.InsertCategory("Macs", CategoryNbr)
InsertData	Supported	gr_1.InsertData(SeriesNbr, CategoryNbr + 1, 1250) Note: The data point will be inserted according to the sorting rules of the Category axis.
InsertSeries	Supported	gr_1.InsertSeries("Costs", SeriesNbr)
ModifyData	Supported	gr_1.ModifyData(SeriesNbr, & CategoryNbr, 1250)
Move	Supported	gr_1.Move(150, 200)
ObjectAtPointer	Supported	Object_type = gr_1.ObjectAtPointer(SeriesNbr, ItemNbr)
PointerX	Supported	li_value = gr_1.PointerX()
PointerY	Supported	li_value = gr_1.PointerY()
PostEvent	Supported	gr_1.PostEvent(Clicked!)
Print	Supported	gr_1.Print(1, "1-5", FALSE, TRUE)
Reset	Supported	gr_1.Reset(Series!)
ResetDataColors	Supported	gr_1.ResetDataColors(SeriesNbr, 10)
Resize	Supported	gr_1.Resize(100, 150)
SaveAs	Supported	gr_1.SaveAs()
SeriesCount	Supported	li_value = gr_1.SeriesCount()
SeriesName	Supported	ls_value = gr_1.SeriesName(5)
SetDataPieExplode	Supported	gr_1.SetDataPieExplode(series, datapoint, percentage)
SetDataStyle	Supported	gr_1.SetDataStyle("gr_depts" , SeriesNbr, 6, Background!, 0)
SetFocus	Supported	gr_1.SetFocus()
SetPosition	Supported	gr_1.SetPosition(ToTop!)
SetRedraw	Supported	gr_1.SetRedraw(FALSE)
SetSeriesStyle	Supported	gr_1.SetSeriesStyle("Costs", & SymbolPlus!) Note: When using this function to set the overlay style of Graph DataWindow to "Scatter", it returns true and displays the DataWindow with all data points connected by lines. This is different from that in PowerBuilder.
Show	Supported	gr_1.Show()

Function	Support Level	Example Code
TriggerEvent	Supported	gr_1.TriggerEvent(Clicked!)
TypeOf	Supported	gr_1.Typeof()
GetContextService	Unsupported	

6.2.1.8 grAxis object

6.2.1.8.1 Properties for grAxis object

Table 6.129:

Property	Support Level	Example Code
AutoScale	Supported	gr_1.Values.AutoScale = FALSE Note: When this property is set to False, the maximum value for the axis cannot be smaller than the minimum value.
DataType	Supported	gr_1.Values.DataType = AdtDate! Notes: 1. On the Web, if the datatype of axis is set to AdtDateTime!, a graph can only display the data for one day. 2. It is unsupported to dynamically modify the GraphType property, if the datatype of the axis will be changed after the modification.
DispAttr	Supported	
DisplayEveryNLabels	Supported	gr_1.Series.DisplayEveryNLabels = 10
DropLines	Supported	gr_1.Series.DropLines = Dash!
Frame	Supported	gr_1.Series.Frame = Dash!
Label	Supported	gr_1.Values.Label = 'Lawsuits per 1000'
LabelDispAttr	Supported	
MajorDivisions	Supported	gr_1.Values.MajorDivisions = 10
MajorGridLine	Supported	gr_1.Value.MajorGridLine = Dash!
MajorTic	Supported	gr_1.Values.MajorTic = Straddle!
MaximumValue	Supported	gr_1.Values.DataType = AdtDouble! gr_1.Values.MaximumValue = 500000.00
MaxValDateTime	Supported	gr_1.Values.DataType = AdtDate! gr_1.Values.MaxValDateTime = 12/31/1999
MinimumValue	Supported	gr_1.Values.DataType = AdtDouble! gr_1.Values.MinimumValue = 0.00

Property	Support Level	Example Code
MinorDivisions	Supported	gr_1.Values.MinorDivisions = 10
MinorGridLine	Supported	gr_1.Value.MinorGridLine = Dot!
MinorTic	Supported	gr_1.Values.MinorTic = Outside!
MinValDateTime	Supported	gr_1.Values.DataType = AdtDate! gr_1.Values.MinValDateTime = 01/31/1900
OriginLine	Supported	gr_1.Values.OriginLine = Dash!
PrimaryLine	Supported	gr_1.Values.PrimaryLine = Dash!
RoundTo	Supported	gr_1.Values.DataType = AdtDate! gr_1.Values.RoundToUnit = RndMonths! gr_1.Values.RoundTo = 6
RoundToUnit	Supported	gr_1.Values.DataType = AdtDate! gr_1.Values.RoundToUnit = RndMonths! gr_1.Values.RoundTo = 6
ScaleType	Supported	gr_1.Values.ScaleType=Log10! Notes: 1. Modifying the ScaleType property for an axis on the Web will affect only the involved axis. This may differ from PowerBuilder. 2. Graphs on the Web will be displayed exactly as the settings of MajorDivisions, MinorDivisions, MaximunValue, and RoundTo. There is no displaying difference between the linear and logarithmic scaling.
ScaleValue	Supported	gr_1.Values.ScaleValue = Actual!
SecondaryLine	Supported	gr_1.Values.SecondaryLine = Dash!
ShadeBackEdge	Supported	gr_1.ShadeColor = RGB(240,250,150) gr_1.Category.ShadeBackEdge = TRUE
ClassDefinition	Unsupported	

6.2.1.8.2 Functions for grAxis object

Table 6.130:

Function	Support Level	Example Code
ClassName	Supported	gr_1.value.ClassName()
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = gr_1.value.GetParent()
TypeOf	Supported	If gr_1.value.TypeOf() =grAxis! Then

Function	Support Level	Example Code
		ls_returnvalue = "grAxis!" End If
GetContextService	Unsupported	

6.2.1.9 grDispAttr object

6.2.1.9.1 Properties for grDispAttr object

Table 6.131:

Property	Support Level	Example Code
Alignment	Supported	gr_1.Alignment = Center!
AutoSize	Supported	gr_1.TitleDispAttr.AutoSize = FALSE
BackColor	Supported	gr_1.BackColor = RGB(255, 255, 0)
DisplayExpression	Supported	gr_1.TitleDispAttr.DisplayExpression = 'title + " " + Today()' Note: When specifying the display expression in the Display Expression field, please note that several items in the Columns box may be unsupported in certain situations: 1. (For all graphs) The percentofcategory and The percentofgraph are unsupported. 2. (For all graphs) The percentofseries is unsupported when the text object is set to Value Axis Text or Legend. 3. The categorypercentofgraph and percentofseries are unsupported when the text object is set to Pie Graph Labels and there are multiple series. 4. (For pie graph only) The percentofseries is unsupported when the text object is set to Legend.
Escapement	Supported	gr_1.Value.LabelDispAttr.Escapement = 900 Note: 1. When the value of Escapement is a negative number, the text will not be rotated. 2. The text rotated can be displayed out of the control in PowerBuilder, but not in Web applications. 3. After rotated, the multiple-line text still displays in multiple lines in PowerBuilder, but displays in one line on Web.

Property	Support Level	Example Code
FaceName	Supported	gr_1.FaceName = ""
FillPattern	Supported	gr_1.FillPattern = Diamond!
FontCharSet	Supported	gr_1.FontCharSet = ANSI!
FontFamily	Supported	gr_1.FontFamily = Roman!
FontPitch	Supported	gr_1.FontPitch = Fixed!
Format	Supported	gr__1.Values.DispAttr.Format = "0.00"
Italic	Supported	gr_1.Italic = TRUE
TextColor	Supported	gr_1.Series.DispAttr.TextColor = RGB(0,0,255)
TextSize	Supported	gr_1.Values.LabelDispAttr.TextSize = 12
Underline	Supported	gr_1.Values.LabelDispAttr.Underline = TRUE
Weight	Supported	gr_1.Weight = 700
ClassDefinition	Unsupported	

6.2.1.9.2 Functions for grDispAttr object

Table 6.132:

Function	Support Level	Example Code
ClassName	Supported	gr_1.ClassName()
GetParent	Supported	luo_value = gr_1.GetParent()

TypeOf	Supported	If gr_1.TypeOf() = grDispAttr! Then
		ls_returnvalue = "grDispAttr!"
		End If

GetContextService	Unsupported	

6.2.1.10 Inet object

6.2.1.10.1 Properties for Inet object

Table 6.133:

Property	Support Level	Example Code
ClassDefinition	Unsupported	

6.2.1.10.2 Events for Inet object

Table 6.134:

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor

6.2.1.10.3 Functions for Inet object

Table 6.135:

Function	Support Level	Example Code
ClassName	Supported	inet_1 = CREATE inet li_classnm = inet_1.ClassName( )
GetParent	Supported	lpo_return = inet_1.GetParent()
GetURL	Supported	inet_1.GetURL( )
HyperLinkToURL	Supported	inet_1.HyperlinkToURL("http://www.appeon.com")
PostEvent	Supported	inet_1.PostEvent(Clicked!)
PostURL	Supported	li_return =Inet_1.PostURL() Note: Appeon supports https address, though PB does not support it.
TriggerEvent	Supported	inet_1.TriggerEvent(Clicked!)
TypeOf	Supported	inet_1.Typeof()
GetContextService	Supported	GetContextService("Internet", inet_base) inet_base.HyperlinkToURL("http://www.appeon.com") Note: This function is supported only when the service name parameter is "Internet".

6.2.1.11 InternetResult object

6.2.1.11.1 Properties for InternetResult object

Table 6.136:

Property	Support Level	Example Code
ClassDefinition	Unsupported	

6.2.1.11.2 Events for InternetResult object

Table 6.137:

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor

6.2.1.11.3 Functions for InternetResult object

Table 6.138:

Function	Support Level	Example Code
ClassName	Supported	internetresult_1 = Create InternetResult

Function	Support Level	Example Code
		li_classnm= internetresult_1.ClassName()
GetParent	Supported	lpo_return = internetresult_1.GetParent()
InternetData	Supported	li_return = internetresult_1.InternetData(data)
InternetStatus	Supported	(Not used)
PostEvent	Supported	Internetresult_1.PostEvent(Clicked!)
TriggerEvent	Supported	Internetresult_1.TriggerEvent(Clicked!)
TypeOf	Supported	Internetresult_1.Typeof()
GetContextService	Unsupported	

6.2.1.12 ListViewItem object

6.2.1.12.1 Properties for ListViewItem object

Table 6.139:

Property	Support Level	Example Code
CutHighlighted	Supported	lvi_item1.CutHighlighted = True
Data	Supported	any_value = lvi_item1.Data
DropHighlighted	Supported	lvi_item1.DropHighlighted = True
HasFocus	Supported	lvi_item1.HasFocus = True
ItemX	Supported	
ItemY	Supported	
Label	Supported	ls_value = lvi_item1.Label InPowerBuilder, if the string label is too long to display in one line, the string will display in multiple lines. However, in the Web application, the string will only display in one line.
OverlayPicture Index	Supported	li_value = lvi_item1.OverlayPictureIndex
PictureIndex	Supported	li_value = lvi_item1.PictureIndex
Selected	Supported	lb_value = lvi_item1.Selected
StatePictureIndex	Supported	li_value = lvi_item1.StatePictureIndex
ClassDefinition	Unsupported	

6.2.1.12.2 Functions for ListViewItem object

Table 6.140:

Function	Support Level	Example Code
ClassName	Supported	ls_objectname = lvi_item1.ClassName()

Function	Support Level	Example Code

TypeOf	Supported	IF lvi_item1.TypeOf() = ListViewItem! THEN
		ls_returnvalue = "ListViewItem"
		END IF

GetContextService	Unsupported	
GetParent	Unsupported	

6.2.1.13 mailFileDescription object

6.2.1.13.1 Properties for mailFileDescription object

Table 6.141:

Property	Support Level	Example Code
FileType	Supported	mailFileDescription mAttach mAttach.FileType = mailAttach!
Filename	Supported	mAttach.Filename = ls_filename
Pathname	Supported	mAttach.Pathname = ls_pathname
Position	Supported	mAttach.Position = ll_position
ClassDefinition	Unsupported	

6.2.1.13.2 Functions for mailFileDescription object

Table 6.142:

Function	Support Level	Example Code
ClassName	Supported	ls_classnm = mAttach.classname()
TypeOf	Supported	type_obj = mAttach.typeof()
GetContextService	Unsupported	
GetParent	Unsupported	

6.2.1.14 mailMessage object

6.2.1.14.1 Properties for mailMessage object

Table 6.143:

Property	Support Level	Example Code
AttachmentFile[ ]	Supported	mMsg.AttachmentFile[1] = mAttach
ConversationID	Supported	ls_ID = mMsg.ConversationID
DateReceived	Supported	mMsg.DateReceived = ls_today
MessageSent	Supported	lb_sent = mMsg.MessageSent

Property	Support Level	Example Code
MessageType	Supported	ls_type = mMsg.MessageType
NoteText	Supported	ls_note = mMsg.NoteText
ReceiptRequested	Supported	lb_return = mMsg.ReceiptRequested
Recipient[ ]	Supported	mailRecipient recip mMsg.Recipient[1] = recip
Subject	Supported	mMsg.subject = ls_subject
Unread	Supported	mMsg.Unread = false
ClassDefinition	Unsupported	

6.2.1.14.2 Functions for mailMessage object

Table 6.144:

Function	Support Level	Example Code
ClassName	Supported	ls_classnm = mMsg.classname()
TypeOf	Supported	type_obj = mMsg.typeof()
GetContextService	Unsupported	
GetParent	Unsupported	

6.2.1.15 mailRecipient object

6.2.1.15.1 Properties for mailRecipient object

Table 6.145:

Property	Support Level	Example Code
Address	Supported	mRecip.Address = ls_address
EntryID	Supported	ls_return = string(mRecip.EntryID)
Name	Supported	mRecip.Name = ls_name
RecipientType	Supported	mRecip.RecipientType = mailTo!
ClassDefinition	Unsupported	

6.2.1.15.2 Functions for mailRecipient object

Table 6.146:

Function	Support Level	Example Code
ClassName	Supported	ls_classnm = mRecip.classname()
TypeOf	Supported	type_obj = mRecip.typeof()
GetContextService	Unsupported	
GetParent	Unsupported	

6.2.1.16 mailSession object

6.2.1.16.1 Properties for mailSession object

Table 6.147:

Property	Support Level	Example Code
MessageID[]	Supported	ls_message = mSes.MessageID[li_i]
SessionID	Supported	ll_sessionid = mSes.SessionID
ClassDefinition	Unsupported	

6.2.1.16.2 Events for mailSession object

Table 6.148:

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor

6.2.1.16.3 Functions for mailSession object

Table 6.149:

Function	Support Level	Example Code
ClassName	Supported	ls_classnm = mSes.classname()
GetParent	Supported	lobj_parent = mSes.GetParent
mailAddress	Supported	mRet = mSes.mailAddress()
mailDelete Message	Supported	mRet = mSes.mailDeleteMessage(sID)
mailGetMessages	Supported	mRet = mSes.mailGetMessages()
mailHandle	Supported	mHandle = mSes.mailHandle()
mailLogoff	Supported	mRet = mSes.mailLogoff()
mailLogon	Supported	mRet = mSes.mailLogon ()
mailReadMessage	Supported	mRet = mSes.mailReadMessage(sMessageID, mMsg , mailEntireMessage!, TRUE)
mailRecipient Details	Supported	mRet = mSes.mailRecipientDetails(mMsg.Recipient[1],TRUE)
mailResolve Recipient	Supported	mRet = mSes.mailResolveRecipient(mRecip,true)
mailSaveMessage	Supported	mRet = mSes.mailSaveMessage(mMsg.ConversationID, mMsg)
mailSend	Supported	mRet = mSes.mailsend(mMsg)
PostEvent	Supported	lb_return = mSes.PostEvent("ue_user")
TriggerEvent	Supported	lb_return = mSes.TriggerEvent("ue_user")

Function	Support Level	Example Code
TypeOf	Supported	type_obj = mSes.typeof()
GetContextService	Unsupported	

6.2.1.17 MDIClient object

Important Requirements

Multiple MDI windows are supported.

Defining the size of the client area in an MDI window is supported.

If there are multiple MDI frames, the second frame window of the Web application will be loaded at a much slower speed up to four times slower than the first one. If the first MDI is closed, the second one will not function.

When the MDI frame is closed in an Appeon-deployed application, the Web browser will also close automatically.

6.2.1.17.1 Properties for MDIClient object

Table 6.150:

Property	Support Level	Example Code
BackColor	Supported	w_main.mdi_1.BackColor = ll_value
Height	Supported	w_main.mdi_1.Height = li_value
MicroHelpHeight	Supported	li_return = w_main.mdi_1.MicroHelpHeight
Tag	Supported	w_main.mdi_1.Tag = ls_value
Width	Supported	w_main.mdi_1.Width = li_value
X	Supported	w_main.mdi_1.X = li_value
Y	Supported	w_main.mdi_1.Y = li_value
BringToTop	Unsupported	
ClassDefinition	Unsupported	
Visible	Supported	

6.2.1.17.2 Functions for MDIClient object

Table 6.151:

Function	Support Level	Example Code
ClassName	Supported	ls_classname = w_main.mdi_1.Classname()
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = this.GetParent()
Hide	Supported	w_main.Hide()
Move	Supported	w_main.Move(150,200)
Resize	Supported	w_main.Resize(100, 150)

Function	Support Level	Example Code
SetRedraw	Supported	w_main.SetRedraw(FALSE)
Show	Supported	m_main.Show()
TypeOf	Supported	ls_type = w_main.mdi_1.TypeOf()
GetContextService	Unsupported	

6.2.1.18 Menu object

Important Requirements

- Menus can be dynamically created (for example, using the CREATE statement).
- When a menu is dynamically created, it is unsupported to create multiple items to the menu at one time.
- Use of Main menu should be limited to no more than five layers of submenus for a single menu item.
- More levels of dropdown menus under a menu bar item are supported.
- Appeon supports user-defined menu events and functions.
- For more information on menu object, refer to: Menu Features and Toolbar Features.

6.2.1.18.1 Menu features

Supported

- Menu object may be implemented with the following functionalities:

  1. Modifying menu items based on user rights using the enable and visible properties.
  2. User-defined functions in menu object.

- It is supported to have a menu object as a structure variable, or define a structure/declare a structure instance in a menu object.
- It is supported to declare a menu type array.
- It is supported to qualify the item of a child menu with the class name of the parent menu. For example:

  m_city is the menu for the window w_citysheet, and is inherited from m_main. m_file is an item of the menu m_city.

  The following two lines of code have the same effect on the Web:

```
m_main.m_file.enabled = FALSE
m_city.m_file.enabled = FALSE
```

- It is supported to declare a menu variable, assign value to it, and pass the variable as argument. For example:

```
menu lm_onemenu
lm_ onemenu = create menu
lm_ onemenu.text = "Open a window" //or
lm_ onemenu = m_main.m_file //or
lm_ onemenu = m_main.Item[1].Item[1]
```

- A window control can use the ChangeMenu function to dynamically change the menu. See the ChangeMenu function for Window control.
- Switching between different menus is allowed. However, only one menu can be displayed at one time. For example, the MDI window menu can be dynamically refreshed when a sheet is opened or closed.
- Append the names of open sheets (window list) to a menu item.
- Using the UpArrow/DownArrow/LeftArrow/RightArrow to change the currently selected menu item is supported.
- Shortcut keys for menu items are supported.

Unsupported

- In PowerBuilder, the menu items always show in front of main window or popup window. In Web applications, the menu items may get hidden behind a main window or a popup window.
- Different from PowerBuilder as follows: In the case of multiple Web windows loading the same menu, when a window (window A) accesses the menu object, the menu object accessed is not the menu in the window A, but the menu in the window that is most recently opened.
- It is not supported to access the menu of the window that currently does not have focus.

6.2.1.18.2 Toolbar features

Supported

- The toolbar item synchronizes with the relevant menu item. If the property of the menu item changes, the property of the toolbar item changes accordingly.
- Multiple toolbars are supported.
- Multiple toolbars can be displayed at the same time, but docked toolbar or toolbar toggle are not supported.
- Toolbar can only be displayed in the MDI and MDIHelp windows.

Unsupported

- In Web applications, the toolbars are always placed directly below the menu. The Web toolbar cannot be dragged around or placed at any other place.

- In PowerBuilder, if the user right mouse clicks over a toolbar, a popup menu is displayed with the available options, for example: Left, Right, Floating, Show Text, Show PowerTips. The popup menu will not display in Web applications. The toolbar item cannot show text but can show PowerTips.
- Clicking the toolbar item will not trigger the Selected event for Menu object.
- If the ShiftToRight property is set to TRUE, menu objects may shift to right and toolbar items will automatically adjust its places according to the menu changes in PowerBuilder. However, toolbar items will not automatically adjust its places on the Web.

6.2.1.18.3 Properties for Menu object

Table 6.152:

Property	Support Level	Example Code
Checked	Supported	m_function.m_testitemforfunction.Checked = lb_value
Default	Supported	m_function.m_testitemforfunction.Default = TRUE
Enabled	Supported	m_function.m_testitemforfunction.Enabled = lb_value
Item [ ]	Supported	Menu lmenu_item[] lmenu_item =m_menu.Item[] Support using Item[] in different ways: lm_ onemenu = m_main.Item[1].Item[1] m_main.Item[1].Item[1].text = "open a Window" m_main.Item[1].Item[1].triggerevent(clicked) Dynamically adding menu items with Item[] is supported.
MicroHelp	Supported	m_function.m_testitemforfunction.MicroHelp = ls_value
ParentWindow	Supported	w_value = m_function.Parentwindow
ShiftToRight	Supported	m_function.m_testitemforfunction.ShiftToRight = true
Shortcut	Supported	
Tag	Supported	m_function.m_testitemforfunction.Tag = ls_value
Text	Supported	m_function.m_testitemforfunction.Text = ls_value
ToolbarItemBarIndex	Supported	m_function.m_testitemforfunction.ToolbarItemBarIndex = li_value Note: If the property ToolbarItemBarIndex is set to 0, the toolbar does not display in PowerBuilder but displays on the Web.
ToolbarItemDown	Supported	m_function.m_testitemforfunction.ToolbarItemDown = true
ToolbarItemDownName	Supported	m_function.m_testitemforfunction.ToolbarItemDownName = ls_value

Property	Support Level	Example Code
ToolbarItemName	Supported	m_function.m_testitemforfunction.ToolbarItemName = ls_value
ToolbarItemOrder	Supported	m_function.m_testitemforfunction.ToolbarItemOrder = li_value
ToolbarItemSpace	Supported	m_function.m_testitemforfunction.ToolBarItemSpace = 5
ToolbarItemText	Supported	m_function.m_testitemforfunction.ToolbarItemText = ls_value
ToolbarItemVisible	Supported	m_function.m_testitemforfunction.ToolbarItemVisible = lb_value
Visible	Supported	m_function.m_testitemforfunction.Visible = lb_value
ClassDefinition	Unsupported	
MenuItemType	Unsupported	
MergeOption	Unsupported	

6.2.1.18.4 Events for Menu object

Table 6.153:

Event	Support Level	Example Code
Clicked	Supported	Clicked Note: The Clicked event cannot be triggered for a menu item that has a submenu.
Selected	Supported	Selected
Help	Unsupported	

6.2.1.18.5 Functions for Menu object

Table 6.154:

Function	Support Level	Example Code
Check	Supported	ll_returnvalue = m_function.m_testitemforfunction.Check()
ClassName	Supported	ls_returnvalue = m_function.m_testitemforfunction.ClassName()
Disable	Supported	ll_returnvalue = m_function.m_testitemforfunction.Disable()
Enable	Supported	ll_returnvalue = m_function.m_testitemforfunction.Enable()
GetParent	Supported	lobj_returnvalue = m_function.GetParent()
Hide	Supported	ll_returnvalue = m_function.m_testitemforfunction.Hide()

Function	Support Level	Example Code
PopMenu	Supported	ll_returnvalue = m_function.m_testitemforfunction.PopMenu(xlocation, ylocation)
PostEvent	Supported	ll_returnvalue = m_function.m_testitemforfunction.PostEvent(event)
Show	Supported	ll_returnvalue = m_function.m_testitemforfunction.Show()
TriggerEvent	Supported	ll_returnvalue = m_function.m_testitemforfunction.TriggerEvent(event)

TypeOf	Supported	If m_function.m_testitemforfunction.TypeOf() = menu! Then
		ls_returnvalue = "menu"
		end if

Uncheck	Supported	ll_returnvalue = m_function.m_testitemforfunction.Uncheck()
GetContextService	Unsupported	

6.2.1.19 MenuCascade object

Important requirements

- It is unsupported to place a cascading menu in another cascading menu.
- When you extend a cascading menu, the first dropdown item automatically get focus on the Web. Pressing the blank key can trigger the Click event of the item with focus and the focus can be shifted between items if you pressing the Up and Down arrow keys.
- Changes to a Web menu will be updated automatically without calling the Hide and Show functions for the Menu.

6.2.1.19.1 Properties for MenuCascade object

Table 6.155:

Property	Support Level	Example Code
Checked	Supported	m_function.m_testitemforfunction.Checked = lb_value
Column	Supported	m_menu.Columns = li_count
CurrentItem	Supported	
Default	Supported	m_menu.Default = TRUE
DropDown	Supported	m_menu.DropDown = False
Enabled	Supported	m_function.m_testitemforfunction.Enabled = lb_value
Item [ ]	Supported	lmenu_item = m_menu.Item[ ]

Property	Support Level	Example Code
MicroHelp	Supported	m_function.m_testitemforfunction.MicroHelp = ls_value
ParentWindow	Supported	w_value = m_function.Parentwindow
ShiftToRignt	Supported	m_menu.ShiftToRight = True
Shortcut	Supported	li_return = m_menu.Shortcut
Tag	Supported	m_function.m_testitemforfunction.Tag = ls_value
Text	Supported	m_function.m_testitemforfunction.Text = ls_value
ToolbarItemDown	Supported	m_function.m_testitemforfunction.ToolbarItemDown = lb_flag
ToolbarItemDownName	Supported	m_function.m_testitemforfunction.ToolbarItemDownName = ls_value
ToolbarItemBarIndex	Supported	m_function.m_testitemforfunction.ToolbarItemBarIndex = li_value Note: If the property ToolbarItemBarIndex is set to 0, the toolbar does not display in PowerBuilder but displays on the Web.
ToolbarItemName	Supported	m_function.m_testitemforfunction.ToolbarItemName = ls_value
ToolbarItemOrder	Supported	m_function.m_testitemforfunction.ToolbarItemOrder = li_value Note: the value of ToolbarItemOrder cannot be a negative number.
ToolbarItemText	Supported	m_function.m_testitemforfunction.ToolbarItemText = ls_value
ToolbarItemVisible	Supported	m_function.m_testitemforfunction.ToolbarItemVisible = lb_value
Visible	Supported	m_function.m_testitemforfunction.Visible = lb_value
ClassDefinition	Unsupported	
MenuItemType	Unsupported	
MergeOption	Unsupported	

6.2.1.19.2 Events for MenuCascade object

Table 6.156:

Event	Support Level	Example Code
Clicked	Supported	Clicked Note: The Clicked event cannot be triggered for a menu item that has a submenu.
Selected	Supported	Selected
Help	Unsupported	

6.2.1.19.3 Functions for MenuCascade object

Table 6.157:

Function	Support Level	Example Code
Check	Supported	ll_returnvalue = m_function.m_testitemforfunction.Check()
ClassName	Supported	ls_returnvalue = m_function.m_testitemforfunction.ClassName()
Disable	Supported	ll_returnvalue = m_function.m_testitemforfunction.Disable()
Enable	Supported	ll_returnvalue = m_function.m_testitemforfunction.Enable()
GetParent	Supported	lobj_returnvalue = m_function.GetParent()
Hide	Supported	ll_returnvalue = m_function.m_testitemforfunction.Hide()
PopMenu	Supported	ll_returnvalue = m_function.m_testitemforfunction.PopMenu(xlocation, ylocation)
PostEvent	Supported	ll_returnvalue = m_function.m_testitemforfunction.PostEvent(event)
Show	Supported	ll_returnvalue = m_function.m_testitemforfunction.Show()
TriggerEvent	Supported	ll_returnvalue = m_function.m_testitemforfunction.TriggerEvent(event)

TypeOf	Supported	If m_function.m_testitemforfunction.TypeOf() = menu! Then
		ls_returnvalue = "menu"
		end if

Uncheck	Supported	ll_returnvalue = m_function.m_testitemforfunction.Uncheck()
GetContextService	Unsupported	

6.2.1.20 Message object

Important Requirements

- Both the system message and the user-defined message are supported.

6.2.1.20.1 Properties for Message object

Table 6.158:

Property	Support Level	Example Code
DoubleParm	Supported	OpenWithParm(w_1, ld_value)

Property	Support Level	Example Code
		ld_value = Message.DoubleParm
LongParm	Supported	ls_value = Message.LongParm
PowerObjectParm	Supported	OpenWithParm(w_1, s_value) ls_value = Message.PowerObjectParm
Processed	Supported	
StringParm	Supported	OpenWithParm(w_1, ls_value) ls_value = Message.StringParm
WordParm	Supported	
ClassDefinition	Unsupported	
Handle	Unsupported	
Number	Unsupported	
ReturnValue	Unsupported	

6.2.1.20.2 Events for Message object

Table 6.159:

Event	Support Level	Example Code
Constructor	Supported	
Destructor	Supported	

6.2.1.20.3 Functions for Message object

Table 6.160:

Function	Support Level	Example Code
ClassName	Supported	string varname varname = ClassName(gd_double)
GetParent	Supported	lobj_parent = cb_1.GetParent()
PostEvent	Supported	cb_1.PostEvent(Clicked!)
TriggerEvent	Supported	cb_1.TriggerEvent(Clicked!)
TypeOf	Supported	dw_1.Typeof()
GetContextService	Unsupported	

6.2.1.21 NonVisualObject object

6.2.1.21.1 Properties for NonVisualObject object

Table 6.161:

Property	Support Level	Example Code
ClassDefinition	Unsupported	

6.2.1.21.2 Events for NonVisualObject object

Table 6.162:

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor

6.2.1.21.3 Functions for NonVisualObject object

Table 6.163:

Function	Support Level	Example Code
ClassName	Supported	nvo_test.classname()
GetParent	Supported	lobj_parent = nvo_test.GetParent()
PostEvent	Supported	nvo_test.PostEvent(Constructor!)
TriggerEvent	Supported	nvo_test.TriggerEvent(Constructor!)

TypeOf	Supported	If nvo_test.typeof()=nonvisualobject! Then
		ls_value='nonvisualobject!'
		End If

GetContextService	Unsupported	

6.2.1.22 OLEObject object

Unsupported

The shortcut key for OCX is unsupported.

It is unsupported to have blob, DataTime, Date, or Time parameters in functions of OLEObject object.

The datatype of the property value cannot be Datetime, Date, or Time.

6.2.1.22.1 Properties for OLEObject object

Table 6.164:

Property	Support Level	Example Code
ClassDefinition	Unsupported	
Handle	Unsupported	

6.2.1.22.2 Events for OLEObject object

Table 6.165:

Event	Support Level	Example Code
Constructor	Supported	Constructor

Event	Support Level	Example Code
Destructor	Supported	Destructor
Error	Unsupported	
ExternalException	Unsupported	

6.2.1.22.3 Functions for OLEObject object

Table 6.166:

Function	Support Level	Example Code
ClassName	Supported	ls_return = ole_1.ClassName()
ConnectToNew Object	Supported	OLEObject myoleobject myoleobject = CREATE OLEObject li_return = myoleobject.ConnectToNewObject("excel.application")
ConnectToObject	Supported	OLEObject myoleobject myoleobject = CREATE OLEObject li_return = myoleobject.ConnectToObject("c:\excel \expense.xls")
DisconnectObject	Supported	li_return = myoleobject.DisconnectObject()
GetAutomation NativePointer	Supported	UnsignedLong lul_oleptr li_return = ocx_spell.GetNativePointer(lul_oleptr)
IsAlive	Supported	IF ole_1.IsAlive() THEN return
PostEvent	Supported	ole_1.PostEvent("Error")
Release AutomationNative Pointer	Supported	li_return = ocx_spell.ReleaseNativePointer(lul_oleptr)
SetAutomation Pointer	Supported	li_return = oleChild.SetAutomationPointer(myoleobject)
TriggerEvent	Supported	li_return = ole_1.TriggerEvent("Error")
TypeOf	Supported	
ConnectToNew RemoteObject	Unsupported	
ConnectTo RemoteObject	Unsupported	
GetContextService	Unsupported	
GetParent	Supported	
SetAutomation Locale	Unsupported	
SetAutomation Timeout	Unsupported	

6.2.1.23 Timing object

6.2.1.23.1 Properties for Timing object

Table 6.167:

Property	Support Level	Example Code
Interval	Supported	ld_return = timing_1.interval
Running	Supported	lb_return = timing_1.running
ClassDefinition	Unsupported	

6.2.1.23.2 Events for Timing object

Table 6.168:

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor
Timer	Supported	Timer(10, w_main)

6.2.1.23.3 Functions for Timing object

Table 6.169:

Function	Support Level	Example Code
ClassName	Supported	timing_1 = CREATE uo_timer ls_classnm = timing_1.ClassName( )
GetParent	Supported	lobj_parent_name = timing_1.GetParent()
PostEvent	Supported	timing_1.PostEvent(Clicked!)
Start	Supported	timing_1.Start(60)
Stop	Supported	timing_1.Stop()
TriggerEvent	Supported	timing_1.TriggerEvent(Clicked!)
TypeOf	Supported	timing_1.Typeof()
GetContextService	Unsupported	

6.2.1.24 Transaction object

Important Requirements

The Transaction object can either be global or local. The instance transaction object is unsupported.

The user-defined Transaction object is supported.

The type of SQLCA can be transaction or transaction user object. It is supported if the user defines a transaction user object (for example, u_trans) and sets the type of SQLCA to the transaction user object (for example, u_trans).

An application can have multiple Transactions Objects, and the Transaction Objects can be connected to different type of databases.

The specification of isolation level for a database profile in PowerBuilder is not supported.

6.2.1.24.1 Properties for Transaction object

Appeon ignores the dynamic property settings for the data source and Transaction Object except for the AutoCommit property.

Table 6.170:

Property	Support Level	Example Code
AutoCommit	Supported	sqlca.AutoCommit = lb_value Notes: 1. If the database server is Oracle and the driver is native driver, the AutoCommit setting is always interpreted as FALSE. 2. The AutoCommit property is the only property that may be dynamically modified.
Database	Supported	ls_value = sqlca.Database
DBMS	Supported	ls_value = sqlca.DBMS
DBPass	Supported	
DBParm	Supported	i_sqlca.DBParm= "ConnectionString= 'DSN=AppeonSample;UID=dba;PWD=sql'" //DBMS-specific parameters
LogID	Supported	ls_value = sqlca.LogID
LogPass	Supported	ls_value = sqlca.LogPass
ServerName	Supported	ls_value = sqlca.ServerName
SQLCode	Supported	ll_value = sqlca.SQLCode The returned value of SQLCode in Appeon: <0 – Error; the statement failed. This is different from PowerBuilder where -1 will be returned, while in Appeon, it could be any negative number when there is an error. Therefore, it is recommended to use <0 as the evaluation value for errors in both PowerBuilder and Appeon. 0 – Success. 100 – No result.
SQLDBCode	Supported	ll_value = sqlca.SQLDBCode Notes: 1. In PowerBuilder, if sqlca.SQLCode=100, sqlca.SQLDBCode = 3. In Appeon, when sqlca.SQLCode=100, sqlca.SQLDBCode = 0. 2. In the Web application, SQLDBCode returns a different value from PowerBuilder. It is

Property	Support Level	Example Code
		recommended not to use SQLDBCode in statements such as IF…ELSE.
SQLErrText	Supported	ls_value = sqlca.SQLErrText
SQLNRows	Supported	ll_value = sqlca.SqlNRows
UserID	Supported	ls_value = sqlca.UserID
ClassDefinition	Unsupported	
Lock	Unsupported	
SQLReturnData	Unsupported	

6.2.1.24.2 Events for Transaction object

Table 6.171:

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor
DBError	Unsupported	
SQLPreview	Unsupported	

6.2.1.24.3 Functions for Transaction object

Table 6.172:

Function	Support Level	Example Code
ClassName	Supported	ls_returnvalue = sqlca.ClassName()

TypeOf	Supported	if this.TypeOf() = transaction! Then
		Is_returnvalue = "transaction!"
		end if

DBHandle	Supported	
PostEvent	Supported	
SyntaxFromSQL	Supported	ls_dw_syntax = SQLCA.SyntaxFromSQL(ls_sql_syntax, ls_style, ls_dw_err) Notes: 1. The following syntax is unsupported: ls_dw_syntax = SyntaxFromSQL(sqlca, ls_sql_syntax, ls_style, ls_dw_err) 2. The data type of computed columns in SyntaxFromSQL cannot be the Appeon unsupported data type.

Function	Support Level	Example Code
		3. The length of return value on the Web is different from PowerBuilder 4. The return value of the SyntaxFromSQL function on the Web contains a column name for each column although it is not in PowerBuilder. 5. The function cannot generate correct source code for TreeView DataWindow.
TriggerEvent	Supported	
GetContextService	Unsupported	
GetParent	Supported	

6.2.1.25 TreeViewItem object

6.2.1.25.1 Properties for TreeViewItem object

Table 6.173:

Property	Support Level	Example Code
Bold	Supported	tv_treeviewitem.Bold = TRUE
Children	Supported	tv_treeviewitem.Children = FALSE
Data	Supported	tv_treeviewitem.Data = sle_prop.text
Expanded	Supported	tv_treeviewitem.Expanded = TRUE Note: the property is supported but is read-only.
ExpandedOnce	Supported	tv_treeviewitem.ExpandedOnce = TRUE Note: the property is supported but is read-only.
HasFocus	Supported	tv_treeviewitem.HasFocus = TRUE Note: the property is supported but is read-only.
ItemHandle	Supported	tv_treeviewitem.Itemhandle = long(sle_prop.text) Note: This property is supported but is read-only.
OverlayPicture Index	Supported	l_tvi.OverlayPictureIndex = 2
Label	Supported	tv_treeviewitem.Label = sle_prop.text
Level	Supported	tv_treeviewitem.Level = long(sle_prop.text)
PictureIndex	Supported	tv_treeviewitem.PictureIndex = long(sle_prop.text)
SelectedPictureIndex	Supported	tv_treeviewitem.SelectedPictureIndex = long(sle_prop.text)
Selected	Supported	lb_value = tv_treeviewitem.Selected
StatePicture Index	Supported	li_value = tv_treeviewitem.StatePictureIndex

Property	Support Level	Example Code
ClassDefinition	Unsupported	
CutHighLighted	Unsupported	
DropHighLighted	Unsupported	

6.2.1.25.2 Functions for TreeViewItem object

Table 6.174:

Function	Support Level	Example Code
ClassName	Supported	ls_returnvalue = tv_treeviewitem.ClassName()

TypeOf	Supported	If ltvi_1.TypeOf() = TreeViewItem! Then
		ls_returnvalue = "TreeViewItem!"
		End If

GetContextService	Unsupported	
GetParent	Unsupported	

6.2.1.26 UserObject object

Important Requirements

The UserObject object can be dynamically created (for example, by using the CREATE statement).

6.2.1.26.1 Properties for UserObject object

Table 6.175:

Property	Support Level	Example Code
BackColor	Supported	uo_test.BackColor = ll_value
Border	Supported	uo_test.Border = lb_value
BorderStyle	Supported	uo_test.BorderStyle = StyleBox! uo_test.BorderStyle = StyleLowered! uo_test.BorderStyle = StyleRaised! Note: StyleShadowBox! is unsupported.
ClassName	Supported	ls_value = uo_test.ClassName
BringToTop	Supported	uo_test.BringToTop = TRUE
Control[]	Supported	WindowObject lobj_control[] lobj_control[] = uo_test.control[]
DragAuto	Supported	uo_test.DragAuto=TRUE
DragIcon	Supported	uo_test.DragIcon='c:\archive\arrow.ico'
Enabled	Supported	uo_test.Enabled = TRUE
Height	Supported	uo_test.Height = 300
HScrollBar	Supported	uo_1.HScrollBar = false

Property	Support Level	Example Code
Pointer	Supported	uo_test.Pointer='Help!' uo_test.Pointer='d:\archive\IBEAM.BMP.cur'
PictureName	Supported	uo_test.PictureName = "c:\pictures\pb1.bmp"
PowerTipText	Supported	uo_test.PowerTipText="UserObject PowerTip"
TabBackColor	Supported	uo_tab.tab_page1.TabBackColor = RGB(0,0,255)
TabOrder	Supported	uo_test.TabOrder = li_value
TabTextColor	Supported	uo_tab.tab_page1.TabTextColor =5566
Tag	Supported	uo_test.Tag = ls_value
Text	Supported	uo_test.Text = ls_value
Visible	Supported	uo_test.Visible = lb_value
VScrollBar	Supported	uo_test.VScrollBar = lb_value
Width	Supported	uo_test.Width = li_value
X	Supported	uo_test.X = li_value
Y	Supported	uo_test.Y = li_value
ClassDefinition	Unsupported	
ColumnsPerPage	Unsupported	
LibraryName	Unsupported	
LinesPerPage	Unsupported	
ObjectType	Unsupported	
PictureMaskColor	Unsupported	
Style	Unsupported	
UnitsPerColumn	Unsupported	
UnitsPerLine	Unsupported	

6.2.1.26.2 Events for UserObject object

Table 6.176:

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor
DragDrop	Supported	DragDrop
DragEnter	Supported	DragEnter
DragLeave	Supported	DragLeave
DragWithin	Unsupported	DragWithin
RButtonDown	Supported	RButtonDown
Help	Unsupported	
Other	Unsupported	

6.2.1.26.3 Functions for UserObject object

Table 6.177:

Function	Support Level	Example Code
ClassName	Supported	uo_test.classname()
Drag	Supported	uo_test.Drag(Begin!)
GetParent	Supported	lobj_parent_name = this.GetParent()
Hide	Supported	uo_test.Hide()
Move	Supported	uo_test.Move(x1,y1)
PointerX	Supported	uo_test.PointerX()
PointerY	Supported	uo_test.PointerY()
PostEvent	Supported	uo_test.PostEvent(Clicked!)
Resize	Supported	uo_test.Resize(x1,y1)
SetFocus	Supported	uo_test.SetFocus()
SetPosition	Supported	uo_test.SetPosition(ToTop!) uo_two.SetPosition(Behind!, uo_three)
SetRedraw	Supported	li_return= uo_test.SetRedraw(TRUE)
Show	Supported	uo_test.Show()
TriggerEvent	Supported	uo_test.TriggerEvent(Clicked!)

TypeOf	Supported	If uo_test.typeof()=CheckBox! Then
		ls_value='CheckBox!'
		End If

AddItem	Unsupported	
CreatePage	Unsupported	
DeleteItem	Unsupported	
GetContextService	Unsupported	
InsertItem	Unsupported	
PageCreated	Unsupported	
Print	Unsupported	

6.2.1.27 WSConnection object

6.2.1.27.1 Properties for WSConnection object

Table 6.178:

Property	Support Level	Example Code
Endpoint	Supported	
Authentication Mode	Unsupported	

Property	Support Level	Example Code
ClassDefinition	Unsupported	
ClientCertificate File	Unsupported	
Password	Unsupported	
ProxyServerHost Name	Unsupported	
ProxyServer Password	Unsupported	
ProxyServerPort	Unsupported	
ProxyServer UserName	Unsupported	
Timeout	Unsupported	
UserDomain	Unsupported	
UserName	Unsupported	
UseWindows Integrated Authentication	Unsupported	

6.2.1.27.2 Events for WSConnection object

Table 6.179:

Event	Support Level	Example Code
Constructor	Supported	Note: Activate and Deactivate events are unsupported for dialogue boxes.
Destructor	Supported	Clicked

6.2.1.27.3 Functions for WSConnection object

Table 6.180:

Function	Support Level	Example Code
ClassName	Supported	w_mdi.ArrangeSheets(arrangetype)
GetParent	Supported	Parent.ChangeMenu(m_test_menu2)
PostEvent	Supported	ls_value = w_test.ClassName()
TriggerEvent	Supported	w_test.CloseChannel()
TypeOf	Supported	w_test.CloseUserObject()
GetContextService	Unsupported	

6.2.2 Unsupported Objects

The following are the system objects that Appeon does not support:

Table 6.181:

ADOResultSet	ArrayBounds	ClassDefinition	ClassDefinitionObject
ConnectionInfo	ConnectObject	ContextInformation	ContextKeyword
CORBABadType Context	CORBABadInvorder	CORBABadOperation	CORBABadParam
CORBABadTypeCode	CORBACommFailure	CORBACurrent	CORBADataConversion
CORBAFreeMem	CORBAImpLimit	CORBAInitialize	CORBAInternal
CORBAIntFrePos	CORBAInvalid Transaction	CORBAInvFlag	CORBAInvIdentInvOBJRef
CORBAMarshal	CORBANoImplement	CORBANoMemory	CORBANoPermission
CORBANoResources	CORBANoResponse	CORBAOBJAdapter	CORBAObjectNoTexist
CORBAPersistStore	CORBASystem Exception	CORBATransaction Required	CORBATransaction Rolledback
CORBATranslent	CORBAUnion	CORBAUnknown	CORBAUserException
CPlusPlus	DivideByZeroError	DWRuntimeError	EnumerationDefinition
EnumerationItem Definition	Error	ErrorLogging	Exception
ExtObject	JaguarORB	NullObjectError	OLERuntimeError
OLEStorage	OLEStream	OLETxnObject	OMControl
OMCustomControl	OMEmbeddedControl	OMObject	OMStorage
OMStream	ORB	PBTocppObject	Pipeline
ProfileCall	ProfileClass	ProfileLine	ProfileRoutine
Profiling	RemoteObject	ResultSet	ResultSets
RuntimeError	ScriptDefinition	Service	SimpleTypeDefinition
SSLCallBack	SSLServiceProvider	SystemFunctions	Throwable
TraceActivityNode	TraceBeginEnd	TraceError	TraceESQL
TraceFile	TraceGarbageCollect	TraceLine	TraceObject
TraceRoutine	TraceTree	TraceTreeError	TraceTreeESQL
TraceTreeGarbageCollect	TraceTreeLine	TraceTreeNode	TraceTreeObject
TraceTreeRoutine	TraceTreeUser	TraceUser	TransactionServer
Transport	TypeDefinition	VariableCardinality Definition	VariableDefinition

7 PowerScript Reference

7.1 PowerScript Topics

7.1.1 Overview

The PowerBuilder language is PowerScript. PowerScript is used in scripts and user-defined functions to build PowerBuilder applications.

During conversion of a PowerBuilder application, script compiled in PowerScript is converted to JavaScript.

In the following section, PowerScript refers to the PowerBuilder programming language, while JavaScript refers to the programming language used for PowerBuilder-to-Web conversion.

7.1.2 Object-Oriented programming

The application may make use of the powerful object-oriented programming techniques that are typically found in PowerBuilder applications:

- Inheritance

  Creating a new control in a child object by copying and pasting an existing control in the PowerBuilder painter is unsupported if the existing control is inherited from the parent object.

- Encapsulation
- Polymorphism

7.1.3 Language basics

7.1.3.1 Comments

Supported

1. Common Comments

   - Double-slash method: Code // Comment

     ```
     A= B + C// Comment
     ```

   - Slash-and-asterisk method: /* Comment */

     ```
     A= /* comment */ B + C
     ```

2. Embedded comments

   // Comment1 /*Comment2

   /* Comment1 // Comment2 */

   /* Comment1 /* Comment2*/ Comment3 */

For example:

```
A = B + C /* This comment starts here.
/* This is the start of a nested comment. The nested comment ends here. */
The first comment ends here.
*/ + D + E + F
```

Unsupported

None.

7.1.3.2 Identifiers

Supported

Identifiers in Appeon refer to global/instance/local variable names, or object names, or menu names & menu item names. Note that menu names & menu item names are regarded as identifiers.

Rules for identifiers:

- Identifiers can be reserved words in JavaScript, apart from the identifier word "Object".
- Can have up to 40 characters but no spaces.
- Must start with a letter or an _ (underscore) and can include any combination of the letters, numbers and special characters listed in Appeon supported identifiers.
- In the case of duplicate objects, objects earlier in the PBL list overwrite those later in the PBL list.

Unsupported

- Objects of different types cannot have the same names.
- Objects of the same types, even if they are in different PBLs, cannot have the same name.
- The “#” will not be automatically converted to “#” in Appeon.
- Identifiers cannot be reserved words in Appeon: appeondatawindow, appeondatastore, appeonservice, appeon_nvo_db_update, appeonextfuncs, appeonfileservice, ejbserial, ejbobject and parse_retval_object.

7.1.3.2.1 Appeon supported identifiers

Letters, numbers or special characters with Unicode listed below or within the intervals are supported.

- All intervals are closed intervals.
- Cannot start with the letter, number or special character corresponding to the Unicode (or within the intervals) marked with "*".

Table 7.1:

30-39 *	41-5A	5F	61-7A	B7 *	C0-D6
D8-F6	F8-131	134-13E	141-148	14A-17E	180-1C3
1CD-1F0	1F4-1F5	1FA-217	250-2A8	2BB-2C1 *	2D0-2D1 *
300-345 *	360-361 *	386	387*	388-38A	38C
38E-3A1	3A3-3CE	3D0-3D6	3DA	3DC	3DE
3E0	3E2-3F3	401-40C	40E-44F	451-45C	45E-481
483-486 *	490-4C4	4C7-4C8	4CB-4CC	4D0-4EB	4EE-4F5
4F8-4F9	531-556	559 *	561-586	591-5A1 *	5A3-5B9 *
5BB-5BD *	5BF *	5C1-5C2 *	5C4 *	5D0-5EA	5F0-5F2
621-63A	640 *	641-64A	64B-652 *	660-669 *	670 *
671-6B7	6BA-6BE	6C0-6CE	6D0-6D3	6D5	6D6-6E8 *
6EA-6ED *	6F0-6F9 *	901-903 *	905-939	93C-94D *	951-954 *
958-961	962-963 *	966-96F *	981-983 *	985-98C	98F-990
993-9A8	9AA-9B0	9B2	9B6-9B9	9BC *	9BE-9C4 *
9C7-9C8 *	9CB-9CD *	9D7 *	9DC-9DD	9DF-9E1	9E2-9E3 *
9E6-9EF *	9F0-9F1	A02 *	A05-A0A	A0F-A10	A13-A28
A2A-A30	A32-A33	A35-A36	A38-A39	A3C *	A3E-A42 *
A47-A48 *	A4B-A4D *	A59-A5C	A5E-A5E	A66-A74 *	A81-A83 *
A85-A8B	A8D	A8F-A91	A93-AA8	AAA-AB0	AB2-AB3
AB5-AB9	ABC-AC5 *	AC7-AC9 *	ACB-ACD *	AE0	AE6-AEF *
B01-B03 *	B05-B0C	B0F-B10	B13-B28	B2A-B30	B32-B33
B36-B39	B3C-B43 *	B47-B48 *	B4B-B4D *	B56-B57 *	B5C-B5D
B5F-B61	B66-B6F *	B82-B83 *	B85-B8A	B8E-B90	B92-B95
B99-B9A	B9C	B9E-B9F	BA3-BA4	BA8-BAA	BAE-BB5
BB7-BB9	BBE-BC2 *	BC6-BC8 *	BCA-BCD *	BD7 *	BE7-BEF *
C01-C03 *	C05-C0C	C0E-C10	C12-C28	C2A-C33	C35-C39
C3E-C44 *	C46-C48 *	C4A-C4D *	C55-C56 *	C60-C61	C66-C6F *
C82-C83 *	C85-C8C	C8E-C90	C92-CA8	CAA-CB3	CB5-CB9
CBE-CC4 *	CC6-CC8 *	CCA-CCD *	CD5-CD6 *	CDE	CE0-CE1
CE6-CEF *	D02-D03 *	D05-D0C	D0E-D10	D12-D28	D2A-D39
D3E-D43 *	D46-D48 *	D4A-D4D *	D57 *	D60-D61	D66-D6F *
E01-E2E	E30	E31 *	E32-E33	E34-E3A *	E40-E45
E46-E4E *	E50-E59 *	E81-E82	E84	E87-E88	E8A
E8D	E94-E97	E99-E9F	EA1-EA3	EA5	EA7
EAA-EAB	EAD-EAE	EB0-EB9 *	EBB-EBD *	EC0-EC4 *	EC6 *
EC8-ECD *	ED0-ED9 *	F18-F19 *	F20-F29 *	F35 *	F37 *

F39 *	F3E-F3F *	F40-F47	F49-F69	F71-F84 *	F86-F8B *
F90-F95 *	F97 *	F99-FAD *	FB1-FB7 *	FB9 *	10A0-10C5
10D0-10F6	1100	1102-1103	1105-1107	1109	110B-110C
110E-1112	113C	113E	1140	114C	114E
1150	1154-1155	1159	115F-1161	1163	1165
1167	1169	116D-116E	1172-1173	1175	119E
11A8	11AB	11AE-11AF	11B7-11B8	11BA	11BC-11C2
11EB	11F0	11F9	1E00-1E9B	1EA0-1EF9	1F00-1F15
1F18-1F1D	1F20-1F45	1F48-1F4D	1F50-1F57	1F59	1F5B
1F5D	1F5F-1F7D	1F80-1FB4	1FB6-1FBC	1FBE *	1FC2-1FC4
1FC6-1FCC	1FD0-1FD3	1FD6-1FDB	1FE0-1FEC	1FF2-1FF4	1FF6-1FFC
20D0-20DC *	20E1 *	2126 *	212A-212B *	212E *	2180-2182 *
3005 *	3007	3021-302F *	3031-3035 *	3041-3094	3099-309A *
309D-309E *	30A1-30FA	30FC-30FE	3105-312C	4E00-9FA5	AC00-D7A3

7.1.3.3 Labels

Labels and GOTO statements are unsupported.

7.1.3.4 Special ASCII characters

Supported

Table 7.2:

ASCII character	To Specify This	Enter This
Common ASCII characters	Newline	~n
	Tab	~t
	Carriage return	~r In Appeon, "~r" is considered a newline character.
	Formfeed	~f
	Backspace	~b
	Double quote	~"
	Single quote	~'
	Tilde	~~
Any ASCII character	Hexadecimal	~h##
	Octal	~o###

Unsupported

1. In a Web application, a character whose ASCII value is greater than 127 cannot be saved to the database.

2. In PowerBuilder, for characters whose ASCII value is greater than 128, the equal operator will consider them the same. However, in Appeon, the equal operator will not consider them the same.

3. Common ASCII character: Vertical tab (~v).

4. Any ASCII character: Decimal (~###) Note: Appeon ignores the escape character ("~") specified in the PowerBuilder painter. In SQL statements, "~" is handled as escape character.

5. Tilde ("~") on the web may not take effect if it is contained in a nested string that is a variable or it is contained in a string whose nested level is up to two.

7.1.3.5 Null values

Supported

NULL means undefined or unknown. It is not the same as an empty string, a zero, or a date of 0000-00-00. For example, NULL is neither 0 nor "" (empty string).

Null in PowerBuilder is directly translated into Null in JavaScript.

Unsupported

Expressions involving Null values may arrive at different values in JavaScript from their values in PowerScript. For example, the expression A + B in PowerScript will return Null if any of A or B is Null. However, in JavaScript, the expression will not return Null even if A or B is Null. In addition, an arithmetic or relational operation involving a null value always returns null in PowerBuilder. As shown in the following table, the return value is not always null in JavaScript:

Table 7.3:

Operation		Return Value	
Assuming SetNull(A), SetNull(B)		PowerBuilder	JavaScript
Arithmetic	A+1	Null	1
	A+B	Null	0
	A*B	Null	0
Relational	A=1	Null	False
	A<>1	Null	True
	NOT (A=1)	Null	True
	A=A	Null	True
	A=B	Null	True
	IsNull(A=1)	True	False
String concatenation	A+"ABC"	Null	"NULLABC"
	A+B	Null	"NULLNULL"

7.1.3.6 Reserved words

Supported Reserved Words

Table 7.4:

and	call	case	catch	choose	close	commit	connect
constant	continue	create	cursor	declare	delete	describe	disconnect
do	dynamic	else	elseif	end	event	execute	exit
false	fetch	first	for	forward	from	function	global
halt	if	immediate	insert	into	is	last	loop
next	not	of	on	open	or	parent	post
prior	prepare	ref	return	rollback	rpcfunc	select	step
subroutine	super	then	this	to	trigger	true	try
type	until	update	using	while	with	within	selectblob
updateblob							

- The reserved word HALT is supported, but the code after the Halt statement will not be executed in Appeon Mobile.
- PROCEDURE can only be used in the DECLARE Procedure SQL statement.

Unsupported Reserved Words

Table 7.5:

enumerated	external	finally	goto	indirect	intrinsic	library	system
systemread	systemwrite	throw	throws	Eon	EonApp	EonObject	EonStatic
EonMenu	PB	PBGlobal	PBArray				

7.1.3.7 Pronouns

Supported

The following pronouns in PowerScript are supported:

- Parent - refers to the object that contains the current object.

  Using Parent in the script for a visual user object is supported.
- This - refers to the window, user object, menu, application object, or control that owns the current script.
- Super - refer to the immediate ancestor for a descendant object or control.

Unsupported

None.

7.1.3.8 Statement continuation & separation

Supported

- The statement continuation character (&) and syntax are supported. The syntax is as follows:

```
Start of statement &
   more statement &
end of statement
```

- The statement separation character (;) and syntax are supported. The syntax is as follows:

```
Statement1; statement2
```

Note: White Space (Blanks, tabs, formfeeds, and comments) in the statement is supported and treated the same way as in PowerBuilder.

Unsupported

None.

7.1.4 Data types

7.1.4.1 Standard data types

Supported

Appeon supports all standard PowerBuilder data types as outlined in the table below:

Table 7.6:

Blob	Boolean	Byte	Char or character	Date
DateTime	Decimal or Dec*	Double	Integer or Int	Long
Real	String	Time	UnsignedInteger, UnsignedInt, or Uint	UnsignedLong or ULong

The decimal precision is extended from 15 digits to 28 digits in DataWindow. However, you cannot manipulate 28-digit Decimal in a datawindow via Get/Set function and dot notation.

Decimal in DataWindow

The decimal precision extends to 28 digits in a DataWindow by enabling the 28-digital Decimal option in AEM.

1. Precision

   With 28-digit Decimal option, the decimal precision in DataWindow supports up to 28 digits and ranges from -39,614,081,257,132,168,796,771,975,167 to +39,614,081,257,132,168,796,771,975,167.

2. Function Limitations

   - 28-digit Decimal supports the following arithmetic operators:

     Arithmetic operators: “+”, “-“, “*”, “/”

     Relational operators: =, >, <, <>, >=, <=

   - 28-digit Decimal is supported in the following DataWindow expressions: Abs, Avg, CumulativeSum, Median, Sign, and Sum.

3. Function Differences

   - In the calculation, values/variables of Double or Real data type are handled as values/variables of Decimal data type.
   - Constants in an expression will be handled as Double datatype if the expression is being dynamically modified and contains relational operators (=, >, <, <>, >=, <=).
   - Constants in an expression will be handled as Decimal data type if the expression is being dynamically modified and contains arithmetic operators ("+", "-", "*", "/").
   - Appending the letter D in uppercase or lowercase to identify a number as a decimal constant in a DataWindow expression is unsupported in 28-digit decimal.
   - Decimal displaying in the DataWindow in Appeon may differ from that in PowerBuilder. Sometimes Decimal displays in scientific notation in PowerBuilder but displays in standard notation in Appeon.
   - If exceeding 28 digits, the decimal number cannot be correctly displayed on the Web.

Differences

- UnsignedLong (32-bit) and UnsignedInteger (16-bit)

  The UnsignedLong (32-bit) and UnsignedInteger (16-bit) data types are handled as Long and Integer. Therefore, the supported range for UnsignedLong is from 0 to 4,294,967,295, and the supported range for UnsignedInteger is from 0 to 65,535.

- In operations that contain calculations of precision, values/variables of Decimal or Real data type are handled as values/variables of Double data type.

  In operations that do not contain calculations of precision, values/variables of Double or Real data type are handled as values/variables of Decimal data type.

- You must ensure that the result of a numeric expression will not cause overflow in PowerBuilder. Otherwise the result on the Web is different from that in PowerBuilder.

7.1.4.2 Any data type

Any data type is fully supported, with the exception as follows:

- Assigning the return value of a String function to an Any variable that has not been previously assigned is unsupported.
- Using Any in the OCX is unsupported.
- Using Any array is only supported in the retrieval argument.
- Using Any array in a dot notation is unsupported. To work around it, you can using Any variable instead of Any array.

  For example:

  Unsupported code:

```
la_data[] = dw_1.object.data
```

Supported code:

```
any la_data1
la_data1 = dw_1.object.data
```

7.1.4.3 System object data types

Supported

- Autoinstantiated object data types:

  NVOs (Class User Object) with their Autoinstantiate property checked;

  Structure data types;

  Autoinstantiated system objects: ListViewItem, TreeViewItem.

- Nonautoinstantiated object data types:

  NVOs (Class User Object) with their Autoinstantiate property unchecked;

  Visual controls (user-defined visual controls and system controls)

  Nonautoinstantiated system objects, including: Application, DataStore, DataWindowChild, DynamicStagingArea, MDIClient, Menu, Message, Transaction Object, Window.

- PowerObject classes:

  The supported PowerObject classes include DragObject, DWObject, GraphicObject, PowerObject, and WindowObject.

  Note: In the Web application, because the declaration of a PowerObject does not specify the object type, the initial value of the variable is NULL. Therefore, when the IsValid function is executed, IsValid returns NULL.

Unsupported

- Among all the supported system objects and system controls, if the data type is a non-global Transaction Object, the data type is not supported.
- If the data type is an unsupported system object or system control, it is unsupported.
- Do not refer to a DataWindow object by using global functions or instance variables. Otherwise the executing result will has error on the Web.

7.1.4.4 Enumerated data types

Supported

All system-defined enumerated data types and values are supported, unless the enumerated data type belongs to an unsupported feature.

Unsupported

Enumerated data types have predefined sets of values. Most of the PowerBuilder enumerated data types have default values. Using the default values of enumerated data types is unsupported.

7.1.4.5 Forced conversion between data types

Supported

1. Forced conversion between string and char data type:

   - Assigning strings to char arrays
   - Assigning char arrays to strings
   - When a string literal is assigned to a char variable, the first character of the string literal is assigned to the variable. For example:

     ```
     char c = "xyz" //results in the character x
     being assigned to the char variable c.
     ```

   - Special characters (such as new line, form feed, octal, hex, etc.) can be assigned to char variables using string conversion. For example:

     ```
     char c = "~n"
     ```

   - A char variable assigned to a string variable results in a one-character string.
   - Expressions using both strings and char arrays promote the chars to strings before evaluation. For example:

     ```
     char c
     if (c = "x") then
     // promotes the contents of c to a string
     before comparison with the string "x"
     ```

   - Using chars in PowerScript functions. All PowerScript functions that take strings also take chars, subject to the above described conversion rules.

2. Forced conversion between numeric data types:

   - The order of precedence in PowerBuilder regarding numeric data types is supported. The following is the order of precedence from highest to lowest (based on the range of values for each data type):

     (High) Double >> Real >> Decimal >> UnsignedLong >> Long >> UnsignedInteger >> Integer (Low)

   - If operands in an expression have different data types, the value whose type has lower precedence is converted to the data type with higher precedence. For example:

     ```
     int x; x=2.4
     PowerScript result: the value of x is actually 2,
     because x is defined as an integer.
     PowerBuilder will cut the digit after the decimal point,
     then assign 2 to x.
     ```

   - Unsigned type has precedence over signed. So if one operand is signed and the other is unsigned, both are promoted to the unsigned version of the higher type. For example, if one operator is a Long and another an UnsignedInteger, both are promoted to UnsignedLong.

- Transform other numeric data types recessive to integer or int data. A force conversion from other numeric data to int/long data will arrive at the same value in PowerScript and JavaScript.

Unsupported

- In PowerBuilder, when Time data type is automatically converted into DateTime, the date is always 1900-01-01. Thus, the DataWindow always gets a value of DateTime type: Retrieve (1900-01-01 xx:xx:xx).However, if connected to a JDBC SQL Server, the default date is the current day. In other words, DataWindow gets Retrieve(today xx:xx:xx) when Time data type is automatically converted into DateTime.

Difference

When assigning a numeric number to an Any data, a forced conversion from Any to Integer may arrive at different values in PowerBuilder and on the Web. For example,

```
Integer li_value
any la_value = 3.5
li_value = Integer(la_value) // li_value will be 3 on the Web and 4 in
 PowerBuilder.
```

7.1.5 Declarations

7.1.5.1 Variables and constants

Table 7.7:

Variable scope	**Supported**
	Global, local and instance
	Unsupported
	Shared
Variable declaration syntax	**Supported**

	`datatype { { size } } { { precision } }`
	`variablename { = value } {, variablename2 { = value2 } }`
	The datatype can be any standard type and system object type.
	Declaring multiple variables of the same type at one time is supported. For example:
	`integer li_a=5, li_b=10.`
	Naming variables with Non-English characters or numbers.

	Unsupported
	When a global decimal variable is declared, or a decimal constant is used to declare a variable, the specification of precision for the global decimal or the decimal constant will be ignored. A variable having the same name as a global variable is or a global function is unsupported.

	A global variable cannot have the same name as a control in a window. For example, if there is a GroupBox control named as gb_1, the following syntax is unsupported: Global Boolean gb_1. Placing a cursor or stored procedure declaration in the declaration of instance variables is unsupported.
Global scope operator (::)	**Unsupported**
	Referring to a global variable by using the global scope operator (::) before the variable name is unsupported; (i.e. The syntax with "::globalname" is unsupported).
Constant type	**Supported**
	All of the standard data types
Constant declaration syntax	**Supported**

	`CONSTANT datatype constname = value`
	The constant can only be public.
	It is supported if the value is an expression.
	For example:
	`constant date ld_date = today()`

Access to Instance Variables	**Unsupported**
	When the instance variable is passed by reference to a script and before the execution of this script is finished, if the value of this instance variable is changed, Appeon cannot capture the changes. When the instance variable is defined in a custom class, it cannot be accessed directly using the class dot notation, for example, the following script will not work on Web: mle_Result.Text = n_cst_test.COMPANYNAME, n_cst_test is the name of the custom class and COMPANYNAME is its instance variable. But you can work it around by creating an object first and then using the object to access the instance variable.
Initial values of variables and constants	**Supported**
	When a variable or constant is declared, a default initial value is automatically assigned or an initial value can be specified in the declaration. The initial values of enumerated data types are converted to null in JavaScript. Except for the initial values of enumerated data types and the Any data type, the default initial values in PowerScript are supported.
	Unsupported
	There are different rules in PowerBuilder than in JavaScript if specifying an expression as an initial value: It is unsupported to use the instance constants of a non-instantiated object.

	With the syntax datatype variable = expression, in PowerBuilder, the expression's value is assigned to the variable when the script is compiled (not during execution). In JavaScript, the expression's value is set to the variable during execution.
	For example, if the declaration is the following:
	`date d_date = Today( )`
	The value of d_date is the date when the script is compiled in PowerBuilder, and it is the date when the application is running in JavaScript.
	Therefore, making the declaration of a variable and assigning it with the initial value in separate PowerScript statements is recommended.
	For example:
	`date d_date`
	`d_date = Today( )`

7.1.5.2 Arrays

7.1.5.2.1 Declaration syntax

- The declaration syntax for arrays is supported:

```
{ access } datatype variablename { d1, ..., dn } { = { valuelist } }
```

- The access is always PUBLIC.
- The datatype cannot be an unsupported data type (refer to the data type section). For decimals, you can specify the precision of the data by including an optional value in brackets after the datatype. For example, decimal {2} ld_prices[ ].
- Both variable-size arrays and fixed-size arrays are supported.

```
date ld_birthdays[ ]
```

```
string ls_array[10 ]
```

- Arrays with specified lower bound and upper bound are supported, even if the lower bound is a non-integer or a negative value. If the lower bound is not an integer, it will be rounded off.

```
string ls_name[-10 to 15]
```

- Both single-dimensional and multi-dimensional arrays are supported.

```
integer ls_array[10]
```

```
integer li_score[2,3]
```

- Using TO to change array index values is supported.

```
integer li_staff[100, 0 to 20, - 3 to 5]
```

- Multiple arrays of the same type can be declared simultaneously.

```
string ls_array1[10], ls_array2[5], ls_array3[100]
```

- Assigning an array to an array is unsupported if the array elements of Any data type. For example,

```
any la_1[3], la_2[4]
...
la_1[1] = la_2[4] // Unsupported
```

7.1.5.2.2 Initialization and assignment

Supported

- Each element of an array can be initialized to the same default value as its underlying data type. The default value for string data type elements is ""; the default value for numeric data type element is 0.

- The default length of variable-size arrays is 0, which means that the array does not have any elements. Initializing several elements of variable-size array is supported if the element after the several elements is assigned with a value. For example:

```
integer li_array []
li_array[8]=8         //The values of li_array[1], …,
li_array[7] are initialized to the default value 0
```

- Using arraylists to assign values to an array, or assign values to array elements separately, is supported. There can be expression(s) in the arraylist. The result will be the same as in PowerBuilder.

```
li_array[10] = {1,2,3,4,5}
li_array[3,2] = {1,2,3,4,5}
ld_date = {today(),relativedate(today(),1), 2002-12-31}
```

- Assigning one array to the another is supported. The result will be the same as in PowerBuilder.

  - Assigning a variable-size array to another variable-size array. For example:

    ```
    integer li_test1[]={1,2,3,4,5}
    integer li_test2[] ={10,20,21,22,23,24,25,26,27}
    …
    li_test2=li_test1
    ```

    If the array type is NVO, the assignment operation does not trigger the Constructor event.

  - Assigning a fixed-size array to another fixed-size array. It can be between one-dimensional arrays, or between one-dimensional array and a multi-dimensional array, or between multi-dimensional arrays. For example:

    ```
    integer li_test0[10] ={21,22,23,24,25,26,27,28}
    integer li_test1[5]={1,2,3,4,5}
    integer li_test2[2,6] = {11,12,13,14,15,16,17,18}
    integer li_test3[2,3,5] = {11,12,13,14,15,16,17,18}
    ```

```
li_test1=li_test0 //assign a one-dimensional array to another one-dimensional
 array
li_test2=li_test1 //assign a one-dimensional array to a multi-dimensional array
li_test1=li_test2 //assign a two-dimensional array to a one-dimensional array
li_test3=li_test2 //assign a multi-dimensional array to another multi-
dimensional array
```

- Assigning a variable-size array to a fixed-size array. For example:

```
integer li_test0[] ={21,22,23,24,25,26,27,28}
integer li_test1[10]={1,2,3,4,5}
li_test1=li_test0
```

- Assigning a fixed-size array to a variable-size array. For example:

```
integer li_test1[10]={1,2,3,4,5}
integer li_test0[]
li_test0[30] = 100
li_test0=li_test1 //the element number of the
li_test0 will be 10
```

- Reinitializing arrays with the following steps is supported:

  1. Declare a dummy array of the same type (never put any values into the array).
  2. Simply set originalarray = dummyarray.

  This will clear out the original array and cause UpperBound to return the correct value. The same rule applies to PowerBuilder.

- The index of an array can be an expression. For example:

```
for A = 1 to upperbound(ls_array)
ls_array[A,1] = ......
end for
ls_array[integer(ltvi_item.data)] =123
```

Unsupported

- Assigning values between instance variables of structure type is unsupported. For example, the following lines of code are unsupported:

```
Str_a L1
Str_b L2
L1 = L2
```

- Assigning structure arrays to non-structure arrays is unsupported. For example, the following lines of codes are unsupported:

```
str_dwstruct lst_dwstr[]
powerobject po_ary[]
po_ary = lst_dwstr // Unsupported
```

7.1.5.2.3 Passing arrays as arguments

Passing a variable-size or fixed-size array as an argument is supported. For example:

```
uf_convertarray(a)    //integer a[]
uf_convertarray(a)   //integer a[10]
```

The following two syntaxes will get the same result for one-dimensional, multi-dimensional, or variable-size arrays:

```
li_upper = upperbound(ls_array)     // string ls_array[]
li_upper = upperbound(ls_array [])  // string ls_array[]
```

7.1.5.2.4 Complex arrays

Complex arrays

- Structure arrays are supported. The declaration, initialization and assignment of structure arrays comply with the general rules for standard arrays.
- UserObject arrays are supported. Please refer to the User Object section for details.
- Nested arrays are supported.
- Enumerated type arrays are supported.

7.1.5.2.5 Unsupported

If arguments of a function are arrays, arguments and corresponding actual parameters cannot be of different dimensions.

7.1.5.3 External functions

Declare external functions

Supported

- Declaring and calling local external functions
- Declaring and calling global external functions
- Data types of external function arguments can be the following:

  Array, Boolean, Blob, Char, Date, DateTime, Double, Decimal, Integer, Long, Real, String, Structure, and Void
- Data types of external function return values can be the following:

  Array, Boolean, Blob, Char, Date, DateTime, Double, Decimal, Integer, Long, Real, String, Structure, Unsigned Integer, Unsigned Long, and Void

7.1.6 Operators & expressions

Supported operators and expressions

- Arithmetic operators for numeric data types: +, -, *, /, ^, ++, --, +=, -=, *=, /=, ^=
- Relational operators for all data types: =, >, <, <>, >=, <=, NOT, AND, OR
- The concatenation operator for String data types: +
- DataWindow expressions are supported

- The supported features of operators & expressions are related to null values and forced conversion.

Precedence of supported operators

These operators will follow the order of precedence listed in the table below:

Table 7.8:

Order	Operator	Purpose
1	( )	Grouping
2	+, -, NOT	Unary plus (indicates positive number), unary minus (indicates negative number), Negation
3	^	Exponentiation
4	*, /	Multiplication and division
5	+, -	Addition and subtraction; string concatenation
6	>, <, <=, >=	Relational operators
7	=,	Relational operators
8	AND	Logical and
9	OR	Logical or

Unsupported

Avoid data overflow in the application. If there is data overflow, the Web application handles the data differently from the PowerBuilder application.

7.1.7 Structures

7.1.7.1 Definition and declaration of structures

Defining a structure in the Structure painter or an object painter is supported:

- If the structure is defined in the Structure painter, the structure is global structure, and instances of the structure can be declared in a script or in an object's instance variable.
- If the structure is defined in an object painter, the structure is an object structure, and instances of the structure can only be declared in the object's instance variables and script. An object structure can be defined in any of the following painters: Application, Window, Menu, Function or UserObject.

Declaring global or local instances for global structures is supported. Declaring instance and local instances for object structures is also supported. Several instances for a structure can be declared at one time.

For example:

```
str_emp_data str_emp1, str_emp2
```

Declaring structure arrays is supported.

The arrays can be single or multi-dimensional, variable-sized or fixed sized. The supported features for structure arrays are the same as the supported features for any other arrays.

For example:

```
s_ employee lstr_temp[10] s_ employee lstr_employee[]
```

If an object structure's name is identical to that of a global structure, the object structure has higher priority than the global structure (same as in PowerBuilder).

7.1.7.2 Referring to structure variables

The following syntaxes are supported:

```
structurename.variable   //referring to a structure variable by using dot notations
objectname.structurename.variable   //referring to a variable of an object
 structure
```

For example:

```
str_emp1.emp_lname = "Jones"
This.str_cust1.name
w_customer.str_cust1.name
```

7.1.7.3 Initialization and assignment of structure variables

For each variable of a structure, the variable will be initialized with default values according to its data type. If it is a string type variable, the default value is ""; if it is numeric type variable, the default value is 0.

A structure instance can be assigned when it is declared or after it is declared. For example:

```
s_employee lstr_ouremployee1, lstr_outemployee2
s_employee lstr_youremployee = lstr_ouremployee1
lstr_youremployee = lstr_ouremployee2
```

As in PowerBuilder, when assigning one structure to another, the whole structure is copied so that there are two copies of the structure.

7.1.7.4 Passing structures as arguments

Passing structures as arguments by value, by reference, or as read-only is supported.

7.1.7.5 Complex structures

The following types of complex structures are supported:

- Structures which have one or more of their structure variables being referred to another structure.
- One or more structure variables which are autoinstantiated or non-autoinstantiated objects. For example:

  ```
  s_employee lstr_employee[]    //the structure
  s_employee is defined with a DataWindow type variable
  long ll_employee
  lstr_employee[1].adw_employee=w_employee.dw_1
  ll_employee=lstr_employee[1].adw_employee.Retrieve()
  ```

- One or more structure variables which are arrays that have no unsupported features.

7.1.7.6 Unsupported

- Cross-definition of structures. For example, structure A has a structure variable that is structure B, while structure B has a structure variable that is structure A.
- Inheritance of an object structure is unsupported; inheritance of an object structure instances is unsupported as well.
- If one or more member variables in the structure are unsupported data types, Appeon still supports the structure as long as unsupported variables are not used.
- If there are member variables of enumerated data type in the structure, the default values of the enumerated variables are unsupported.
- Comments of the structure are unsupported.

7.1.8 User objects

7.1.8.1 User objects

Important Requirements

- For standard class user objects:
  1. Standard class user objects can only inherit from the following non-visual system objects: DataStore, Transaction Object, DynamicStagingArea and OLEObject.
  2. Non-visual standard class user objects must be defined in a PowerBuilder painter. They can be dynamically created (for example, by using the CREATE statement).
  3. If a non-visual object is a local variable, the Destructor event in the non-visual object cannot be triggered unless there is a Destroy statement for the non-visual object as well.
- For custom class user objects:
  1. Non-visual custom class user objects must be defined in PowerBuilder painter. They can be dynamically created (for example, by using the CREATE statement).
  2. The specification of EAServer Project properties is unsupported.
  3. If a non-visual object is a local variable, the Destructor event in the non-visual object cannot be triggered unless there is a Destroy statement for the non-visual object as well.
- For standard and custom visual user objects:
  1. Must be defined in PowerBuilder painter.
  2. The SetFocus function is not supported for custom visual user objects, but is supported for standard visual user objects.
  3. Standard Visual Object is an extension of the visual system object (control), and it is used to customize the function of the visual system object (control). For more details, please refer to System Objects and System Controls.

Supported

- Custom class user objects
- Custom visual user objects
- Standard class user objects
- Standard visual user objects

Unsupported

1. External visual user objects

   In the PowerBuilder application, the Destructor event sequence for a user object will be triggered in accordance with the Control[] property of the user object. In the Web application, the Destructor event sequence is unsupported.

2. Nonvisual objects (Custom Class Objects and Standard Class Objects)

   To insert nonvisual object(s) between objects (window, user object, NVO, application), you can select any items on the *Insert menu* | *Object menu* of the PowerBuilder painter. However, Appeon does not support this.

7.1.8.2 Autoinstantiated NVO

- Declaring an autoinstantiated NVO

  1. Declaring an autoinstantiated user object creates an instance of that object (just like a structure), and the Constructor events are triggered for the instance variables.
  2. If an instance variable contains an autoinstantiated NVO(b) and the Constructor event is triggered for the instance variable, an instance of NVO(b) is automatically created, and the Constructor events are triggered for the instance variables of NVO(b).

- Assignment for autoinstantiated NVO

  1. When an autoinstantiated object is assigned to another autoinstantiated object, the whole object is copied to the second variable. For example:

```
n_cst_string lnv_string1, lnv_string2

lnv_string2 = lnv_string1  //lnv_string2 is a
copy of lnv_string 1
```

  2. Assigning a NonVisualObject object to an autoinstantiated NVO or a NonVisualObject object is **unsupported** (please note that NonVisualObject is a system object and it is different from an NVO (non visual user object)). For example:

```
NonVisualObject lnv_test  //lnv_test is a
NonVisualObject object

n_cst_string lnv_string  //n_cst_string is an
autoinstantiated NVO

lnv_test = lnv_string   //unsupported
```

- Defining autoinstantiated NVO

  The following can be included in the definition of an autoinstantiated NVO:

  1. Instance variable. The instance variable can be an autoinstantiated NVO, an object, or have the same name as a window instance variable
  2. System function or user defined function or object function
  3. Constructor event, or object event or user-defined event
  4. Using a Destructor event in the definition of an autoinstantiated NVO is unsupported.

- Autoinstantiated NVO array

  1. If the autoinstantiated NVO array is a fixed-size array and the array is declared, instances of each NVO are created, the instance variables of each NVO are instantiated, and the Constructor event is triggered for each instance variable. For example:

     ```
     n_cst_string lnv_string[10]
     ```

  2. If the autoinstantiated NVO array is variable-size array, the NVO instances are not created when the array is declared. When an array element is named (an NVO), the instances of the element and the foregoing elements are created, the instance variables of each NVO are instantiated, and the Constructor event is triggered for each instance variable. For example:

     ```
     n_cst_string lnv_string[]
     lnv_string[10].is_source = ls_model
     ```

- Autoinstantiated NVO as a structure member

  When the structure is declared, an instance of the autoinstantiated NVO is created. When the structure is called, the Constructor event is not triggered for the NVO. In PowerBuilder, the Constructor event is triggered when the NVO is first used.

- Inheritance

  Inheritance is supported if the ancestor of an autoinstantiate NVO is a nonautoinstantiated NVO.

7.1.8.3 Nonautoinstantiated NVO

- Declaring a non-autoinstantiated NVO

  1. Declaring a non-autoinstantiated user object does not create an object instance until there is a CREATE statement. Declaring an object variable declares an object reference. Example:

     ```
     n_base lnv_base     //n_ base is a nonautoinstantiated NVO
     lnv_base = Create n_base    //Create an instance of n_base
     ```

  2. When the object instance is created, the instance variables of the NVO are instantiated, and the Constructor event is triggered for each instance variable.

3. Instantiating an ancestor variable with an instance of one of its descendants is supported. Example:

```
n_base lnv_base  //n_ base is a nonautoinstantiated NVO
lnv_base = Create using "n_cst_sqlspy"  //n_cst_sqlspy is a descendant of
 n_base
```

- Assignment for nonautoinstantiated NVO

1. When a non-autoinstantiated object is assigned to another non-autoinstantiated object, a reference to the object instance is copied. Only one copy of the object exists. For example:

```
n_cst_string lnv_string1, lnv_string2
lnv_string2 = lnv_string1  //Both point to same object instance
```

2. Unlike autoinstantiated NVOs, assigning a NonVisualObject object to a non-autoinstantiated NVO or assign a non-autoinstantiated NVO to a NonVisualObject object is **supported**. For example:

```
NonVisualObject lnv_test    //lnv_test is a NonVisualObject object
n_cst_string lnv_string2   //n_cst_string is a non-autoinstantiated NVO
lnv_test = lnv_string2   //supported
```

- Defining non-autoinstantiated NVOs

  The following can be included in the definition of a non-autoinstantiated NVO:

1. Instance variables. An instance variable can be an autoinstantiated NVO, an object, or have the same name as a window instance variable

2. System, user defined, or object functions

3. Constructor, Destructor, object, and user-defined events. The Destructor event cannot be triggered unless there is a Destroy statement for the object as well.

- Non-autoinstantiated NVO array

  There can be fixed-size or variable size non-autoinstantiated NVO arrays. When the array is declared, an instance of the object is not created. A non-autoinstantiated NVO will only be created when there is a CREATE statement for the NVO.

- Non-autoinstantiated NVO as a structure member

1. Defining a non-autoinstantiated NVO in a structure is supported. Example:

```
global type str_model from structure
string s_emplid
Date ld_inputday
n_cst_base lnv_base
end type
```

2. When the structure is declared, an instance of the non-autoinstantiated NVO is not created. A non-autoinstantiated NVO will only be created when there is a CREATE statement for the NVO.

7.1.9 Calling functions and events

7.1.9.1 Syntax for calling functions and events

Supported

- The syntax used to call all PowerBuilder functions and events:

```
{objectname.} {type} {calltype} {when} name
({argumentlist})
```

  If the *calltype* argument is DYNAMIC, it is unsupported to specify a reference argument in the argumentlist.

- The syntax used to call functions and events in an object's ancestor:

```
{objectname.} ancestorclass ::{type}
{when} name ({argumentlist})
```

- If the function name is not qualified with an object or a control, PowerBuilder searches for the function and executes the first one it finds that matches the name and arguments. This is supported in Appeon.

Unsupported

- Referring to a global function by using the global scope operator (::) before the function name is unsupported.

7.1.9.2 Triggering & Posting

Table 7.9:

Triggering	**Supported**
	• Triggering for functions • Triggering for events
	Unsupported
	For application and message objects, triggering for functions and events are unsupported.
Posting	**Supported**

	• Posting for functions
	• Posting for events
	• Posting function B that is called inside function A. For example:
	`Function A ()`
	`{`
	`......`
	`Post Function B () //unsupported to post function B`
	`......`
	`}`

	Unsupported

	Posting a function containing reference arguments, which is a local variable is unsupported. Please use the instance variable to replace the local variable to work around this unsupported feature.
Post function	**Supported**

	There are three POST syntax supported, and two of them are supported with limitations.
	• Post Close(window)
	• Post Open(window)
	Limitations: the window argument cannot be an array variable. For example:
	`//Unsupported`
	`window winname[2]`
	`.....`
	`Post open(winname[1],....)`
	`- Post open(windowvariable)`
	• Post user_function()
	Limitations: The reference argument cannot be a local variable. For example:
	`//Unsupported`
	`Integer gf_string (ref string as_parm1)`
	`Post gf_String(ls_Parm) //`
	`ls_parm is a local variable.`

7.1.9.3 Static & dynamic calls

Supported

- Static calls to functions
- Static calls to events
- Dynamic calls to functions
- Dynamic calls to events

Unsupported

- Dynamic calling for overloaded functions is unsupported.
- Dynamically calling a function that has an argument passed by reference is unsupported .
- If a function is dynamically called, its return value cannot be passed as an argument of another function.
- Nested call of more than one layer dynamic code is unsupported. For example:

```
ll_row2 = invo_test.dynamic of_dynamic1
(invo_test.dynamic of_dynamic2(ll_row))
```

The workaround is to execute the dynamic nested call separately. The above example should be modified as below:

```
tt = invo_test.dynamic of_dynamic2(ll_row)
ll_row2 = invo_test.dynamic of_dynamic1(tt);
```

7.1.9.4 Overloading, overriding, and extending functions and events

Supported

- Function overriding is supported.
- Extending and overriding events are supported.

Unsupported

- Overloading system functions is unsupported.
- Dynamic calling for overloaded functions is unsupported.
- Overloading a function that has a dot notation as an argument is unsupported. For example, overloading the following function is unsupported:

  ```
  wf_getname(dw_1.object.s_id[1])
  ```

- Using the local variable AncestorReturnValue in an event of a descendant object is unsupported, unless the event of the descendant object is an extended event from the ancestor object.
- The following scenario is unsupported:

  In object A (parent object), function g() calls function f(type1 arg 1).

  ```
  g()
  {
  f(type1 arg1);
  }
  ```

  In object B (child object), function f(type2 arg2) is the overloading function of function f(type1 arg 1), and object B inherits function g() from object A.

7.1.9.5 Passing arguments to functions and events

Passing arguments

There are three ways to pass arguments to functions and events:

- By value
- By reference
- Read-only

Arguments can be passed with one **limitation** that each function or event can have a maximum of 20 arguments. If the number of arguments exceeds 20, the arguments after the 20th argument will be invalid.

Unsupported

- If both function A and function B have an argument passed by reference, calling function A that has one argument calling to function B, while the argument (that is passed by reference) in the two functions uses the same variable, is unsupported. For example:

  Supported syntax:

  ```
  f(int a, ref int b); g(ref int a)
  ```

  Unsupported syntax:

  ```
  f(g(a),a);
  ```

- Function A has two arguments passed by reference. It is unsupported for the two arguments to use the same variable. For example:

  Unsupported syntax:

  ```
  f(a,a) //f(ref int a,ref int b);
  ```

- Passing a property dot notation as the function argument is unsupported if the property refers to an object, however, this can be worked around. For example:

  Unsupported syntax:

  ```
  lvn_security.Of_setmenuright(this.MENUID)
  ```

  Workaround:

  ```
  menu m_1=this.MENUID
  lvn_security.Of_setmenuright(m_1)
  ```

- Passing an argument that is an object property by reference is unsupported.

  Unsupported example:

  ```
  /*Define a function of display()*/
  public Function string of_display(ref string str_data)
  ......//The code in the function
  return str_data
  end function
  /*Call to the function in the Clicked event of a CommandButton control*/
  string ls_string1
  ls_string1 = of_display(this.text)
  ```

7.1.9.6 Using return values of functions and events

Supported

To use the return value, assign it to a variable of the appropriate data type or call the function where you can use a value of that data type:

- Return values for built-in PowerScript functions
- User-defined functions or events that have return values
- Return values for system events
- User-defined events that have return values

- The function has parameters of standard data type passed by reference and the return value is used as a condition in RETURN, IF ... THEN, CHOOSE...ASE, or DO ... LOOP statements.
- The function has parameters of object data type passed by value and the return value is used as a condition in RETURN, IF ... THEN, CHOOSE...ASE, or DO ... LOOP statements.
- The return value of one function is used as the parameter of another function, for example, func1(func2()).

To use cascaded call and return values:

- It is supported to get/set the property of an object that is the return value of a function:

  Syntax:

  ```
  function.property
  ```

  For example:

  ```
  ParentWindow( ).Enabled = TRUE
  ```

- It is supported to call the function of an object that is the return value of a function.

  Syntax:

  ```
  function1.function2
  ```

  For example:

  ```
  ParentWindow( ).Hide( )
  Ls_test = String(m_main.GetParent( ).ClassName
  ( )
  ```

Limitations of using cascaded call and return values:

- It is unsupported to use the DYNAMIC keyword.
- Except the first call in the chain of cascaded calls, it is unsupported to use reference arguments.

7.1.10 Document Interface

Important Requirements

The following are limitations for using MDI and SDI:

- When a sheet window is opened within MDI, the toolbar of the sheet window will be added as a whole new row below the toolbar of the MDI window. (This is more limited than in PowerBuilder, where you can choose for the sheet toolbar to be added after the MDI toolbar, on the left/right of the screen, to be floating, etc.)
- Appeon supports the toolbar if the window is opened by opensheet () function, but please don't use open() function. And the opensheet toolbar will be displayed right under the toolbar of the MDI window.
- SDI can have only one menu and one toolbar.

Supported

- Both MDI and SDI are supported, and both multiple MDI windows and multiple MDI frames are supported.
- An MDI window can have its menu and toolbar, and every sheet window can have its own menu and toolbar, exactly like in PowerBuilder.
- When a sheet window is opened within MDI, the menu of the sheet window will replace the menu of the MDI window. (This is called a menu switch, just as it is in PowerBuilder).
- Appending the names of open sheets (Window list) to a menu item is supported.

7.2 PowerScript statements

Supported

The following table shows the supported PowerScript statements with examples:

Table 7.10:

Statement	Examples
Assignment	`a = b + 2` Note: There must not be any space between the following operators: ++, --, +=, -=, *=, /=, ^=.
CALL	CALL ancestorobject {controlname}::event Call super:: *eventname* Example: `Call super::clicked` Note: It is supported to use the local variable AncestorReturnValue in an event of a descendant object, if the AncestorReturnValue is generated in a Call Super statement. Call windowname:: eventname Example: `Call w_parent::ue_ok` Call windowname`Controlname:: eventname Example: `Call w_parent`dw_1::ue_retrieve`

CHOOSE...CASE	`CHOOSE CASE testexpression`
	`CASE expressionlist`
	`statementblock`
	`...`
	`{ CASE expressionlist`
	`statementblock`
	`CASE expressionlist`
	`statementblock }`
	`CASE ELSE statementblock }`
	`END CHOOSE`

	Notes: Expressionlist can be one of the following expressions: 1. A single value 2. A list of values separated by commas (such as 2, 4, 6, 8) 3. A TO clause (such as 1 TO 30) 4. IS followed by a relational operator and comparison value (such as IS>5) 5. Functions 6. Any combination of the above with an implied OR between expressions (such as 1, 3, 5, 7, 9, 27 TO 33, IS >42)

CONTINUE	`integer A=1, B=1`
	`DO WHILE A < 10`
	`A ++`
	`IF A < 3 THEN CONTINUE`
	`B+=A`
	`LOOP`

CREATE	CREATE
	Support creating object instances for all objects except for PowerObject object, GraphicObject object, WindowObject object, DragObject object, DrawObject object, Function_object object and SystemFunctions object.
	CREATE USING
	Support dynamically creating object instances except for:
	1. Dynamically creating PowerObject object, GraphicObject object, WindowObject object, DragObject object, DrawObject object, Function_object object and SystemFunctions object.
	Example:
	`UserObject luo_1`
	`luo_1 = create using "PowerObject"`
	2. Dynamically creating Transaction object.
	Example:
	`lds_main = Create using`
	`"Transaction"`
	3. Dynamically creating object instances for nested objects.
	Example:
	`w_main'cb_1 lcb`
	`lcb = Create using "w_main'cb_ 1`

DESTROY	DESTROY DBTrans Supported:

	1. The Destroy statement in non-visual system objects (DataStore, DynamicStagingArea, and Transaction Object) and non_visual user objects is supported.
	Example:
	`Destroy lnv_string //`
	`lnv_string = create`
	`n_cst_string`
	2. The Destroy statement in visual controls and visual user objects is supported.
	Example:
	`commandbutton lcb_1`
	`lcb_1 = create using "cb_2"`
	`Destroy lcb_1`
	Note: As one of the best development practices, it is strongly recommended that you use DESTROY statement.

DO...LOOP	Four formats of Do...Loop:
	Do...Until
	`DO UNTIL a > 15`
	`   a = (a + 1) * b`
	`LOOP`
	Do...While
	`Integer a = 1, b = 1`
	`DO WHILE a <= 15`
	`   a = (a + 1) * b`
	`LOOP`
	Loop...Until
	`Integer a = 1, b = 1`
	`DO`
	`   a = (a + 1) * b`
	`LOOP UNTIL a > 15`
	Loop...While
	`Integer a = 1, b = 1`
	`DO`
	`   a = (a + 1) * b`
	`LOOP WHILE a <= 15`
	Nesting of Do...Loop statement.
	Example:

```
Int li_array[100,50,200]
FOR i = 1 to 100
        FOR j = 1 to 50
                FOR k = 1 to 200
                ll_array[i,j,k]= i
+ j + k
                NEXT
         NEXT
NEXT
```

Nesting of Do...Loop statement and For...Next statement.

Example:

```
FOR ll_i = 5 to 25
DO UNTIL ll_j > 15
ll_j ++
LOOP
           ll_j = 1
NEXT
```

EXIT

```
DO WHILE a < 10
a ++
IF a > 3 THEN
EXIT
b += a
LOOP
```

FOR…NEXT

```
Integer a=1
Integer start, end, increment
...
For n=start TO end STEP increment
a*=n
Next
```

1. End the FOR loop with the keywords END FOR instead of NEXT.

   Example:

   ```
   FOR ll_i = 5 to 25
   ll_j = ll_j+10
   END FOR
   ```

2. Using a positive or negative variable for the step increment.

   Example:

   ```
   FOR N = 5 TO 25 STEP 5
   A = A+10
   NEXT
   ```

3. Nesting of the For...Next statements or For ... Next statement with Do ... Loop statement.

   Example:

	``
	Int li_array[100,50,200]
	FOR i = 1 to 100
	      FOR j = 1 to 50
	            FOR k = 1 to 200
	            ll_array[i,j,k]= i +
	j + k
	             NEXT
	       NEXT
	NEXT
	``

HALT	```
	IF sle_password.Text <> CorrectPassword
	THEN HALT CLOSE
	``
	Notes:
	1. The code following the Halt statement will not be executed in Appeon Mobile. For example, in the following script, "close(parent)" will be ignored in the Appeon conversion process.
	``
	Halt //supported
	Close(parent) //this will be ignored
	``
	2. The reserved word HALT is supported.
	3. If HALT CLOSE statement is invoked in a nested call, Web application won't be closed immediately at the execution of the statement. Instead Web application will be closed when the nest call is completed executed.

IF...THEN	```
	IF num >= 1 THEN result = 1 ELSE result = 0
	IF num >= 1 then
	result = 1
	ELSEIF num <= -1 then
	result = -1
	ELSE
	result = 0
	END IF
	``

RETURN	``` RETURN 0 ```

Unsupported

The following statements are unsupported:

- GOTO
- THROW
- THROWS
- TRY...CATCH...FINALLY...END TRY

7.3 Using PowerBuilder Source Editor

A script that is manually added in the Edit Source window will not be converted to the Web. For example, the following script that follows "on w_1.create"; in the Edit Source window will have no effect: Messagebox ("Welcome","Welcome to Appeon!").

For a descendant object, it is unsupported to declare the event of its ancestor object in the Edit Source window.

8 Embedded SQL

8.1 Database server and data types

Supported

The supported database servers and data types are listed in the table below:

Table 8.1:

MS SQL Server	**Supported**		
	binary*	bit	bigint
	char	datetime	decimal
	float	int	image*
	money	nchar	numeric
	ntext	nvarchar	real
	small datetime	small money	smallint
	sql_variant	text	tinyint
	timestamp	unique identifier	varbinary*
	varchar		
	You can only use the default value for the user_quoted_identifier property.		
	Unsupported		
	The data types that are marked with a "*" symbol can only be used in SELECTBLOB and UPDATEBLOB SQL statements. binary & varbinary On the Web, the data length of the two types can exceed 255 digits. Digits that exceed 255 can either be read from or updated to the database by SELECTBLOB or UPDATEBLOB statement.		
Sybase ASE Server	**Supported**		
	binary	bit	char
	date	datetime	decimal
	float	Image	int
	money	nchar	nvarchar
	numeric	real	smalldatetime
	smallint	smallmoney	text
	timestamp	tinyint	varbinary
	varchar		
	Unsupported		
	In a column, data of timestamp type cannot be displayed correctly.		

	Timestamp data type is unsupported when dynamically creating DataWindows.		
Sybase ASA/SQL Anywhere Server	**Supported**		
	binary*	bit	bigint
	char	date	decimal
	double	float	integer
	long_binary*	long_varchar	money
	numeric	smallint	smallmoney
	time	timestamp	tinyint
	unsigned_bigint	unsigned_int	unsigned_smallint
	varbinary*	varchar	
	Unsupported		
	The data types that are marked with a "*" symbol can only be used in SELECTBLOB and UPDATEBLOB SQL statements.		
Oracle Server	**Supported**		
	blob*	char	character
	clob*	date	dec
	decimal	float	integer
	interval_day_to_second	interval_year_to_month	long
	nchar	nvarchar2	number
	numeric	raw*	real
	ref cursor	rowid*	timestamp
	timestampwithtimezone	timestampwithlocaltimezone	urowid*
	varchar	varchar2	
	Unsupported		
	The data types that are marked with a "*" symbol can only be used in SELECTBLOB and UPDATEBLOB SQL statements.		
IBM DB2 Server	**Supported**		
	bigint	blob*	char
	clob*	date	dbclob*
	decimal	double	graphic
	integer	real	smallint
	time	timestamp	varchar
	vargraphic		
	Unsupported		
	The data types that are marked with a "*" symbol can only be used in SELECTBLOB and UPDATEBLOB SQL statements.		

	Use Double variables instead of Real variables when obtaining float data from the database.		
Sybase IQ	**Supported**		
	bigint	binary*	blob*
	char	clob	date
	datetime	decimal	float
	hs_blockmapidentity*	hs_vdoidentity*	hs_vdorecid*
	image	int	long_binary*
	long_varchar	money	numeric
	oldbit	real	rowid
	smalldatetime	smallint	sysname
	time	timestamp	text
	tinyint	uniqueidentifier*	uniqueidentifierstr
	unsigned bigint	unsigned int	unsigned smallint
	varbinary*	varchar	xact_id
	xml		
	Unsupported		
	The data types that are marked with a "*" symbol can only be used in SELECTBLOB and UPDATEBLOB SQL statements.		
Informix	**Supported**		
	blob*	boolean	byte*
	char	clob*	date
	datetime	decimal	float
	int8	integer	interval
	lvarchar	money	nchar
	nvarchar	real	serial
	serial8	smallint	text
	time	varchar	
	Unsupported		
	The data types that are marked with a "*" symbol can only be used in SELECTBLOB and UPDATEBLOB SQL statements. (* .NET only) The NumericScale attribute is unsupported.		

8.2 Operators

Supported

The following are the SQL operators that can be used:

Table 8.2:

=	<	>	<=	>=	<>	between	exists
in	like	not between	not exists	not in	not like	is	is not
=all	!=all	<all	>all	<=all	>=all	=any	!=any
<any	>any	<=any	>=any	!=any	|	~	

Note: Concatenation operator (+) for String datatype is supported.

Unsupported

"^" is unsupported.

8.3 Transaction management statements

Supported

Table 8.3:

Statement	Examples
COMMIT	Commit
CONNECT	Connect
DISCONNECT	Disconnect
RollBack	Rollback
Savepoint*	Note: this statement is only supported in Oracle database.

8.4 Non-cursor statements

Requirements

1. For SQL Server database, Appeon Server cannot recognize SQL statements with double quotation marks if the SET QUOTED_IDENTIFIER is OFF.

2. For SQL Server and ASE database with JDBC driver, use "SET NOCOUNT ON" before the calls to SQL statements when you create a trigger object. Otherwise, update to the database on the Web may fail with an error "Failed to update database due to java.sql.SQLException… cannot insert the value NULL…" .

3. In SQL statements, if table names and column names are in double quotation mark, string constant should be quoted as single quotation marks.

Supported

Appeon currently supports six types of non-cursor SQL statements:

Table 8.4:

DELETE	INSERT	SELECT
SELECTBLOB	UPDATE	UPDATEBLOB

The following table provides more information for supported non-cursor SQL statements:

Table 8.5:

Type	Supported	Details

SELECT	Retrieval list (SELECT clause)	The retrieval list can be column names, IDs, or aliases. The column names can be expressed as characters (for example, "empid") or as dot notation (for example, emp.empid). Example:
		`select s_emplid, s_emplname //Column names`
		`select e.emp_lname //Table aliases`
		`select s_emplid + s_emplname as emplinfo`
		`//Column (expression) aliases`
		The retrieval list can be expressions that are functions, sub queries, arithmetic operators or any combination of columns, constants, and expressions. Example:
		`select s_emplid + s_emplname as emplinfo`
		`//Concatenation of strings`
		`select substring(s_emplid + s_emplname,1,3)`
		`as emplinfo,getdate()`
		`//Function expression select f_salary / 12`
		`as f_persalary //Operation expression`
		The retrieval list can be an asterisk that represents all columns in one table. Example:
		select * //Asterisk symbol representing all columns.
		Web Enhancement: It is supported to use variable in the Select statement. For example, the following syntax is supported on the Web:
		`string`
		`ls_colname="emp_name"`
		`string ls_value`
		`select :ls_t_employee where id = 1;colname`
		`into:ls_value from`
		`t_employee where id = 1;`

	FROM clause	The FROM clause can contain a single table (view) or multiple tables (views). The table can be expressed using table name or table alias. Example:
		`from Employee, viewbonus`
		`from Employee a, viewbonus b`
		The tables can be locked or unlocked.

	WHERE clause	The WHERE clause can contain any of the following: • Retrieval parameters • Standard comparison operators (=, >, <, <>, >=, <=) • Standard logical operators (NOT, AND, OR) • Special operators (UNION, BETWEEN, IN, LIKE, IS NULL)

Type	Supported Details	
		• Join conditions • Special characters ('&', '~', '[]', '!~~!', '```', '!~@~~~~!', etc.)
	HAVING clause	Supported
	GROUP BY clause	Supported
	COMPUTED clause	Supported
	FOR UPDATE clause	Supported
	Variables list (INTO statement)	The Variables list can be variable(s) or reference(s) to a control property. Example: *into: ls_emplid into: sle_1.Text*

	Example	`SELECT f1, f2, ..., fn into :v1, :v2, ..., :vn`
		`FROM table`
		`WHERE w1 = :p1 and w2 = :p2.prop and/or ... and/or wn = :pn`

SELECT BLOB	SELECT BLOB INTO clause	**Supported** On the Web, if the result set of SELECTBLOB contains several rows, the first row is returned and the return value of SQLNRows is 1 for every supported DBMS. This is different from PowerBuilder. SelectBlob statement may return "" (empty string) on the Web when it returns NULL in PowerBuilder. The difference is caused by the JDBC driver.
	FROM clause	Supported
	WHERE clause	Supported

	Example	`Blob Emp_id_pic`
		`SELECTBLOB Emp_pic`
		`INTO :Emp_id_pic`
		`FROM Employee`
		`WHERE Employee.Emp_Num = 100`
		`USING Emp_tran ;`
		`p_1.SetPicture(Emp_id_pic)`

INSERT	INSERT INTO clause	The INSERT INTO clause can be either table (view) name(s) or a column list.
	VALUES clause	The VALUES clause can contain any of the following: • All supported data types in allowed scope • Space, special characters ('<>', '!', '@', '#', '$', '%', '^', '&', '*')

Type	Supported Details	
		• Initial value, a single record, multiple records (up to 500)
	Validation	The validation will be automatically done by the system.
	Example	`INSERT INTO table` `VALUES(“v1”, 12.3, :p1,:p2, ..., :pn)`
UPDATE	SET clause	The SET clause can update a single record or multiple records (up to 200). It can contain special characters including '<>', '!', '@', '#', '$', '%', '^', '&', '*'.

	Example	`UPDATE table`
		`SET f1 = :p1, f2 = :p2`
		`WHERE w1 = :p3 and/or … wn = :pn`

UPDATE BLOB	WHERE clause	**Supported** If data is updated by UPDATEBLOB statements without a WHERE clause, on the Web all the data will be updated to the database, and the return value of SQLNRows is equal to the number of rows that you have updated. On the Web, the *BlobVariable* could be NULL.

	Example	`UPDATEBLOB Employee SET`
		`emp_pic = :Emp_id_pic`
		`WHERE Emp_num = 100`
		`USING Emp_tran ;`

DELETE	DELETE FROM clause	The FROM clause can contain a single table (view) or multiple tables (views). The table can be expressed using a table name or table alias. The DELETE statement can delete a single record or multiple records.
	WHERE clause	The WHERE clause can contain any of the following: • Retrieval parameters • Standard comparison operators (=, >, <, <>, >=, <=) • Standard logical operators (NOT, AND, OR) • Special operators (UNION, BETWEEN, IN, LIKE, IS NULL) • Join conditions • Special characters (‘&’,’~’,’[]’, ‘!~~!’, ‘```’, ‘!~@~~~~!’, etc.)

	Example	`DELETE`
		`FROM table`
		`WHERE f1 = '1' and f2`
		`= :p1 ... fn = :pn`

Unsupported

- Updateblob statements cannot update multiple blob columns at one time.

- It is unsupported to parenthesize SELECT statements when using EXCEPT ALL or INTERSECT in SQL statements for DB2 database. For example:

```
(SELECT DEP_ID FROM PUB_T_DEPARTMENT) EXCEPT ALL (SELECT DEP_ID FROM
 PUB_T_EMPLOYEE)
(SELECT DEP_ID FROM PUB_T_DEPARTMENT) INTERSECT (SELECT DEP_ID FROM
 PUB_T_EMPLOYEE) //Unsupported
```

```
SELECT DEP_ID FROM PUB_T_DEPARTMENT EXCEPT ALL SELECT DEP_ID FROM PUB_T_EMPLOYEE
SELECT DEP_ID FROM PUB_T_DEPARTMENT INTERSECT SELECT DEP_ID FROM
 PUB_T_EMPLOYEE //Supported
```

- If the database is Sybase ASE or SQL Server, SQL statements can contain column name(s) enclosed in double quotes (same as in PowerBuilder). If the database is not Sybase ASE or SQL Server, SQL statements cannot contain column name(s) enclosed in double quotes (same as in PowerBuilder).

- Using a COMPUTE BY clause in the SELECT statement is unsupported if an application uses a JDBC driver.

8.5 Cursor statements

Supported

- The following statements for retrieving and updating cursors are supported:

Table 8.6:

CLOSE	DECLARE	DELETE	FETCH
FETCH FIRST	FETCH LAST	FETCH NEXT	FETCH PRIOR
OPEN	UPDATE		

- Local cursors are supported.

- Global and instance cursors are supported.

Unsupported

- The Cursor SQL statement UPDATE Where Current is unsupported.

  Syntax:

```
UPDATE TableName SetStatement WHERE CURRENT OF CursorName;
```

- The Cursor SQL Statement DELETE Where Current is unsupported.

  Syntax:

```
DELETE FROM TableName WHERE CURRENT OF CursorName;
```

- If a cursor is declared for retrieving rows from a table, modifying (inserting, deleting, or updating) the table during the cursor open close period is unsupported . Otherwise, the data retrieved is different on the Web than in the PowerBuilder application. For example:

```
DECLARE cur_empl CURSOR FOR select s_emplid, s_emplname from employee;
OPEN cur_empl;
     INSERT INTO employee (s_emplid, s_emplname) VALUES (:ls_addid, :ls_addname);
     FETCH cur_empl INTO :ls_emplid, :ls_emplname;
     DO WHILE sqlca.sqlcode=0
        FETCH cur_empl INTO :ls_emplid, :ls_emplname;
  LOOP
CLOSE cur_empl;
Commit;
```

- Placing the cursor declaration syntax in a statement block that may not be executed at runtime is unsupported. In PowerBuilder, cursor declaration syntax is treated the same way as variable declaration, so the syntax will not be skipped although the statement block is not executed. However, in the Web application, the syntax may be skipped and cause errors. For example:

```
if li_lenth = 10 then
    DECLARE cur_empl CURSOR FOR select s_emplid, s_emplname from employee;
    ......
End if\nOPEN cur_empl;\nFETCH cur_empl INTO :ls_emplid, :ls_emplname;
......

//In a Web application with the above syntax, if the li_length is not 10, the
 cursor
declaration syntax cannot be read, and errors occur.
```

8.6 Database stored procedures

Table 8.7:

SQL statements	**Supported**	
	Statements	**Examples**
	DECLARE	`DECLARE lproc_1 PROCEDURE FOR StoreProcedure @f1 = :p1 IN, @f2 = :p2 OUT, ... @fn = :pn USING trans_obj;`
	EXECUTE	`EXECUTE lproc_1;`
	FETCH	`FETCH lproc_1 INTO :v1, :v2, :v3, ...;`
	FETCH FIRST	`FETCH FIRST lcur_1 INTO :v1, :v2, :v3, ...;`
	FETCH LAST	`FETCH LAST lcur_1 INTO :v1, :v2, :v3, ...;`
	FETCH NEXT	`FETCH NEXT lcur_1 INTO :v1, :v2, :v3, ...;`
	FETCH PRIOR	`FETCH PRIOR lcur_1 INTO :v1, :v2, :v3, ...;`
	CLOSE	`CLOSE lproc_1;`
	Notes Input & output parameters are supported. Return value for stored procedure is supported.	
Declaration syntax	All supported, except this: Placing the stored procedure declaration syntax in a statement block that may not be executed at runtime is unsupported.	

	In PowerBuilder, stored procedure declaration syntax is treated the same way as variable declaration, so the syntax will not be skipped although the statement block is not executed. However, in the Web application, the syntax may be skipped and cause errors. For example:
	`if li_length = 10 then`
	`  DECLARE proc_empl PROCEDURE FOR dbo.java_debug_request`
	`  debugger = a1,`
	`  request = a2,`
	`  out_request = a3 ;`
	`...`
	`End if`
	`OPEN proc_empl;`
	`FETCH proc_empl INTO :ls_emplid;`
	`...`
	In a Web application with the above syntax, if the li_length is not 10, the cursor declaration syntax cannot be read, and errors occur.

Arguments of stored procedures	**Supported** Input, output & inout parameters is supported. The data type of the parameters should match the corresponding data type in database when you declare an stored procedure. Multiple result sets for a stored procedure.
	Unsupported (for Oracle database) Parameters can not have default values. Parameters can not be expressions.
User-defined data types	**Unsupported** **(.NET only) You can work around for SQL Server database with the following instructions.**
	To use the user-defined data type of SQL Server for stored procedure, you need to take the following steps to modify the configuration file: 1. Open the configuration file named "user2systemdbtype.config" in the <AppeonServerRootDirectory>/appeon/AEM/config directory. 2. Add data type mappings in the section named "sqlserver". For example, if you have created a user-defined data type named "mydatetime" that is derived from the system data type "datetime", you would need to set the value of the "userdefine-type" attribute to "mydatetime" and then set the value of the "system-type" attribute to "datetime" in the section named "datatype". If you have more than one user-defined data type, you should add all of them to the configuration file, as shown in the example below.

	<database name="sqlserver"> <datatype userdefine-type="mydatetime" system-type="datetime" /> <datatype userdefine-type="myvarchar" system-type="varchar" /> </database> 3. Restart IIS after you make changes to the configuration file.
Calling stored procedures	**Supported** Appeon provides nearly full support for calling stored procedures, except for the following requirements and unsupported features.
	Requirements (For Informix database) The sequence of input parameters must stay the same as that in the Informix database. For example: `Create procedure ProcName(Variable1 int,Variable2 varchar(10), ...) return varchar(10) End procedure ProcName Call ProcName(10,"Appeon", ...)`
	Unsupported Appeon does not support using an expression as a parameter for calling to an Oracle stored procedure. Appeon does not support using default values for parameters in the stored procedure. Appeon does not support calling an ASA/SQL Anywhere stored procedure with output arguments. Appeon does not support calling an overloading stored procedure. **(.NET only)** If the parameter is null, dynamically calling stored procedure in Informix is unsupported.

8.7 Dynamic SQL

Supported

- Dynamic SQL Format 1: executing a SQL statement does not produce a result set and does not require input parameters.

  Example code:

```
EXECUTE IMMEDIATE :strSQL USING trans_obj;
/*Executing a SQL statement does not
produce a result set and does not require
input parameters*/
```

  Notes:

  1. In EXECUTE IMMEDIATE SQL statement, if the number of fetched row(s) is 0, the SQLCODE in the transaction object is 0 in PowerBuilder while it is 100 in Appeon.

2. Using the syntax EXECUTE IMMEDIATE "set transaction isolation level n" is unsupported.

3. (.NET* only) Input parameters are unsupported.

- Dynamic SQL Format 2: executing a SQL statement that does not produce a result set but does require input parameters.

  Example code:

```
INT emp_id = 56
String fname = "jack";
PREPARE sqlsa FROM "Delete From employee
Where emp_id=? And fname=?"
EXECUTE sqlsa USING :emp_id, :fname;
/*Executing a SQL statement that does not
produce a result set but does require
input parameters*/
```

- Dynamic SQL Format 3: Use this format to execute a SQL statement that produces a result set in which the input parameters and result set columns are known at compile time. Example code:

```
DECLARE Cursor | Procedure DYNAMIC CURSOR
| PROCEDURE

FOR DynamicStagingArea ;
PREPARE DynamicStagingArea FROM
SQLStatement {USING TransactionObject} ;
OPEN DYNAMIC Cursor {USING
ParameterList} ;
EXECUTE DYNAMIC Procedure {USING
ParameterList} ;
FETCH Cursor | Procedure INTO
HostVariableList ;
CLOSE Cursor | Procedure ;
/*Use this format to execute a SQL
statement that produces a result set in
which the input parameters and result set
columns are known at compile time*/
```

  Note: The default Transaction object name SQLCA is supported.

- Dynamic SQL Format 4: executing a SQL statement that produces a result set in which the number of input parameters, result set columns, or both, are unknown at compile time. Example code:

```
DECLARE Cursor | Procedure DYNAMIC CURSOR
| PROCEDURE
FOR DynamicStagingArea ;
PREPARE DynamicStagingArea FROM SQLStatement
{ USING TransactionObject} ;
DESCRIBE DynamicStagingArea INTO
DynamicDescriptionArea ;
OPEN DYNAMIC Cursor | Procedure USING
DESCRIPTOR DynamicDescriptionArea ;
EXECUTE DYNAMIC Cursor | Procedure USING
DESCRIPTOR DynamicDescriptionArea ;
FETCH Cursor | Procedure USING DESCRIPTOR
DynamicDescriptionArea ;
CLOSE Cursor | Procedure ;
/ *Use this format to execute a SQL statement
that produces a result set in which the input
parameters and result set columns are unknown
at compile time*/
```

- Notes:

  1. The default transaction object name SQLDA is supported.

  2. If you need a DynamicStagingArea variable other than SQLSA, you must declare it and instantiate it with the CREATE statement before using it.

  3. (.NET only) Because of the .NET driver for Informix, Web application differs from PowerBuilder application in the following aspect:

     a. The Time data type will be returned as DateTime on the Web.

     b. The Money and Float data type will be returned as Decimal data type.

  4. In Appeon for PowerBuilder, when executing a dynamic embedded SQL in Dynamic SQL Format 4 (example code as shown below), ASE database cannot get the output parameters from the stored procedure.

     This problem is caused by ASE database driver. If there are similar codes described below causing problems in ASE database, please turn to relevant technicians of Sybase for support.

```
String ls_execute_sql = 'begin
transaction apb ' + '~r~n' +&
'declare @myparm numeric(4) ' + '~r~n' +&
'exec appeon_inout @parin = 4, @parout =
@myparm output' + '~r~n' +&
'SELECT @myparm ' + '~r~n' +&
'commit transaction apb '
prepare sqlsa from :ls_execute_sql using
sqlca;
describe sqlsa into sqlda;
DECLARE my_cursor DYNAMIC procedure FOR
SQLSA ;
execute DYNAMIC my_cursor using
descriptor sqlda ;
if sqlca.sqlcode < 0 then
messagebox('1',sqlca.sqlerrtext)
end if
FETCH my_cursor using descriptor sqlda ;
if sqlca.sqlcode < 0 then
messagebox('2',sqlca.sqlerrtext)
else
ls_Value = String(SQLDA.GetDynamicNumber(1))
If IsNull(ls_Value) Then
ls_value = 'Null'
ElseIf ls_value = '' then
ls_value = "Empty string('')"
End If
messagebox('3',ls_Value)
end if
close my_cursor;
```

Difference

- Calling Fetch Prior when the cursor position is on the first row or calling Fetch Next when the cursor position is on the last row returns a different sqldbcode than in PowerBuilder and on the Web.
- It is suggested to use "colname is null" instead of "colname = null".

9 Functions

9.1 System functions

Supported

The following are the categories with some or all of their functions supported:

Table 9.1:

Array Functions	Blob Function	Data Type Checking and Conversion Functions
Date, Day, and Time Functions	DDE Client Functions	File Functions
International Functions	Miscellaneous Functions	Numeric Functions
Print and Printer Functions	Registry Functions	String Functions
System and Environment Functions	Timing Functions	Window Functions

Unsupported

- The following are the unsupported system function categories that do not contain any supported individual functions:

  Table 9.2:

Class Definition Functions	DDE Server Functions	Garbage Collection Functions
Help Functions	Library Functions	Shared Object Functions (SharedObject Functions)
Tracing Functions		

- Having a parent object dynamically call a system function that is overloaded by its child object is unsupported.

9.1.1 Supported types

9.1.1.1 Array functions

Table 9.3:

Function	Support Level	Coding Examples
LowerBound	Supported	Int b[-5 to 2,5],a[5],c[] li_lower = LowerBound(b,2) li_lower = LowerBound(b,1) li_lower = LowerBound(a) li_lower = LowerBound(a,1) li_lower = LowerBound(c)
UpperBound	Supported	Int b[-15 to -5,5],a[5],c[]

Function	Support Level	Coding Examples
		li_upper = UpperBound(b,2) li_upper = UpperBound(b,1) li_upper = UpperBound(a) li_upper = UpperBound(a,1) li_upper = UpperBound(c)

9.1.1.2 Blob functions

Table 9.4:

Function	Support Level	Coding Examples
Blob	Supported	blob lb_data = blob("convert string to blob") string ls_data lb_data = blob(ls_data)
BlobEdit	Supported	ulong ll_var blob lb_data ll_var = BlobEdit(lb_data, 1, "String") ll_var = BlobEdit(lb_data, 3, 12345) ll_var = BlobEdit(lb_data, 9, 2004-12-03) ll_var = BlobEdit(lb_data, 1, blob("blob")) Note: The *data* argument cannot be Any data type.
BlobMid	Supported	blob lb_data,lb_subdata lb_subdata = BlobMid(lb_data,3) lb_subdata = BlobMid(lb_data,5)
Len	Supported	long ll_len blob lb_data,lb_subdata ll_len = len(lb_data) ll_len = len(lb_subdata)
Lenw	Supported	long ll_len blob lb_data,lb_subdata ll_len = LenW(lb_data) ll_len = LenW(lb_subdata)
String	Supported	blob lb_data,lb_subdata string ls_data ls_data = string(lb_data) ls_data = string(lb_subdata)

9.1.1.3 Byte functions

Table 9.5:

Function	Support Level	Coding Examples
Byte	Supported	
GetByte	Supported	
SetByte	Supported	
GetByteArray	Supported	

9.1.1.4 Data type checking and conversion functions

Table 9.6:

Function	Support Level	Coding Examples
Asc	Supported	li_test = Asc('adfsd') li_test = Asc('~nern') li_test = Asc(ls_test) li_test = Asc(wf_getstring()) //wf_getstring() is a function that has a string return value
Char	Supported	ls_test1 = Char(65) ls_test1 = Char(65) ls_test1 = Char("~n~~djfksdl")

Dec	Supported	ldec_test = Dec("1.234567")
		//Return the string as a decimal
		ldec_test = Dec("1234567")
		ldec_test = Dec(dw_1.object.data[1,2])

Double	Supported	ldb_test = Double("78.7956") //Return the string as a double
Integer	Supported	li_test = Integer("93") //Return the string as an integer

Long	Supported	ll_test = long("99.88")
		//Return the string as a long
		ll_test = long (16119,26930)
		//Convert the two UnsignedIntegers into a long

Real	Supported	lr_test = Real ("88.56") //Return the string as a real
Date	Supported	ld_test = Date(ldt_test) //ldt-test is a datetime variable

		ld_test = Date(now())
		ld_test = Date(ls_test)
		//ls_test is a string variable
		ld_test = Date (2003, 4, 1)
		Note: if the argument contains an invalid date, Date returns NULL in PowerBuilder, whereas in the Web application, it returns an empty string (“”).

DateTime	Supported	ldt_test = Datetime(ld_test) ldt_test = Datetime(ld_test, lt_test) Note: After conversion, the microsecond portion in the time argument will be omitted.
IsDate	Supported	If IsDate("Jan 32, 1993") = TRUE Then... //Determines whether the string is a valid date
IsNull	Supported	Integer a, b... lb_value = isnull(a+b) //If the value of expression a+b is null, the lb_value is set as True; otherwise, False
IsNumber	Supported	If IsNumber("23.45") Then ... //Return True since the value of the string is a number Note: PowerBuilder does not support 1E123 (more than two digits after E). Functions such as IsNumber ("1E123") in PowerBuilder return FALSE, but after conversion, IsNumber returns TRUE.

IsTime	Supported	IsTime(timevalue)
		If IsTime("23: 11") Then ...
		//hh:mmIf
		IsTime("23: 11:33") Then ...
		//hh:mm:ssIf IsTime("23: 11:33.123456") Then ...
		//hh:mm:ss.xxxxxx

String	Supported	String(data{, format})
		ls_test = string(1993-05-17, "mm/dd/yyyy")
		//Convert a date to string
		ls_test = string(07:12:28,"hh:mm:ss")
		//Convert a time to string
		ls_test = string(44.56, "$#,##0.00")
		//Convert a numeric to string
		ls_test = string("gf", "@*@")

		//Convert a string to formatted string Notes: 1. String(data, format) cannot be used in DataWindow property expression. 2. When a format is specified in the String function, make sure that the format is one of the supported types for EditMask. The string argument cannot contain special case conversion formats such as "!!!", "aaa" 3. The format argument cannot be [currency].
Time	Supported	lt_test = time("23:00") Note: After conversion, the microsecond portion of the time will be omitted.
LongLong	Unsupported	

9.1.1.5 Date, Day and Time functions

Table 9.7:

Function	Support Level	Coding Examples
Day	Supported	li_test = Day(2003-04-01) li_test = Day(ld_today) li_test = Day(today())
DayName	Supported	ls_test = DayName(2003-04-01) ls_test = DayName(ld_today) ls_test = DayName(today())
DayNumber	Supported	li_test = DayNumber(2003-04-01) li_test = DayNumber (ld_today) li_test = DayNumber (today())
DaysAfter	Supported	ll_test = DaysAfter(2003-04-01, 2003-04-01) ll_test = DaysAfter(ld_test1, today())
Hour	Supported	Hour(*time*) li_test = Hour(21:00:00) li_test = Hour(lt_test) li_test = Hour(Now())
Minute	Supported	li_test = Minute (21:00:00) li_test = Minute (lt_test) li_test = Minute (Now())
Month	Supported	li_test = Month (2003-04-01)

Function	Support Level	Coding Examples
		li_test = Month (ld_today) li_test = Month (today())
Now	Supported	ldt_test = Now()
RelativeDate	Supported	ld_test = RelativeDate (2003-04-01, 27) ld_test = RelativeDate (ld_test1, li_after) ld_test = RelativeDate (today(), li_after)
RelativeTime	Supported	lt_test = RelativeTime (21:00:00, 60) lt_test = RelativeTime (ld_test1, li_after) lt_test = RelativeTime (now(), li_after)
Second	Supported	li_test = Second (21:00:00) li_test = Second (lt_test) li_test = Second (Now())
SecondsAfter	Supported	ll_test = SecondsAfter (21:00:00, 09:00:00) ll_test = SecondsAfter (lt_test1, now())
Today	Supported	ld_test = Today()
Year	Supported	li_test = Year (2003-04-01) li_test = Year (ld_today) li_test = Year (today())

9.1.1.6 DDE client functions and events

Table 9.8:

Function	Support Level	Coding Examples
CloseChannel	Supported	CloseChannel(handle, Handle(w_sheet))
ExecRemote	Supported	
GetDataDDE	Supported	string Str20 GetDataDDE(Str20)
GetDataDDEOrigin	Supported	string WhichAppl, WhatTopic, WhatLoc GetDataDDEOrigin(WhichAppl, WhatTopic, WhatLoc)
GetRemote	Supported	
OpenChannel	Supported	long handle handle = OpenChannel("Excel", "REGION.XLS") Note: It is recommended to determine whether the execution succeeds by its returning -1, -9 or a positive integer. Do not assume that it has failed because it returns a negative integer.

Function	Support Level	Coding Examples
RespondRemote	Supported	IF GetDataDDE(Value) = 1 THEN RespondRemote(TRUE)
SetRemote	Supported	
StartHotLink	Supported	StartHotLink("Any", "MyPBApp", "Any")
StopHotLink	Supported	StopHotLink("Any", "MyPBApp", "Any")
HotLinkAlarm	Unsupported	

9.1.1.7 File functions

Important Requirements

On the Web, the reference of the relative path is the Desktop directory, however in PowerBuilder, the reference of relative path is the target application's current directory.

Table 9.9:

Function	Support Level	Coding Examples
ChangeDirectory	Supported	int li_return li_return = ChangeDirectory(sle_path.text)
CreateDirectory	Supported	int li_return li_return = CreateDirectory(sle_path.text)
DirectoryExists	Supported	if DirectoryExists(sle_path.text) then ls_tosting = "The directory exists!"
FileClose	Supported	li_return = FileClose(li_FileNum)
FileCopy	Supported	li_return = FileCopy(li_FileNum)
FileDelete	Supported	if FileDelete(sle_filename.text) then ls_tosting = "True:FileDelete"+sle_filename.text
FileEncoding	Unsupported	
FileExists	Supported	if FileExists(sle_filename.text) then ls_tosting = "The file exists!"
FileOpen	Supported	integer li_FileNum li_FileNum = FileOpen(“D:\temp\new.txt”, TextMode!, Write!,LockReadWrite!, Replace!, EncodingUTF8!)
FileLength	Supported	long ll_FileLen ll_FileLen = FileLength(sle_filename.text)
FileLength64	Unsupported	
FileMove	Supported	li_return = FileMove(sle_filename.text,sle_2.text)
FileRead	Supported	blob Emp_Id_Pic li_bytes = FileRead(li_FileNum, Emp_Id_Pic)
FileReadEx	Supported	blob Emp_Id_Pic

Function	Support Level	Coding Examples
		li_bytes = FileReadEx(li_FileNum, Emp_Id_Pic)
FileSeek	Supported	ll_FileLen = FileSeek(li_FileNum, 14, FromCurrent!) FileSeek(li_FileNum, -14, FromEnd!)
FileSeek64	Unsupported	
FileWrite	Supported	blob Emp_Id_Pic FileWrite(li_FileNum, "New Employees") FileWrite(li_FileNum, emp_id_pic)
FileWriteEx	Supported	blob Emp_Id_Pic FileWriteEx(li_FileNum, "New Employees") FileWriteEx(li_FileNum, emp_id_pic)
GetCurrent Directory	Supported	sle_1.text = GetCurrentDirectory() Note: GetCurrentDirectory returns full path name of Desktop on the Web, and it returns the full path name of the current directory in PowerBuilder.
GetFileOpenName	Supported	li_return = GetFileOpenName(sle_po.text, sle_path.text, ls_file[], "TXT", "Text Files (*.TXT),*.TXT, Doc Files (*.DOC),*.DOC") li_return = GetFileOpenName(sle_po.text, sle_path.text, sle_filename.text, "TXT", "Text Files (*.txt),*.txt" , sle_2.text)
GetFileSaveName	Supported	li_return = GetFileSaveName(sle_po.text, sle_path.text, ls_file[], "TXT", "Text Files(*.TXT),*.TXT, Doc Files (*.DOC),*.DOC") li_return = GetFileSaveName(sle_po.text, sle_path.text, sle_filename.text, "TXT", "Text Files (*.txt),*.txt", sle_2.text)
GetFolder	Supported	li_return = GetFolder("Folder", sle_path.text)
RemoveDirectory	Supported	li_return = RemoveDirectory(sle_path.text)

9.1.1.8 International functions

Table 9.10:

Function	Support Level	Coding Examples
Reverse	Supported	ls_return = Reverse(ls_test)
IsAllArabic	Unsupported	
IsAllHebrew	Unsupported	
IsAnyArabic	Unsupported	
IsAnyHebrew	Unsupported	
IsArabic	Unsupported	

Function	Support Level	Coding Examples
IsArabic AndNumbers	Unsupported	
IsHebrew	Unsupported	
IsHebrew AndNumbers	Unsupported	
FromAnsi	Unsupported	
FromUnicode	Unsupported	
ToAnsi	Unsupported	
ToUnicode	Unsupported	

9.1.1.9 Miscellaneous functions

Table 9.11:

Function	Support Level	Coding Examples
Beep	Supported	Beep(5)
ChooseColor	Supported	li_color = ChooseColor(red, custom [ ])
ClassName	Supported	ls_classname = ClassName(li_array) Note: If the argument is a numeric data type, the function returns number.
DebugBreak	Supported	IF IsNull(auo_ext) THEN DebugBreak()
DraggedObject	Supported	DragObject poj_ctr poj_ctr = DraggedObject()
GetFolder	Supported	integer li_result li_result = GetFolder("my targets", ls_path)
IntHigh	Supported	li_high = IntHigh(ll_value)
IntLow	Supported	li_low = IntLow(ll_value)
IsValid	Supported	IF IsValid(w_emp) = FALSE THEN Open(w_emp)
KeyDown	Supported	KeyDown (keycode) Notes: 1. In the Appeon deployed application, the function returns TRUE only when the key is being pressed and FALSE if not. However, the KeyDown function returns TRUE once the key is pressed and FALSE if not. 2. The following enumerated Keycode value are unsupported: KeyApps, KeyLeftWindows, KeyNull, KeyPrintScreen, KeyRightWindows.

Function	Support Level	Coding Examples
MessageBox	Supported	li_return = MessageBox('Title1','Text1',Information!,OK!,1)
PixelsToUnits	Supported	li_return = PixelsToUnits(35, XPixelsToUnits!)
RGB	Supported	ll_color = RGB(255, 255, 255) The RGB value scope supported: 0~16777215 The custom color scope supported: 16777216~33554431 Note: If the color value is -2 or -1 (Transparent), the color display effect will be different in the Web from in the PowerBuilder application. If it is a color unsupported (the color value is less than -2), the color will be replaced with ButtonFace color.
SetNull	Supported	SetNull(ls_test)
SetPointer	Supported	SetPointer(Cross!)
Sleep	Supported	Sleep(5)
UnitsToPixels	Supported	li_return = UnitsToPixels(350, YUnitsToPixels!)
PopulateError	Unsupported	
SignalError	Unsupported	

9.1.1.10 Numeric functions

Table 9.12:

Function	Support Level	Coding Examples
Abs	Supported	ldec_return = Abs(-15725.12) ldec_return = Abs(ai_num)
ASin	Supported	ldb_ return = ASin(.84147) ldb_ return = ASin(af_num)
ACos	Supported	ldb_ return = ACos(.84147) ldb_ return = ACos(af_num)
ATan	Supported	ldb_ return =Atan(.84147) ldb_ return = ATan(af_num)
Ceiling	Supported	li_ return = Ceiling(3558.5) li_ return = Ceiling(af_num)
Cos	Supported	ldb_ return = Cos(10586.3) ldb_ return = Cos(af_num)
Exp	Supported	ldb_ return = Exp(17438.15) ldb_ return = Exp(af_num)
Fact	Supported	ldb_ return = Fact(14)

Function	Support Level	Coding Examples
		ldb_ return = Fact(af_num)
Int	Supported	li_ return = Int(8314.11) li_ return = Int(af_num)
Log	Supported	ldb_ return = Log(7628) ldb_ return = Log(af_num)
LogTen	Supported	ldb_ return = LogTen(30975.5) ldb_ return = LogTen(af_num)
Max	Supported	ldec_ return = Max(1019,21120) ldec_ return = Max(af_num,bf_num)
Min	Supported	ldec_ return = Min(1019,21120) ldec_ return = Min(af_num,bf_num)
Mod	Supported	ldec_ return = Mod(32526,8261.15) ldec_ return = Mod(af_num,bf_num)
Pi	Supported	ldb_ return = Pi(20852) ldb_ return = Pi(af_num)
Rand	Supported	ldec_ return = Rand(14888) ldec_ return = Rand(af_num)
Randomize	Supported	Randomize(0)
Round	Supported	ldec_ return = Round(6655.16973,3) ldec_ return = Round(af_num,b_num) Executing Round in SQL statements, decimals will be ignored on the Web.
Sign	Supported	li_ return = Sign(0) li_ return = Sign(-543534) li_ return = Sign(4563) li_ return = Sign(af_num)
Sin	Supported	ldb_ return = Sin(-751) ldb_ return = Sin(751) ldb_ return = Sin(af_num)
Sqrt	Supported	ldb_ return = Sqrt(740752012) ldb_ return = Sqrt(af_num)
Tan	Supported	ldb_ return = Tan(28713.4) ldb_ return = Tan(af_num)
Truncate	Supported	ldec_ return = Truncate(21133.24473,3)

Function	Support Level	Coding Examples
		ldeci_ return = Truncate(af_num)

9.1.1.11 Print functions

Table 9.13:

Function	Support Level	Coding Examples
Print	Supported	Print(Job, "Sybase Corporation", 5000) Note: You can call Print in DataWindow to workaround the PrintDataWindow function.
PrintBitmap	Supported	PrintBitmap(li_job, "d:\PB\BITMAP1.BMP", &50,100, 0,0)
PrintCancel	Supported	PrintCancel(li_job)
PrintClose	Supported	PrintClose(li_job)
PrintDataWindow	Supported	PrintDataWindow(job, dw_EmpHeader)
PrintDefineFont	Supported	PrintDefineFont(li_job, 1, "Courier 10Cpi", -18, & 400, Default!, Decorative!, FALSE, FALSE) Note: Using user-defined fonts that cannot be supported by the system is unsupported.
PrintLine	Supported	PrintLine(li_job,0,5,7500,5,10)
PrintOpen	Supported	Ulong li_job = PrintOpen("Phone List")
PrintOval	Supported	PrintOval(li_job, 4000, 3000, 1000, 1000, 10)
PrintPage	Supported	PrintPage(li_job)
PrintRect	Supported	PrintRect(li_job, 4000,3000, 1000,1000, 125)
PrintRoundRect	Supported	PrintRoundRect(li_job, 1000,1000, 6250,9000, &300,300, 125)
PrintScreen	Supported	PrintScreen(li_job,500,1000,5000,5000)
PrintSetFont	Supported	PrintSetFont(li_job, 2)
PrintSetSpacing	Supported	PrintSetSpacing(li_job, 1.5)
PrintSetup	Supported	PrintSetup()
PrintText	Supported	PrintText(li_job,"PowerBuilder", 3700, 10,3)
PrintWidth	Supported	Int W = PrintWidth(li_job,"PowerBuilder")
PrintX	Supported	Int LocX = PrintX(li_job)
PrintY	Supported	Int LocY = PrintY(li_job)
PrintSend	Unsupported	Note: This function is obsolete for PowerBuilder, so Appeon does not support it either

9.1.1.12 Printer functions

Table 9.14:

Function	Support Level	Coding Examples
PrintGetPrinter	Supported	String ls_fullstring=PrintGetPrinter( )
PrintGetPrinters	Supported	String ls_prntrs = PrintGetPrinters( )
PrintSetPrinter	Supported	String ls_setprn PrintSetPrinter (ls_setprn)
PrintSetupPrinter	Supported	PrintSetupPrinter ()

9.1.1.13 Registry functions

Table 9.15:

Function	Support Level	Coding Examples
RegistryDelete	Supported	RegistryDelete("HKEY_LOCAL_MACHINE \Software\MyApp.Settings\Fonts","Title")
RegistryGet	Supported	RegistryGet("HKEY_USERS\MyApp.Settings\Fonts", "NameOfEntryNum", RegULong!, ul_num) Note: The ValueType argument: • can be RegString!, RegExpandString!, ReguLong!, ReguLongBigEndian!, or RegMultiString!. • cannot be RegMultiString! when implementing this function on the client. • cannot be RegBinary!, RegMultiString! or RegLink! when implementing this function on the server.
RegistryKeys	Supported	li_rtn = RegistryKeys("HKEY_CLASSES_ROOT \MyApp", & ls_subkeylist)
RegistrySet	Supported	RegistrySet("HKEY_USERS\MyApp.Settings\Fonts", "NameOfEntryNum", RegULong!, ul_num) Notes: 1. The ValueType argument: • can be RegString!, RegExpandString!, ReguLong!, ReguLongBigEndian!, or RegMultiString!. • cannot be RegMultiString! when implementing this function on the client. • cannot be RegBinary!, RegMultiString! or RegLink! when implementing this function on the server.

Function	Support Level	Coding Examples
		2. The ValueType argument must be used in the syntax.
RegistryValues	Supported	string ls_valuearray[] RegistryValues("HKEY_LOCAL_MACHINE \Software\MyApp.Settings\Fonts", ls_valuearray)

When setting the value for a key and value name in the system registry, the value name will be converted into lowercase. Therefore, the functions are case insensitive.

Naming rules for the files, sections, registry keys used in the Registry and Profile functions:

- Must start with an ‘_' (underscore), a '$' (dollar sign), or a letter.
- Can include any combination of characters, numbers, '.' (point), '_' (underscore), '-' (dash), space, '\' (backslash) and '$' (dollar sign).
- Cannot include single quotation marks, quotation marks, '&' (ampersand sign), or '/' (slash).

Functionality difference in different Registry Mode:

If the Registry Mode in AEM is set to "Use client machine Windows registry", the registry functions are supported the same way as in PowerBuilder.

If the Registry Mode in AEM is set to "Use Appeon registry emulation", Web applications cannot use RegistryGet and RegistryValues for accessing the Windows registry directly. The Registry functions are implemented as workarounds. They do not read from the actual Windows registry. Instead, Appeon creates a mock registry in the Appeon Server database, which initially has no values. Therefore, you must first set a value using the RegistrySet function before trying to use RegistryGet and RegistryValues functions. In this case the registry functions are only supported to work with existing registry keys, and it is not supported to create new keys during application runtime.

9.1.1.14 String functions

9.1.1.14.1 String functions for PowerBuilder 9

Table 9.16:

Functions	Support Level	Coding Examples
Asc	Supported	li_test = Asc('adfsd') li_test = Asc('~nern') li_test = Asc(ls_test) li_test = Asc(wf_getstring())
Char	Supported	ls_test1 = Char(65) ls_test1 = Char("~n~~djfksdl")
Fill	Supported	ls_fill = Fill('-+',10) ls_fill = Fill('hello', 6)

Functions	Support Level	Coding Examples
FillW	Supported	ls_fill = FillW('-+',10) ls_fill = FillW('hello', 6)
LastPos	Supported	ll_return = LastPos('Appeon Appeon', 'Ap') ll_return = LastPos('Appeon Appeon', 'Ap',4)
Left	Supported	ls_return = string(Left('z{uDPk7#k',9))
LeftW	Supported	ls_return = string(LeftW('z{uDPk7#k',9))
LeftTrim	Supported	ls_return = string(LeftTrim(' fdfsdf')) ls_return = string(LeftTrim(' 34ttrtertgre'))
LeftTrimW	Supported	ls_return = string(LeftTrimW(' fdfsdf')) ls_return = string(LeftTrimW(' 34ttrtertgre'))
Len	Supported	ls_return = string(Len('gfgdfgsdrgdfg'))
LenW	Supported	ls_return = string(LenW('gfgdfgsdrgdfg'))
Lower	Supported	ls_return = Lower(“You ARE Welcome!”)
Match	Supported	ls_return = string(Match('ABBBC,ABB*C'))
MatchW	Supported	ls_return = string(MatchW('ABBBC,ABB*C'))
Mid	Supported	ls_return = string(Mid('Appeon Appeon',5,2))
MidW	Supported	ls_return = string(MidW('Appeon Appeon',5,2))
Pos	Supported	ls_return = string(Pos('Appeon Appeon','on')) ls_return = string(Pos('Appeon Appeon','peon',2))
PosW	Supported	ls_return = string(PosW('Appeon Appeon','on')) ls_return = string(PosW('Appeon Appeon','peon',2))
Replace	Supported	ls_return = string(Replace('BABE RUTH', 1, 4, 'BABY'))
ReplaceW	Supported	ls_return = string(ReplaceW('BABE RUTH', 1, 4, 'BABY'))
Reverse	Supported	ls_return = string(Reverse('vDl%Qv'))
Right	Supported	ls_return = string(Right('Davis', 4))
RightW	Supported	ls_return = string(RightW('Davis', 4))
RightTrim	Supported	ls_return = string(RightTrim('fsdjd')) ls_return = string(RightTrim('fsdfjdlfsd'))
RightTrimW	Supported	ls_return = string(RightTrimW('fsdjd')) ls_return = string(RightTrimW('fsdfjdlfsd'))
Space	Supported	ls_return = string(Space(14))
Trim	Supported	ls_return = string(Trim('uifusd'))
TrimW	Supported	ls_return = string(TrimW('uifusd'))

Functions	Support Level	Coding Examples
WordCap	Supported	ls_return = string(WordCap('how do you do?'))
Upper	Supported	ls_return = string(Upper('how do you do?'))

9.1.1.14.2 String functions for PowerBuilder 10 or above

Table 9.17:

Functions	Support Level	Coding Examples
AscA	Supported	li_test = AscA('adfsd') li_test = AscA(ls_test) Note: The function returns different value on the Web from that in PowerBuilder if the character encoding is non-SJIS.
Asc	Supported	li_test = Asc('~nern') li_test = Asc(wf_getstring())
CharA	Supported	ls_test = CharA(65) ls_test = CharA("~n~~djfksdl") Note: The function returns different value on the Web from that in PowerBuilder if the character encoding is non-SJIS.
Char	Supported	ls_test = Char(65) ls_test = Char("~n~~djfksdl")
Fill	Supported	ls_fill = Fill('-+',10) ls_fill = Fill('hello', 6)
FillA	Supported	ls_fill = FillA('-+',10) ls_fill = FillA('hello', 6) Note: The function returns different value on the Web from that in PowerBuilder if the character encoding is non-SJIS.
FillW	Supported	ls_fill = FillW('-+',10) ls_fill = FillW('hello', 6)
LastPos	Supported	ll_return = LastPos('Appeon Appeon', 'Ap') ll_return = LastPos('Appeon Appeon', 'Ap',4)
Left	Supported	ls_return = string(Left('z{uDPk7#k',9))
LeftA	Supported	ls_return = string(LeftA('z{uDPk7#k',9)) Note: The function returns different value on the Web from that in PowerBuilder if the character encoding is non-SJIS.
LeftW	Supported	ls_return = string(LeftW('z{uDPk7#k',9))

Functions	Support Level	Coding Examples
LeftTrim	Supported	ls_return = string(LeftTrim(' fdfsdf')) ls_return = string(LeftTrim(' 34ttrtertgre'))
LeftTrimW	Supported	ls_return = string(LeftTrimW(' fdfsdf')) ls_return = string(LeftTrimW(' 34ttrtertgre'))
Len	Supported	ls_return = string(Len('gfgdfgsdrgdfg'))
LenA	Supported	ls_return = string(LenA('gfgdfgsdrgdfg')) Note: The function returns different value on the Web from that in PowerBuilder if the character encoding is non-SJIS.
LenW	Supported	ls_return = string(LenW('gfgdfgsdrgdfg'))
Lower	Supported	ls_return = Lower(“You ARE Welcome!”)
Match	Supported	ls_return = string(Match('ABBBC,ABB*C'))
MatchW	Supported	ls_return = string(MatchW('ABBBC,ABB*C'))
Mid	Supported	ls_return = string(Mid('Appeon Appeon',5,2))
MidA	Supported	ls_return = string(MidA('Appeon Appeon',5,2)) Note: The function returns different value on the Web from that in PowerBuilder if the character encoding is non-SJIS.
MidW	Supported	ls_return = string(MidW('Appeon Appeon',5,2))
Pos	Supported	ls_return = string(Pos('Appeon Appeon','on')) ls_return = string(Pos('Appeon Appeon','peon',2))
PosA	Supported	ls_return = string(PosA('Appeon Appeon','on')) ls_return = string(PosA('Appeon Appeon','peon',2)) Note: The function returns different value on the Web from that in PowerBuilder if the character encoding is non-SJIS.
PosW	Supported	ls_return = string(PosW('Appeon Appeon','on')) ls_return = string(PosW('Appeon Appeon','peon',2))
Replace	Supported	ls_return = string(Replace('BABE RUTH', 1, 4, 'BABY'))
ReplaceA	Supported	ls_return = string(ReplaceA('BABE RUTH', 1, 4, 'BABY')) Note: The function returns different value on the Web from that in PowerBuilder if the character encoding is non-SJIS.
ReplaceW	Supported	ls_return = string(ReplaceW('BABE RUTH', 1, 4, 'BABY'))

Functions	Support Level	Coding Examples
Reverse	Supported	ls_return = string(Reverse('vDl%Qv'))
Right	Supported	ls_return = string(Right('Davis', 4))
RightA	Supported	ls_return = string(RightA('Davis', 4)) Note: The function returns different value on the Web from that in PowerBuilder if the character encoding is non-SJIS.
RightW	Supported	ls_return = string(RightW('Davis', 4))
RightTrim	Supported	ls_return = string(RightTrim('fsdjd')) ls_return = string(RightTrim('fsdfjdlfsd'))
RightTrimW	Supported	ls_return = string(RightTrimW('fsdjd')) ls_return = string(RightTrimW('fsdfjdlfsd'))
Space	Supported	ls_return = string(Space(14))
Trim	Supported	ls_return = string(Trim('uifusd'))
TrimW	Supported	ls_return = string(TrimW('uifusd'))
WordCap	Supported	ls_return = string(WordCap('how do you do?'))
Upper	Supported	ls_return = string(Upper('how do you do?'))

9.1.1.15 System and environment functions

Table 9.18:

Function	Support Level	Coding Examples
Clipboard	Supported	string ls_CoName ls_CoName = Clipboard()
CommandParm	Supported	String ls_command_line ls_command_line = CommandParm()
GetEnvironment	Supported	environment env integer rtn rtn = GetEnvironment(env)
GetFocus	Supported	Lobj_control = GetFocus()
GetApplication	Supported	application app app = GetApplication()
Handle	Supported	Note: the Handle property is partially supported • Supported syntax: Handle (objectname) • Unsupported syntax: Handle (objectname, previous)
Post	Supported	Post(Handle(w_date), 277, 3, 0)

Function	Support Level	Coding Examples
		Note: the *message#* argument cannot be a user defined message number.
ProfileInt	Supported	ProfileInt ("C:\PROFILE.INI", "PB", "maximized", 3)
ProfileString	Supported	ProfileString ("C:\PROFILE.INI", "Employee","Name", "None")
Run	Supported	run("notepad") run("notepad.exe") run("C:\winnt\system32\notepad") run("C:\winnt\system32\notepad.exe")
Send	Supported	Send(Handle(w_emp), 277, 2, 0) Note: the *message#* argument cannot be a user defined message number.
SetProfileString	Supported	SetProfileString("C:\PROFILE.INI", "Position", "Title", "MGR")
Yield	Supported	Yield()
Restart	Unsupported	
SignalError	Unsupported	

Naming rules for the files, sections, registry keys used in the Registry and Profile functions:

- Must start with an '_' (underscore) or an '$' (dollar sign) or a letter.
- Can include any combination of characters, numbers, '.' (point), '_' (underscore), '-' (dash), space, '\' (backslash) and '$' (dollar sign).
- Cannot include single quotation mark, quotation mark, '&' (ampersand sign), or '/' (slash).
- Files with the same name cannot be used in Profile functions.

9.1.1.16 Timing functions

Table 9.19:

Function	Support Level	Coding Examples
CPU	Supported	long ll_start ll_start = Cpu()
Idle	Supported	Idle(300)
Timer	Supported	Timer(10, w_main)
Start	Unsupported	

9.1.1.17 Window functions

Table 9.20:

Function	Support Level	Coding Examples
Close	Supported	close (w_commandbutton) close (parent)
CloseWithReturn	Supported	CloseWithReturn (parent, "return value") Notes: 1. In PowerBuilder, the return value is Null when there are two Nulls in the parameter. In this case, the return value in JavaScript is not Null. 2. On the Web, if the *returnvalue* parameter is NULL, the return value will be stored in the PowerObjectParm property of the Message object.
Open	Supported	Open (w_main, parent)

OpenSheet	Supported	OpenSheet (w_main, w_parent)
		OpenSheet (lw_main, "w_main", w_frame , 2, Cascaded!)
		Notes:
		1. When opening the same local window variables more than once within the same function, the result is unpredictable. Avoid using the OpenSheet window function in the following manner:
		`w_11 ww opensheet (ww , "w_111", w_2 , 2, Cascaded!) opensheet(ww, "w_11", w_2, 0, Cascaded!)`
		2. The following syntax is supported:
		`OpenSheet (sheetrefvar{, windowtype}, mdiframe{, position {, arrangeopen}})`
		3. Appeon supports appending a window list to a menu item on the Web.
		4. The ArrangeOpen argument can be Cascaded!, Layered!, Original!. For the argument is Cascaded!, the size of the sheet on the Web is different from in PowerBuilder.
		5. In Windows 7, the upper part of the window (sheet) that is opened by using OpenSheet or OpenSheetWithParm may be overlapped by the MDI frame window. If this happens, you can work it around by using the Post function with

Function	Support Level	Coding Examples
		OpenSheet or OpenSheetWithParm as shown below: `post opensheet (w_main, w_mdi, ...)`
OpenSheet WithParm	Supported	String ls_str OpenSheetWithParm (w_main, ls_str, w_parent) Note: The window type in the function cannot be MDI or MDIHelp. The following syntax is supported: `OpenSheetWithParm(sheetrefvar, parameter{, windowtype}, mdiframe{, position{, arrangeopen}} )` Appeon supports appending a window list to a menu item on the Web.
OpenWithParm	Supported	OpenWithParm (w_main, w_parent) Notes: 1. The window type in the function cannot be MDI or MDIHelp. 2. On the Web, if the *parameter* argument is NULL, the argument will be stored in the PowerObjectParm property of the Message object.
*Note that Open, OpenSheet, OpenWithParm and OpenSheetWithParm support *windowtype* strings as parameters (see PowerBuilder syntax).		

9.1.2 Unsupported types

Unsupported

- The following are the unsupported system function categories that do not contain any supported individual functions:

  Table 9.21:

Class Definition Functions	DDE Server Functions	Garbage Collection Functions
Help Functions	Library Functions	Shared Object Functions (SharedObject Functions)
Tracing Functions		

- Having a parent object dynamically call a system function that is overloaded by its child object is unsupported.

9.2 User functions

Supported

- Object Functions
- Global Functions
- Global and local External Functions
- Global and local Remote Procedure Calls (RPC)

Unsupported

- Dynamic creation of global functions.

10 Events

10.1 Event types

Supported

- User-defined events can be attached to the Application object.
- User-defined events without an ID are supported.
- User-defined events with an ID are supported to the extent that the system message is supported.

Unsupported

- User-defined system messages are not supported.
- The system messages specific for the unsupported system objects/controls are unsupported.
- Selecting the same event ID to declare two events that have different names is not supported.
- Using the local variable AncestorReturnValue in an event of a descendant object is unsupported unless the event of the descendant object is an extended event from the ancestor object, or the AncestorReturnValue is generated in Call Super statement.
- If a system event involves a UI operation, the subsequent event will not be triggered until the UI operation is done.
- Avoid the following code, because it will cause system error to the deployed application:

  An event in an event sequence destroys some container (such as closing a window or destroying a user object), and then a subsequent event in the sequence calls to the container or controls/objects in the container.

  For example, in case uo_1 contains cb_1 and there is an event sequence (ue_1 -> ue_2 -> ue_3), if uo_1 is deleted in ue_2 but cb_1 is called in ue_3, the Internet Explorer that runs the sequence will crash.

- In some cases, the event triggering sequence is inconsistent between PB and Web, for example, the code as shown below:

```
event open;
String ls_Title,ls_FilePath,ls_FileName
Post Event pfc_PostOpen()
is_EventTrack += This.ClassName() + '.Open1()~r~n'
GetFileOpenName(ls_Title,ls_FilePath,ls_FileName)
is_EventTrack += This.ClassName() + '.Open2()~r~n'
end event
```

  On PB, the pfc_PostOpen event is triggered only after the Open event is executed completely. Whereas, on the Web, the pfc_PostOpen event is triggered right after the GetFileOpenName function is executed.

10.2 System messages

The following table lists the supported system messages.

Table 10.1:

Object/Control	Event ID	System Event using the ID
All	pbm_constructor	Constructor
	pbm_destructor	Destructor
	pbm_keydown	Key
Controls and windows	pbm_rbuttondown	RButtonDown
Window	pbm_activate	Activate
	pbm_close	Close
	pbm_closequery	CloseQuery
	pbm_deactivate	Deactivate
	pbm_dragdrop	DragDrop
	pbm_dragenter	DragEnter
	pbm_dragleave	DragLeave
	pbm_dragwithin	DragWithin
	pbm_hidewindow	Hide
	pbm_lbuttonclk	Clicked
	pbm_lbuttondblclk	DoubleClicked
	pbm_lbuttondown	MouseDown
	pbm_lbuttonup	MouseUp
	pbm_mousemove	MouseMove
	pbm_open	Open
	pbm_showwindow	Show
	pbm_size	Resize
	pbm_timer	Timer
CheckBox, CommandButton, Picture PictureButton, RadioButton, StaticText	pbm_bnclicked	Clicked
	pbm_bndragdrop	DragDrop
	pbm_bndragenter	DragEnter
	pbm_bndragleave	DragLeave
	pbm_bndragwithin	DragWithin
	pbm_bnkillfocus	LoseFocus
	pbm_bnsetfocus	GetFocus
DropDownList Box/ DropDownPicture ListBox	pbm_cbnkillfocus	LoseFocus
	pbm_cbnmodified	Modified
	pbm_cbnselchange	SelectionChanged

Object/Control	Event ID	System Event using the ID
	pbm_cbnsetfocus	GetFocus
DataWindow, DataStore	pbm_dwnchanging	EditChanged
	pbm_dwnbuttonclicked	ButtonClicked
	pbm_dwnbuttonclicking	ButtonClicking
	pbm_dwndragdrop	DragDrop
	pbm_dwndragenter	DragEnter
	pbm_dwndragleave	DragLeave
	pbm_dwndragwithin	DragWithin
	pbm_dwndberror	DBError
	pbm_dwnhscroll	ScrollHorizontal
	pbm_dwnitemchange	ItemChanged
	pbm_dwnitemchangefocus	ItemFocusChanged
	pbm_dwnitemvalidationerror	ItemError
	pbm_dwnkillfocus	LoseFocus
	pbm_dwnlbuttonclk	Clicked
	pbm_dwnlbuttondblclk	DoubleClicked
	pbm_dwnrbuttondown	RButtonDown
	pbm_dwnresize	Resize
	pbm_dwnretrieveend	RetrieveEnd
	pbm_dwnretrievestart	RetrieveStart
	pbm_dwnrowchange	RowFocusChanged
	pbm_dwnrowchanging	RowFocusChanging
	pbm_dwnsetfocus	GetFocus
	pbm_dwnupdateend	UpdateEnd
	pbm_dwnupdatestart	UpdateStart
SingleLineEdit, EditMask, MultiLineEdit, StaticText	pbm_endragdrop	DragDrop
	pbm_endragenter	DragEnter
	pbm_endragleave	DragLeave
	pbm_endragwithin	DragWithin
	pbm_enkillfocus	LoseFocus
	pbm_enmodified	Modified
	pbm_ensetfocus	GetFocus
OLEControl	(none)	Clicked
	(none)	PropertyChanged
	(none)	PropertyRequestEdit

Object/Control	Event ID	System Event using the ID
	pbm_omnclose	Close
	pbm_constructor	Constructor
	pbm_doubleclick	DoubleClicked
	pbm_omndragdrop	DragDrop
	pbm_omndragenter	DragEnter
	pbm_omndragleave	DragLeave
	pbm_omndragwithin	DragWithin
	pbm_omnrename	Rename
	pbm_omnsave	Save
	pbm_omnsaveobject	SaveObject
	pbm_omnviewchange	ViewChange
HScrollBar, HTrackBar, VScrollBar, VTrackBar	pbm_sbndragdrop	DragDrop
	pbm_sbndragenter	DragEnter
	pbm_sbndragleave	DragLeave
	pbm_sbndragwithin	DragWithin
	pbm_sbnlinedown	LineDown, LineRight
	pbm_sbnlineup	LineLeft, LineUp
	pbm_sbnthumbtrack	Moved
	pbm_sbnpagedown	PageDown, PageRight
	pbm_sbnpageup	PageLeft, PageUp
ListBox, PictureListBox	pbm_lbndblclk	DoubleClicked
	pbm_lbndragdrop	DragDrop
	pbm_lbndragenter	DragEnter
	pbm_lbndragleave	DragLeave
	pbm_lbndragwithin	DragWithin
ListView	pbm_lvnbegindrag	BeginDrag
	pbm_lvnbeginlabeledit	BeginLabelEdit
	pbm_lvnbeginrightdrag	BeginRightDrag
	pbm_lvncolumnclick	ColumnClick
	pbm_lvnclicked	Clicked
	pbm_lvndeleteallitems	DeleteAllItems
	pbm_lvndeleteitem	DeleteItem
	pbm_lvndoubleclicked	DoubleClicked
	pbm_lvndragdrop	DragDrop
	pbm_lvndragenter	DragEnter

Object/Control	Event ID	System Event using the ID
	pbm_lvndragleave	DragLeave
	pbm_lvndragwithin	DragWithin
	pbm_lvnendlabeledit	EndLabelEdit
	pbm_lvnsetfocus	GetFocus
	pbm_lvnitemchanging	ItemChanging
	pbm_lvnkillfocus	LoseFocus
	pbm_lvnrclicked	RightClicked
	pbm_lvnrdoubleclicked	RightDoubleClicked
Tab	pbm_tcnclicked	Clicked
	pbm_tcndoubleclicked	DoubleClicked
	pbm_tcnrclicked	RightClicked
	pbm_tcnrdoubleclicked	RightDoubleClicked
	pbm_tcndragdrop	DragDrop
	pbm_tcndragenter	DragEnter
	pbm_tcndragleave	DragLeave
	pbm_tcndragwithin	DragWithin
	pbm_tcnsetfocus	GetFocus
	pbm_tcnselchanging	SelectionChanging
TreeView	pbm_tvnbegindrag	BeginDrag
	pbm_tvnbeginlabeledit	BeginLabelEdit
	pbm_tvnbeginrightdrag	BeginRightDrag
	pbm_tvnclicked	Clicked
	pbm_tvndeleteitem	DeleteItem
	pbm_tvndoubleclicked	DoubleClicked
	pbm_tvndragdrop	DragDrop
	pbm_tvndragenter	DragEnter
	pbm_tvndragleave	DragLeave
	pbm_tvndragwithin	DragWithin
	pbm_tvnitemexpanded	ItemExpanded
	pbm_tvnitemexpanding	ItemExpanding
	pbm_tvnitempopulate	ItemPopulate
	pbm_tvnkillfocus	LoseFocus
	pbm_tvnselchanged	SelectionChanged
	pbm_tvnselchanging	SelectionChanging
	pbm_tvnsetfocus	GetFocus

Object/Control	Event ID	System Event using the ID
	pbm_tvnrclicked	RightClicked
	pbm_tvnendlabeledit	EndLabelEdit
Application	(None)	Open
	(None)	Close
	(none)	Idle
User Object	pbm_uondragdrop	DragDrop
	pbm_uondragenter	DragEnter
	pbm_uondragleave	DragLeave
	pbm_uondragwithin	DragWithin
Menu	(None)	Clicked
	(None)	Selected

Unsupported

The following table lists the unsupported system messages for the supported system objects/ controls:

Table 10.2:

Object/Control	Event ID	System Event using the ID
Controls and windows	pbm_other	Other
Window	pbm_help	Help
	pbm_ddedata	HotLinkAlarm
	pbm_ddeexecute	RemoteExec
	pbm_ddeadvise	RemoteHotLinkStart
	pbm_ddeunadvise	RemoteHotLinkStop
	pbm_dderequest	RemoteRequest
	pbm_ddepoke	RemoteSend
	pbm_syskeydown	SystemKey
	pbm_tbnmoved	ToolbarMoved
DataWindow, DataStore	(none)	Error
	pbm_dwnretrieverow	RetrieveRow
	pbm_dwnvscroll	ScrollVertical
ListView	pbm_lvnitemactivate	ItemActivate
	pbm_lvnsort	Sort
Application	(none)	IdleSystemError
OLEControl	(none)	Error
	(none)	ExternalException
	pbm_help	Help

Object/Control	Event ID	System Event using the ID
	pbm_other	Other

10.3 System message (non-standard EventID)

Supported

The following table lists the supported system messages.

Table 10.3:

Object/Control	Event ID (non-standard)	System Event using the ID
All	pbm_contextmenu	ContextMenu Note: Event bubbling is unsupported.
DataWindow	pbm_dwnkey	KeyDown
	pbm_dwndropdown	DropDown
	pbm_dwnrbuttonup	Rbuttonup
	pbm_dwnprocessenter pbm_rbuttondown pbm_lbuttondown pbm_rbuttonup pbm_lbuttonup	Processenter

Unsupported

Except for the system messages listed in the table above, all other system messages with non-standard event IDs are unsupported by Appeon. For example, pbm_enchange event is unsupported. Refer to PowerBuilder Help for more details.

11 DataWindow

11.1 DataWindow data sources

Supported

The following PowerBuilder data sources are supported:

- Quick Select
- SQL Select
- Query
- Stored Procedure
- External
- Web Service

Notes:

1. You can group your stored procedure in ASE Server.
2. If a DataWindow uses an external data source, the content of the DataWindow can only be saved as TXT type.
3. In the Web Service DataWindow, if the date data is empty or null, it will return a default value "0001-1-1” which will be parsed as “2001-1-1” on Web, whereas parsed as “0001-1-1” in PowerBuilder.

Unsupported

- Dynamically changing the SQL statement of an external data source is unsupported.
- If a DataWindow uses a stored procedure as its data source and the stored procedure has multiple result sets, the deployed DataWindow always takes the first result set.
- If a DataWindow uses a stored procedure as its data source, the Table.Select *property* cannot be modified, and the SetSQLSelect function cannot be executed.
- If a DataWindow uses an ASE stored procedure as its data source, the AutoCommit property must be set to TRUE and the chain must be off.

11.2 Using SQL statements in DataWindows

Requirements

- If you specify the table owner in the Specify Update Properties window in PowerBuilder, please make sure the "Enclose table and column names in quote" option is not selected in Appeon Developer during the deployment or errors will occur.
- If you specify the argument name in Specify Retrieval Arguments window in PowerBuilder, use ASCII characters.

- Do not use FOR UPDATE non-cursor statement in the SQL statement for DataWindows.

11.3 DataWindow presentation styles

Table 11.1:

CrossTab, Composite, Freeform, Graph, Grid, Group, Label, N-Up, Tabular, TreeView and RichText presentation styles	**Supported**
	DataWindow objects and controls in a DataWindow can be dynamically created (e.g. by using the CREATE statement). CrossTab, Composite, Freeform, Graph, Grid, Group, Label, N-Up, Tabular, TreeView and RichText style DataWindows are implemented in XML in Appeon deployed applications. Shortcut keys “Ctrl + Home” and “Ctrl + End” are supported in Grid, Freeform, Tabular and N-up DataWindows. The Web UI and behavior are identical with the original PowerBuilder DataWindows, except for a few differences listed below. Decimal retrieval argument is supported.
	Differences
	For all supported DataWindows, it ignores the upper or lower case when sorting by the column names. *For CrossTab style:* Refer to the detailed description of differences CrossTab DataWindow. *For Composite style:* Refer to the detailed description of differences Composite DataWindow. *For Freeform style:* When a user adjusts a scrollbar up and down, before the user releases the scrollbar, the DataWindow's contents are scrolled up and down in correspondence with the scrollbar's position. This does not happen in PowerBuilder applications. *For Grid style:* 1. Web application supports dragging and dropping columns in different locations on a Grid DataWindow. You can select a column header and drag it over other columns and drop the column in any location within the DataWindow. The selected column will be highlighted. 2. Sorting a column or selecting all the items by clicking the column header is unsupported. 3. In PowerBuilder, the tab sequence in a Grid DataWindow object is always left to right (except for right-to-left operating systems). Changing the tab value to any number other than 0 has no effect. In the Web Grid DataWindow, changing the tab value to any number other than 0 does have an effect. If the user presses Tab, the focus will change according to the predefined tab order.

	4. The color of the DataWindow border will be changed as the background color changes in the PowerBuilder application. This is not so in the Web application. *For Group style:* Refer to the detailed description of differences Grouping in DataWindow. *For Graph style:* Refer to the detailed description of differences Graph DataWindow. *For Label style:* RowFocusChange event is unsupported. *For N-Up style:* When dynamically changing the height of DataWindow, the number of pages is recalculated on the Web but it is not recalculated in PowerBuilder. *For TreeView style:* Refer to the detailed description of differences TreeView DataWindow. *For RichText style:* Refer to the detailed description of differences RichText DataWindow. For more DataWindow user operation differences, refer to DataWindow enhancements and differences.
Other DataWindow presentation styles	**Unsupported**
	OLE DataWindow

11.3.1 Composite DataWindow

Table 11.2:

Nested reports in a Composite DataWindow	**Supported**			
	Nested reports that are of the following presentation styles can be added into the Composite DataWindow, and the Web Composite DataWindow is implemented in XML:			
	CrossTab	Freeform	Graph	Grid
	Group	Label	N-Up	Tabular
	RichText	TreeView		
	Placing reports in the Group band is supported. Placing Group reports in any DataWindow band is supported.			
Retrieval arguments	**Supported**			
	Retrieval arguments of the nested report is supported in the composite DataWindow.			

<table>
<tr><td rowspan="3">Row-scrolling functions</td><td colspan="4">Unsupported</td></tr>
<tr><td colspan="4">Calling the following DataWindow functions in nested reports is unsupported:</td></tr>
<tr><td>ScrollToRow</td><td>ScrollPriorPage</td><td>ScrollNextPage</td><td>ScrollPriorRow</td></tr>
<tr><td></td><td>ScrollNxetRow</td><td></td><td></td><td></td></tr>
<tr><td>Performance differences between PowerBuilder and Web</td><td colspan="4">Differences
On the Web, if the Y property of a nested report is negative, the header band of the nested report overlaps the detail band.
On the Web, if the Trail_Footer property of a nested report control is set to True, the footer band of the nested report will be displayed after the summary band. If the Trail_Footer property is set to False, the footer band will always be displayed at the bottom of current DataWindow band.
On the Web, if the content in a Composite DataWindow cannot be displayed in one page horizontally, it will be displayed on a separate page.
On the Web, if you want to do a data retrieval for the Composite DataWindow, a transaction object always needs to be set.
If the Visible property of a nested report is dynamically changed from TRUE to FALSE, the nested report is counted as visible in PowerBuilder, but invisible on the Web, when the Composite DataWindow breaks pages or counts the height of the detail band.
The NewPage property will have effect in all DataWindow bands on the Web. However, it can only have effect in the detail band in PowerBuilder.
In the Detail band, the Height.Autosize property cannot be changed by using the Modify function.
In PowerBuilder, if the Summary band cannot be displayed within one page, the last row of the detail band and the whole summary band will be displayed in the next new page. However, on the Web, the summary band that cannot be fully displayed within the rest space of the page will be ignored.
If you set different print scales for a Composite DataWindow and the nested reports in the DataWindow, the scale for the nested reports will have no effect; instead, the scale for the Composite DataWindow will be effective for the DataWindow and the reports.
Scrolling a composite DataWindow using the PageUp/PageDown, UpArrow/DownArrow, or LeftArrow/RightArrow keys is unsupported.
If the width of the content of a nested crosstab datawindow cannot be displayed within the report control, the exceeding content will be ignored on the Web.</td></tr>
</table>

11.3.2 CrossTab DataWindow

Table 11.3:

Controls in CrossTab DataWindow	**Supported**	
	Button	Text
	Picture	GroupBox
	Line	Oval
	Rectangle	RoundRectangle
	Computed Field	Graph
	GroupBox is visible before inputting any data and is invisible after inputting data.	
	Unsupported	
	Column	Report
	TableBlob	
DataWindow object properties for CrossTab presentation style	**Supported**	
	Crosstab.SourceNames(r/w)	Crosstab.Rows(r)
	Crosstab.Columns(r)	Crosstab.Values(r)
	Crosstab.StaticMode (r)	Table.CrosstabData (r)
	Properties marked with "(r)" can be read - but not changed - in script. Properties marked with "(r/w)" can be read or changed in script. It is supported to use expressions like sum (units for crosstab) only in CrossTab.Values. It is unsupported to use such expressions in other properties that can contain expressions.	
	Unsupported	
	Help.TypeID.SetCrosstab	
DataWindow control methods for CrossTab DataWindow	**Supported**	
	GetMessageText	
	Unsupported	
	CrosstabDialog	
DataWindow expression functions for CrossTab DataWindow	**Supported**	
	CrosstabAvg	CrosstabCount
	CorsstabMax	CrosstabMin
	CrosstabSum	
Performance differences between PowerBuilder and Web	**Differences**	
	Adding columns in CrossTab DataWindow is unsupported, If you add a column in CrossTab DataWindow by "copy" and "paste" in PowerBuilder, the new column will not display on the Web.	

	When the footer band of a Crosstab DataWindow is higher than the DataWindow, detail band displays in PowerBuilder while footer band displays on the Web. If the argument of SetFilter() is NULL, in the Specify Filter dialog it displays the columns in detail band in PowerBuilder. On the Web, it displays the columns in detail band as well as the dynamically generated columns. After dynamically modifying Expression property of a computed field control in CrossTab DataWindow, data will be recalculated according to the new expression on the Web. In PowerBuilder, the data will not be recalculated. When dynamically modifying the properties that affect position, it automatically refreshes the interface on the Web but it does not in PowerBuilder. After retrieving data in CrossTab DataWindow, the position of controls (excluding Text control, Column control and Computed Filed control) will be rearranged on the Web. This is different from in that in PowerBuilder.
Unsupported features	**Unsupported**
	SlideLeft property is unsupported in CrossTab DataWindow. For graph control in CrossTab DataWindow, Values cannot contain aggregate functions. For a CrossTab nested report with arguments, retrieval arguments are unsupported.

11.3.3 Grouping in DataWindow

Grouped DataWindow objects can be created in two ways:

- Use the Group presentation style to create a grouped DataWindow object from scratch.
- Take an existing DataWindow object and define grouping

Table 11.4:

Group bands	**Supported**	
	Group Header band & Group Trailer band	
	Unsupported	
	None	
Group bands options in the PowerBuilder painter	**Supported**	
	Color	Height*
	Group Definition	Reset Page Count
	New Page on Group Break	Autosize Height*
	Group Sort*	

	Height When the Group Header is higher than the page, the header content will be printed into the footer area, and the content printed in the footer area will be printed again at the beginning of the next page. This is the behaviour on Web which is different from PowerBuilder. *Group Sort* If a DataWindow contains multiple groups and more than one group is specified with sort criterion in the Group Sort option, the criterion of the group with the largest group band ID will be valid for all the groups on the Web, while the other criteria will be ignored. It is only support to use aggregate functions in Group sort. *Autosize Height* the Autosize Height option is unsupported in Group Header and Group Trailer bands.
	Unsupported
	None
Properties for the Group keyword	**Supported**

	New page
	SyntaxFromSQL:
	`Group (colnum1, colnum2 NewPage)`
	ResetPageCount
	SyntaxFromSQL:
	`Group(col1 {col2...}... ResetPageCount)`

	Unsupported
	None
DataWindow functions for grouping	**Supported**
	Groupcalc* FindGroupChange *Groupcalc* In Web applications, the Groupcalc function will be executed automatically when values in a group are changed by dot notation or SetItem method.
DataWindow expressions for grouping	**Supported**
	Last / First Percent / CumulativeSum CumulativePercent
	Unsupported

	Large	Median
	Mode	Small
	Stdev	Stdevp
	Var Varp	

11.3.4 Graph DataWindow

Table 11.5:

Controls in Graph DataWindow	**Supported**	
	Graph DataWindow supports the following controls:	
	Button	Column
	Computed field	Graph
	GroupBox	Line
	Oval	OLE
	Rectangle	RoundRectangle
	Report	Picture
	Text	
	Unsupported	
	None.	
Graph axises	**Differences**	
	Category axis 1. When setting the categories on the Category axis, you should ensure that the category type and the Category axis type are the same. If they are not, Appeon adopts the first category type you input for this axis and will ignore the one that does not match along with the following. 2. If the expressions are for graphs (for example, sum (units for graph)), do not use it for the category axis. **Value axis** When setting the values on the Value axis, you should ensure that the value type and the Value axis type are the same. If they are not, Appeon adopts the first category type you input for this axis and will ignore the one that does not match along with the following.	
Properties for Graph DataWindow	**Differences**	
	The following DataWindow object properties can be read but not changed in Graph DataWindow:	
	Bandname.Text	HorizontalScrollMaximum
	HorizontalScrollMaximum2	QueryMode
	QuerySort	Rows_Per_Detail

	VerticalScrollMaximum	Zoom
Performance differences between PowerBuilder and Web	**Differences**	
	On the Web, the UI will be automatically refreshed after executing RowsCopy, RowsDiscard, RowsMove and Sort. This is different from in PowerBuilder. Executing GetObjectAtPoint function of DataWindow control does not return the row number. When multiple series are of different data types, the series will be sorted and displayed according to the order of the String data type. It is unsupported to have a nested report in a Graph DataWindow. It is unsupported to operate the properties before drawing a Graph DataWindow. It is unsupported to use overlays in graphs in the DataWindow. The value of the header height cannot be smaller than 0. In pie graph, if there is negative data in a series, the graph displays differently on the Web from in PowerBuilder. In PowerBuilder, the total percentage exceeds 100%. But Appeon ignores the negative data, it display 100% on the Web.	

11.3.5 RichText DataWindow

Table 11.6:

DataWindow control properties in RichText DataWindow	**Unsupported**
	DragAuto property is unsupported.
DataWindow control events in RichText DataWindow	**Differences**
	For ItemFocusChange event: After retrieving data in a RichText DataWindow on the Web, it triggers the ItemFocusChange event. Executing SetRow function triggers the ItemFocusChange event on the Web. *For RowFocusChanged event:* On the Web, executing InsertRow or DeleteRow may not trigger the RowFocusChanged event. This is different from that in PowerBuilder. Executing Clipboard, ImportFile and ImportString functions do not trigger the RowFocusChanged event on the Web. This is different from that in PowerBuilder. *For KeyDown event:*

	Some keystrokes do not affect the formatting of documents in RichText DataWindow on the Web. For example, pressing Esc after updating data in an inputfield reverts to the old data in PowerBuilder but pressing the Esc key on the Web does not affect the updating of the data and the changes will be made.
	Unsupported
	PrintStart and Printed events are unsupported.
DataWindow control functions for RichText DataWindow	**Supported**
	The following functions of DataWindow control are effective only in RichText DataWindow:
	CopyRTF / InsertDocument*
	PasteRTF / ShowHeadFoot
	InsertDocument The value of the filetype argument can be FileTypeRichText! or FileTypeText!. the FileTypeHTML! and FileTypeDoc! are still unsupported values in current version.
	Differences
	For Clicked function, Xpos argument specifies the distance of the pointer from the left side of the DataWindow workspace and Ypos argument specifies the distance of the pointer from the top of the DataWindow workspace on the Web. This is different from the actual behavior of PowerBuilder. On the Web, the return value of GroupCalc function in RichText DataWindow is the same as in DataWindows of other presentation styles, which differs from that in PowerBuilder. On the Web, some functions have differences in triggering events in RichText DataWindow with that in PowerBuilder. They behave the same as in DataWindows of other presentation styles. These functions are AcceptText, ShareDataOff, SetSQLSelect, SetSort, Sort, SetFilter, Filter, SetFormat, RowsCopy, RowsDiscard and RowsMove.
	Unsupported
	GetClickedRow and GetBandAtPointer functions are unsupported.
Performance differences between PowerBuilder and Web	**Differences**
	For RichText DataWindows, clicking the header or footer bands triggers the GetFocus event on the Web. This is different from that in PowerBuilder. It will not verify the type of the column if it has the initial value when inserting a column on the Web. This is different from that in PowerBuilder. On the Web, if all data is filtered when executing SetFilter, it displays "??". This is different from that in PowerBuilder.

	Double clicking a picture in a RichText DataWindow triggers DoubleClicked event on the Web. This is different from that in PowerBuilder. On the Web, if the focus is in a computed field, executing GetColumnName returns only the column name excluding the name of the computed field. This is different from that in PowerBuilder. If there are multiple DataWindows of RichText and other presentation styles in a Window, the last-created RichText DataWindow is always on the top. If the size of the pasted picture is larger than the RichTextEdit control, the horizontal and vertical scroll bars will be automatically displayed on the Web, though the HScrollBar and VScrollBar properties are set to False. Alignment buttons on the Web are only effective if the WordWrap property is set to True. This is different from that in PowerBuilder. The display of scroll bars in RichText DataWindow is the same as that in RichTextEdit control. Also, if the WordWrap property of RichTextEdit control is set to True on the Web, it does not display the horizontal scroll bar. The behavior of the RichText bars on the Web is consistent with that in PowerBuilder 9. Find and Replace methods are partially supported. Chinese font family will not take effect on the Web. For RichText DataWindows without containing any filed, if it is created in PowerBuilder 9 and 10, the RichText DataWindow will cause IE crash when it converts to the Web.
Unsupported features	**Unsupported**
	It is unsupported to use Appeon DataWindow menu. It is unsupported to modify the column value in the pop-up window when right clicking a input field in a RichText DataWindow. It is unsupported to dynamically create RichText DataWindows (for example, using Create statement). It is unsupported to use RichText DataWindow as NVO, DataStore and DropDownDataWindow. End identifier is unsupported. It is unsupported to use the shortcut "Ctrl+Z" on the Web. It is unsupported to set the paragraph alignment for the Text Object on the Web. The "text-wrap" is unsupported on the Web.

11.3.6 TreeView DataWindow

Table 11.7:

DataWindow object properties for TreeView presentation style	**Supported**		
	AccessibleDescription	AccessibleName	AccessibleRole
	CollapsedTreeNodeIconName	DefaultExpandToLevel	ExpandedTreeNodeIconNam
	Indent	Level	SelectNodeByMouse
	ShowNodeConnectLines	ShowLeafNodeConnectLines	ShowTreeNodeIcon
	StateIconAlignMode	TreeNodeIconName	
	Unsupported		
	QueryMode	QuerySort	QueryClear
	RtoLLayout		
DataWindow control functions for TreeView DataWindow	**Supported**		
	Collapse	CollapseAll	CollapseAllChildren
	CollapseLevel	Expand	ExpandAll
	ExpandAllChildren	ExpandLevel	IsExpanded
	SelectTreeNode		
DataWindow control events for TreeView presentation style	**Supported**		
	Collapsing	Collapsed	Expanding
	Expanded	TreeNodeSelecting	TreeNodeSelected
	Unsupported		
	None		
Performance differences between PowerBuilder and Web	**Limitations**		
	DataWindows in TreeView presentation style are restrained by all limitations for DataWindows in Group presentation style. Collapsing, Collapsed, Expanding, and Expanded events cannot be triggered if the nested TreeView DataWindows. If the nest level in a TreeView DataWindow is 3, the user may not correctly expand and collapse the tree node when clicking the mouse. The height for the Tree.level band cannot be negative value. You can not set a column to current column using **SetColumn** function if the column cannot be focused. If the **ScrollNextPage** function is called in the last page, the current row on the Web will be set to the last row. This is different from that in PowerBuilder. On the Web, the **SetSort** function sorts data before groups data. This is different from that in PowerBuilder.		

	If the Shortcut in PowerBuilder application is the same as the shortcut of IE, shortcut in IE has high priority than shortcut settings in PowerBuilder.
Event sequence difference	**Difference**
	The TreeNodeSelecting, TreeNodeSelected, RowFocusChanging and RowFocusChanged events cannot be triggered in the preview mode. Selecting Tree Node on the Web will sequentially trigger the events TreeNodeSelecting, TreeNodeSelected, RowFocusChanging, RowFocusChanged. Clicking the TreeView DataWindow on the Web will trigger the following events in sequence: LButtonDown, RowFocusChanging, RowFocusChanged, TreeNodeDelecting and TreeNodeSelected. On the Web calling **ScrollNextPage** function will trigger the following event in sequence: Rowfocuschanging, Rowfocuschanged, and ScrollVertial. Collapsing or expanding the tree node will trigger the collapsing(/expanding), collapsed(/expanded) and Resize events on the Web in sequence. Do not assign values to rows in the grouping column in a loop. You can use dot notation to directly assign values to corresponding rows.

11.4 Displaying and validating data

Data validation

Appeon supports data validation in the following two ways:

- Validating whether the data is of a correct data type and in the allowed data scope
- User-defined validation expression

The validation expressions must be DataWindow expressions that Appeon supports (see DataWindow operators and expressions section).

Define display format

- For Web applications, formats specified in the PowerBuilder application has a higher priority than formats specified in the client machine.
- The format property is supported for column edit style Edit and EditMask. The display formats are listed in the following table:

Table 11.8:

Using sections	**Supported**
	Each type of display format can have multiple sections, with each section corresponding to a form of the number, string, date, or time. Only one

	section is required. On the web, the sections can be no longer than two. For example:
	$#,##0;($#,##0) //Supported
	$#,##0;($#,##0);($#,##0.00) //Unsupported

Number	**Supported**
	Special characters: "#", "0", "%", "$", "E-", "E+", "e-", "e+", "~" Keywords: [General], [Currency] Separator: ",", "." Default format: ###,###.00
	Unsupported
	lf the currency is not US dollars, it is better to specify the format with the dollar symbol replaced with the actual currency symbol to ensure that the currency symbol displays correctly in the deployed Web application. For example, if the format is ##,##0.00, then do not rely on the setting of the client machine to add the actual currency symbol.
String	**Supported**
	Special characters: "d", "dd", "ddd", "dddd", "m", "mm", "mmm", "mmmm", "yy", "yyyy" Separator: "-", ":", "/", ".", " "(space), ", " Separator: Chinese characters for year, month or day can be separators Keywords: [ShortDate], [LongDate] Default format: mm/dd/yy
	Unsupported
	None.
Time	**Supported**
	Special characters: "h", "hh", "m", "mm", "s", "ss", "ffffff", "AM/PM", "am/pm", "A/P", "a/p" Separators: "-", ":", "/", ".", " "(space), "," Keywords: [Time] Default format: hh:mm:ss:fff
	Unsupported
	Chinese characters as AM and PM symbols in the Control Panel | Regional Options are unsupported for Web applications.
DateTime	**Supported**
	Separators: "/", "-" , " " (space) Default format: mm/dd/yyyy hh:mm:ss:ffff When setting the date format to mm/dd/yyyy, an invalid date will turn to 01/01/1900 in Appeon while 00/00/0000 in PowerBuilder.

	Unsupported
	None.

11.5 Dynamic DataWindow

DataWindow objects and all entities in them have a set of properties. You can look at and change the values of these properties during execution using DataWindow methods or property expressions. You can also create DataWindow objects during execution. Appeon supports dynamically creating DataWindows with the following presentation style: FreeForm, Tabular, Grid, Graph, CrossTab, N-Up, Label, Composite, Group, TreeView. However, dynamically created RichText DataWindow is unsupported. Also DataWindow cannot be dynamically created if the database type is Informix.

Important: Configuration required for supporting dynamic DataWindows

Configuration at the database server is required for support of dynamic DataWindows: find the following sql files in *%Appeon_Server_Installation_Path%\appeon\sql\dynamicsql*, and install the corresponding SQL for the database you are using. Note that you should execute the SQL statement in the database console and executing it in PowerBuilder may cause errors.

- install_appeon_syntaxfromsql_ORACLE.sql
- install_appeon_syntaxfromsql_MSSQL.sql
- install_appeon_syntaxfromsql_ASE.sql
- install_appeon_syntaxfromsql_ASA.sql
- install_appeon_syntaxfromsql_DB2.sql
- install_appeon_syntaxfromsql_IQ.sql
- install_appeon_syntaxfromsql_teradata.sql
- install_appeon_userfunction_teradata.sql
- Appeon_teradata_extfun.jar

You can uninstall the SQL files and disable the support of dynamic DataWindows by running the following files in the same folder:

- uninstall_appeon_syntaxfromsql_ORACLE.sql
- uninstall_appeon_syntaxfromsql_MSSQL.sql
- uninstall_appeon_syntaxfromsql_ASE.sql
- uninstall_appeon_syntaxfromsql_ASA.sql
- uninstall_appeon_syntaxfromsql_DB2.sql
- uninstall_appeon_syntaxfromsql_IQ.sql

- uninstall_appeon_syntaxfromsql_teradata.sql
- uninstall_appeon_userfunction_teradata.sql

Notes:

1. When installing or uninstalling Appeon-provided SQL files, you must use the corresponding SQL executing tool of each database as listed below:

   Table 11.9:

ASA/SQL Anywhere	Interactive SQL
SQL Server	Query Analyzer
DB2	Command Editor
Oracle	SQLPlus WorkSheet
Sybase IQ	Interactive SQL
SQL Server	Query Analyzer
Teradata	Teradata client tool

2. (**For Teradata only**) Before installing the SQL file install_appeon_syntaxfromsql_teradata.sql for the Teradata database, you will need to:

   a. Copy **Appeon_teradata_extfun.jar** and **install_appeon_userfunction_teradata.sql** to the machine where the Teradata database client tool is installed.

   b. Open **install_appeon_userfunction_teradata.sql** and modify the first line to point to the physical path of **Appeon_teradata_extfun.jar**.

   c. Execute **install_appeon_userfunction_teradata.sql** via the Teradata client tool.

3. After installing the SQL files, you should restart Appeon Server or redeploy the application to disable the caches.

4. When you use the transaction SyntaxFromSQL method, please note the following differences between PowerBuilder and the Web:

   - The length of return value on the Web is different from PowerBuilder, because the return value on the Web carries the names of all columns while the return value in PowerBuilder does not.

   - Text, the object keyword, is unsupported if there are calculation involved in SQL statements. For example:

   ```
   Select emp_ID,emp_Name,salary/12,
   "Dept="+emp_deptname from employee.
   ```

5. When using transaction SyntaxFromSQL method for DB2 on the Web, please note that the generated DataWindow contains no primary key information and cannot be updated on the Web under the following conditions:

   - The database table contains only a simple index.

- The database table contains a primary key that uses alias.
- Access a table that contains a primary key as the table owner (for example, using SELECT * FROM DB2ADMIN.PUB_T_DEPARTMENT).

Executing SaveAs function for dynamically created DataWindows

1. A dynamic DataWindow can only be saved as TXT, HTML or XLS file. Saving it to WMF will give the error "DataObject information does not exist". The reason is that SaveAs to TXT, HTML or XLS is directly executed at the client while SaveAs to other formats is executed by the server. Appeon Server cannot save the DataWindow since it cannot get the dynamically-created SQL.
2. Timestamp data type is unsupported when dynamically creating DataWindows.
3. It is unsupported to have stored procedure with input parameters as the data source when dynamically creating DataWindow.
4. It is unsupported to use EXECUTE statement to execute the SQL when dynamically creating DataWindows. For example:

```
Execute ('select * from
employee'). //Unsupported
```

5. If the DataWindow object is created dynamically, the SetSQLSelect function is unsupported.

Methods of dynamically creating DataWindows

1. Using DataWindow Create method.

   Supported syntax:

   ```
   Integer dwcontrol.Create (string syntax{, string errorbuffer})
   ```

   Unsupported: Using PBSELECT statement in *syntax* to create DataWindows is unsupported.

2. Using transaction SyntaxFromSQL method for ASE, ASA/SQL Anywhere, SQL Server and Oracle database type.

   Supported syntax:

   ```
   transaction.SyntaxFromSQL(sqlselect, presentation, err)
   ```

   When you use the transaction SyntaxFromSQL method, please note the following differences between PowerBuilder and Web: The length of return value on the Web is different from PowerBuilder, because the return value on the Web carries the names of all columns while the return value in PowerBuilder does not.

Note: It is unsupported to dynamically create DataWindow if the column alias specified in the SQL has the same name as the column name in the quoted table. For example, if a table t1 has the column c1, c2 and c3, syntax like "SELECT c1= c2* 10, c3 FROM t1" is unsupported.

Methods of dynamically modifying DataWindow properties

1. Using property expression.

   Please refer to DataWindow | DataWindow operators and expressions for detailed information.

2. Using DataWindow Modify function .

   Supported syntax:

   ```
   string dwcontrol.Modify(string modstring)
   ```

   When you use the Modify function, be aware of that: If the Modify function is used to modify the WHERE clause of the DataWindow object's SQL SELECT statement, make sure that the new SQL SELECT statement is correct in syntax. Otherwise, syntax errors can occur because PowerBuilder does not validate the statement whereas JavaScript does.

   You should verify that the data types of both sides of the equal symbol in the expression are compatible.

Adding or deleting controls in DataWindow objects

You can use Modify method to create and destroy the controls in a DataWindow object while noting the following issues.

1. When you use this method to add controls in a DataWindow, the name of the controls should not be same.

2. If it is failed to add or delete a control in the DataWindow object, the error message reported on the Web may be different from in PowerBuilder.

11.6 DataWindow operators and expressions

Table 11.10:

DataWindow operators	**Supported**
	Arithmetic operators: +, -, *, /, ( ), ^ Relational operators: =, >, <, <>, >=, <=, BETWEEN...AND, IN, LIKE, and any of these operators in combination with NOT Logical operators for all data types: NOT, AND, OR Concatenation for string data types: +
	Unsupported
	Subtraction (-). The setting of the DashesInIdentifiers property is ignored. For example, "A-B" always means subtract B from A.
Operator precedence in DataWindow expressions	**Difference from PowerBuilder**
	The precedence of "AND" is higher than that of "OR" (in PowerBuilder, "AND" and "OR" have the same precedence). The operators >, <, <=, >= are of higher precedence than = and <> (in PowerBuilder, the operators >, <, <=, >=, =, <> are of the same precedence).

	Note: You can use parentheses to make sure the Web application has correct precedence effect as in the PowerBuilder application. Original expression: *a OR b AND c* Modified expression: *(a OR b) AND c*
Supported & Unsupported expressions	**Supported**
	Conditional expressions for property values (such as Visible, X, Y, Width, Height, BackgroundColor, TextColor) Expressions for computed fields Validation rules Filter and sort criteria Note: The information on the limitations for these types of expressions is provided in the Specific Requirements for Using Expressions.
	Unsupported
	Series and values in graphs Columns, rows, and values in crosstabs
Functions used in DataWindow expressions	**Supported**

Abs	Acos
Asc	Asin
Atan	Avg
Bitmap	Ceiling
Describe	Char
Case	Cos
Count	CrosstabAvg
CrosstabCount	CrosstabMax
CrosstabMin	CrosstabSum
CumulativePercent	CumulativeSum
CurrentRow	Date
DateTime	Day
DayName	DayNumber
DaysAfter	Exp
Fact	Fill
First	GetText
GetRow	Hour
If	Int
Integer	IsDate
IsNull	IsNumber
IsRowModified	IsRowNew

IsSelected	IsTime
Last	Large
LastPos	Left
LeftTrim	Len
Log	LogTen
Long	LookUpDisplay
Lower	Match
Max	Median
Mid	Min
Minute	Mod
Month	Now
Number	Page
PageCount	Percent
Pi	Pos
Rand	Real
RelativeDate	RelativeTime
Replace	RGB
Right	RightTrim
Round	RowCount
Second	SecondsAfter
Sign	Sin
RowHeight	Small
Space	Sqrt
Stdevp	Stdev
String	Sum
Tan	Time
Today	Trim
Truncate	Upper
WordCap	Year
In addition to the above functions, **User functions** can be used in DataWindow expressions.	
Unsupported	
Mode	PageAcross
PageCountAcross	ProfileInt
ProfileString	Var
VarP	

	In addition to the above functions, **System functions** and **External functions** cannot be used in DataWindow expressions.

Specific Requirements for Using Expressions

Rule for all DataWindow expressions

1. Expressions involving Null values may arrive at different values in JavaScript from their values in PowerScript. For more details, please refer to the Null Values section.
2. Expressions cannot contain "~~t".
3. Expressions cannot be used in cursor or stored procedure arguments.
4. For nested structures, please verify that:
   - It is supported to have single quotes nested with single quotes or double quotes nested with single quotes.
   - It is unsupported to have single quotes nested with double quotes or double quotes nested with double quotes.
   - A multi-layer nested structure may cause problems.
5. If using user-defined global functions in DataWindow expressions,
   - Arguments and returned values can only be simple data tapes: Int/Integer, Boolean, String, Character, Long, UnsignedLong, UnsignedInteger.
   - Arguments cannot be references.
   - If the computed field expression is a computed field or global function, the formatting for the computed filed will not take effect on the Web.

Sort and Filter expressions

1. Unsupported functions: Avg, Count, Max, Min & Sum.
2. Behavioral difference: In a Web application, if a Find, Filter or Sort expression contains any special characters (for example, ".", "", "/"), the execution result may differ from PowerBuilder. In a Web application, the DataWindow rows may display in a different order from PowerBuilder.

Property expressions

1. Overlapped quotes, for example, "sdf~"sdf", "dfg'sdf".
2. In the Web application, modifying a DataWindow property in a DataWindow expression may conflict with the settings in the application's source code.

String expressions

Operands in the string expression cannot a mix of constants and variables.

Decimal Precision in DataWindow expression

28-digit Decimal is only supported in the following DataWindow expressions: Abs, Avg, CamulativeSum, Median, Sign, and Sum.

11.7 DataWindow objects and their properties

11.7.1 DataWindow object

Table 11.11:

Controls in a DataWindow	**Supported & Unsupported**	
	Refer to the Controls in a DataWindow and their properties section for details.	
Column edit styles	**Supported**	
	CheckBox	DropDownDataWindow
	DropDownListBox	Edit
	EditMask*	RadioButtons
	Unsupported	
	InkEdit (DropDown Calendar property of EditMask is unsupported.)	RichText
DataWindow bands	**Supported**	
	All DataWindow bands, including Header band, Detail band, Summary band, Footer band, Trailer band, and Tree.Level band are supported. All DataWindow bands can contain any supported control.	
	Unsupported	
	The height of the Header band cannot be smaller than 0. The header and footer bands are unsupported in the DataWindow with the RichText presentation style.	
	Notes	
	1. Do not use the band name as the name of the DataWindow object, otherwise the execution result on Web will be different from that on PowerBuilder. 2. If the content in the summary band cannot be displayed fully on the last page of the deployed DataWindow, an additional page will be displayed to show the rest of the content.	
Other	**Supported**	
	DataWindows that contain Blob columns are unsupported.	

11.7.2 DataWindow object properties

Supported

If a property is marked as "supported (r)" or "(r)", it means that the property can be read in script, but not changed in script, and setting the property in the painter has no effect in the converted application.

If a property is marked as "supported (r/w)" or "(r/w)", it means that the property can be read or changed in script, and setting the property is effective in the painter.

The table below lists the supported properties for DataWindow object:

Table 11.12:

Property	What's Supported	What's Unsupported
Attributes	Supported (r)	
Bandname.property	(r/w): Detail.Color, Detail.Height, Detail.Height.Autosize, Detail.Pointer, Footer.Color, Footer.Height, Footer.Pointer, Header.Color, Header.Height, Header.Pointer, Summary.Color, Summary.Height, Summary.Pointer, Trailer.Color, Trailer.Height, Trailer.Pointer, Height.Autosize	
*Bandname.*Text	Supported	
Bands	Supported	
BitmapName	Supported	
Color	Supported	
Column.count	Supported	
Crosstab.*property*	(r/w): Crosstab.SourceNames (r): Rows, Columns, Values, StaticMode, CrossTabData	
Data	Supported	
Data.XML	Supported	
Data.XMLDTD	Supported	
Data.XMLSchema	Supported	
FirstRowOnPage	Supported	
Grid.ColumnMove	Supported	
Grid.Lines	Supported	
HTMLDW	Supported	
HTML Gen .property	Supported	
HTML Table .property	Supported	
Label.property	Supported	

Property	What's Supported	What's Unsupported
LastRowOnPage	Supported (r)	
Message.Title	Supported	
Name	Supported	
Nested	Supported	
Objects	Fully supported (r) Note: In the Web application, the returned object names are listed in a different order from that in PowerBuilder.	
Pointer	Supported	
Print.*property*	**Supported** (r/w): Color, Collate, Columns, Columns.Width, Copies, DocumentName, Duplex, Filename, Margin.Botton, Margin.Left, Margin.Right, Margin.Top, Orientation, Page.Range, Page.RangeInclude, Paper.Size, Paper.Source, Preview, Preview.Rulers, Preview.Zoom, PrinterName, Prompt, Quality, Scale, CustomPage.Width, CustomPage.Length Notes: 1. When Paper.Size is the default value (0), in PowerBuilder application, the default paper size for the printer will be used for the output; while in Appeon Web application, the paper size for the local printer will be used, but if it cannot be obtained, then the monitor screen size (for example 1024x768) will be used. 2. In Appeon Web application, the Copies property will not take effect and only one copy will be printed. 3. In Appeon Web application, the Scale property will not take	**Unsupported** CanUserDefaultPrinter, ClipText, OverridePrintJob

Property	What's Supported	What's Unsupported
	effect and the output always prints as 100% of scale. 4. The value of MarginBottom, MarginLeft, printMarginTop, MarginRight, PreviewZoom, Scale cannot be negative. 5. In PowerBuilder, if a printer driver does not support scaling, scaling is not supported in either DataWindow printing or preview. However, in the same scenario on the Web, scaling can work in DataWindow preview. 6. When executing PDF printing, if the DocumentName is an empty string, on the Web, the DataWindow will be automatically saved as naming DataWindow.pdf. 7. Filename cannot be an empty string. 8. Newspaper Columns Across option under Columns property is supported for all DataWindows except for composite and nested DataWindows.	
Print.Buttons	Supported	
Print.Preview .Buttons	Supported	
Printer	Supported	
Processing	Supported (r)	
ReadOnly	Supported (r)	
Rows_Per_Detail	Supported	
Tooltip.property	Supported	
QueryClear	Supported Note: By setting the QueryClear property to Yes, events that are triggered by the property on the	

Property	What's Supported	What's Unsupported
	web are different from those in PowerBuilder.	
QueryMode	Supported **Extending support:** For columns of all types (except string), if the QueryMode is set to Yes, calling Find functions will take effect on the Web but not in PowerBuilder. **Limitations:** In the query mode, calling the following functions is not recommended because the result of calling functions on the Web is different from in PowerBuilder: InsertRow, DeleteRow, ResetUpdate, RowsDiscard, Update, SetItemStatus, Sort, GroupSort, CopyData, MoveData, GetFullState, SetFullState, GetChanges, SetChanges. If the QueryMode is set to Yes, it is unsupported to insert or delete rows in a DataWindow. If the QueryMode is set to Yes, retrieving data by column alias will cause errors on the Web. QueryMode is unsupported in TreeView DataWindow. If the QueryMode is set to Yes, executing DataWindow SaveAs function will not reset the query criteria on the Web.	
QuerySort	Supported	
RichText.property	Supported	
Selected	Supported	
Selected.Data	Supported	
Selected.Mouse	Supported	If the Selected.Mouse is set to Yes, it is unsupported to select multiple columns by pressing Shift and Ctrl keys.

Property	What's Supported	What's Unsupported
ShowDefinition	Supported (r)	
SlideLeft	Supported	
SlideUp	Supported	
Sparse	Supported	
Storage	Supported	
StoragePageSize	Supported	
Syntax	Supported	
Syntax.Data	Supported	
Syntax.Modified	Supported	
Table.*property*	(r/w): Arguments, CrosstabData, Data.Storage, Delete.Argument, Delete.Method, Delete.Type, Filter, Select.Attribute, Update.Argument, Update.Method, Update.Type, Select, GridColumns, Insert.Argument, Insert.Method, Insert.Type, Select.Attribute, Sort, Update.Argument, Update.Method, Update.Type, UpdateKeyInPlace, UpdateTable, UpdateWhere (r): SQLSelect, Procedure In PowerBuilder, the Table.Select property contains the escape character "~" But after the execution of the SetTransObject function, "~" will be removed. During parsing of the application, "~" will be translated into a newline return. After parsing, no "~" will be found in the Table.Select property. In PowerBuilder, by default, the Ascending criteria will be included in the value of the Table.Sort property. However, in the Web application, the Sort is a string containing the sort criteria set by the user. Even though the string will not include the default Ascending criteria, the result will be the same as in PowerBuilder. For a Web application that has more than one user accessing the	Table.property is not supported for External data sources.

Property	What's Supported	What's Unsupported
	same tables concurrently, set the DataWindow Table.UpdateWhere property to 1 ("Key and updatable columns") to avoid data loss.	
VerticalScroll Maximum	Supported	
VerticalScroll Position	Supported	
Zoom	**Partially Supported** Notes: 1. It is unsupported to zoom DataWindow object when the window' s size is also changing. 2. The effect of a zoom factor appears differently in PowerBuilder than in the Web application.	
Units	**Supported** Only the type of PowerBuilder units is supported	

Unsupported

None of the Graphical DataWindow properties are supported.

The table below lists the unsupported properties for DataWindow objects:

Table 11.13:

Data.HTML.Table	Data.XMLWeb	HideGrayLine	Data.XSLFO
Font.Bias	Help.property	HorizontalScroll Maximum	HorizontalScroll Maximum2
HorizontalScroll Position	HorizontalScroll Position2	HorizontalScrollSplit	HTMLDW
OLE.Client.property	Retrieve.AsNeeded	Row.Resize	Table.sqlaction. property
Timer_Interval			

11.7.3 Controls in a DataWindow and their properties

Notes:

- Avoid placing controls above or behind other controls in a DataWindow, as overlapping controls are displayed differently on the Web than in PowerBuilder. For example, if there is a Text control behind a DropDownDataWindow column in a DataWindow, the DropDownDataWindow field cannot be pulled down correctly.

11.7.3.1 Button control

Supported

If a property is marked as "supported (r)" or "(r)", it means that the property can be read - but not changed - in script, and properties set in the painter will not take effect in the converted application.

If a property is marked as "supported (r/w)" or "(r/w)", it means that the property can be read or changed in script, and properties set in the painter will take effect.

Table 11.14:

Property	What's Supported	What's Unsupported
Action	(r/w): User Defined, Retrieve, PageNext, PagePrior, PageFirst, PageLast, Sort, Filter, DeleteRow, AppendRow, InsertRow, Update, SaveRowsAs, Print, Cancel, Preview, PreviewWithRulers, QueryMode, QuerySort, Query, Clear	This property cannot be set by using DataWindow expression. Retrieve (Yield)
Attributes	Supported (r)	
Background .property	Fully supported (r/w): Color, Mode This property can be set by using DataWindow expression.	
Band	Supported (r): Band, Background, Foreground	
Color	Fully supported (r/w) This property can be set by using DataWindow expression.	
DefaultPicture	Supported (r/w) This property cannot be set by using DataWindow expression.	
Filename	Fully supported (r/w)	
Font.property	(r/w): Charset, Escapement, Face, Family, Height, Italic, Pitch, Strikethrough, Underline, Weight Note: 1. When the value of Escapement is a negative number, the text will not be rotated. 2. The text rotated can be displayed out of the control in PowerBuilder, but not in the Web application.	Width

Property	What's Supported	What's Unsupported
	3. After rotated, the multiple-line text still displays in multiple lines in PowerBuilder, but displays in one line on Web.	
Height	Fully supported (r/w) This property can be set by using DataWindow expression.	
HTextAlign	Fully supported (r/w)	
Name	Supported (r)	
Pointer	Fully supported (r/w)	
SlideLeft	Supported (r/w)	
SlideUp	Supported (r/w)	
Tag	Fully supported (r/w)	
Text	Fully supported (r/w)	
Type	Supported (r)	
VTextAlign	Supported (r/w)	
Visible	Fully supported (r/w) This property can be set by using DataWindow expression.	
Width	Fully supported (r/w) This property can be set by using DataWindow expression.	
X	Fully supported (r/w) This property can be set by using DataWindow expression.	
Y	Fully supported (r/w) This property can be set by using DataWindow expression.	

Unsupported

- HideSnaked
- Movable
- Resizable
- SuppressEventProcessing

11.7.3.2 Column control

Supported

If a property is marked as "supported (r)" or "(r)", it means that the property can be read - but not changed - in script, and properties set in the painter will not take effect in the converted application.

If a property is marked as "supported (r/w)" or "(r/w)", it means that the property can be read or changed in script, and properties set in the painter will take effect.

Table 11.15:

Property	What's Supported	What's Unsupported
Accelerator	Fully supported (r/w)	
Alignment	Left!, Center!, Right!, Justify! In PowerBuilder, when the Edit.password property is YES and the alignment of the column's text is center-aligned, the text is left-aligned when it is being edited. After the editing, the text is center-aligned. In the Web application, the text is always center-aligned.	None
Attributes	Supported (r)	
Background.*property*	Fully supported (r/w): Color, Mode	
Band	(r): Band	
BitmapName	Fully supported (r/w)	
Border	(r/w): NoBorder!, Box!, Lowered!, Raised!, ShadowBox!, Underline!, ResizeBorder!	
CheckBox.*property*	(r/w): LeftText, Off, On, Other, Scale, Text, ThreeD, ThreeState	It is unsupported to dynamically change the edit style of a column by modifying this property.
Color	Fully supported (r/w) Set this property using DataWindow expression.	

ColType	Supported (r)
	`dw_control.object.columnname.coltype`
	`dw_control.Describe("columnname.coltype")`	

Criteria.*property*	Fully supported (r/w)	

dbName	Supported (r)
	Example:
	`dw_control.object.columnname.dbName`
	`dw_control.Describe("columnname.dbName")`	

Property	What's Supported	What's Unsupported

dddw.*property*	(r/w): AllowEdit, AutoHScroll, AutoRetrieve, DataColumn, DisplayColumn, HscrollBar, Limit, Name, NilIsNull, PercentWidth, Required, UseAsBorder, VscrollBar, Case Lines, ShowList.
	(r): HSplitScroll
	The size and font size of Web DropdownDataWindows are determined by the size of the deployed DataWindow; the width of Web DropdownDataWindows, if set to be smaller than the column width, will be displayed in the same width as the column.
	When executing Modify function to change the Dddw.property on the Web, the value variable must be contained in single quotation marks. For example,
	Unsupported code:
	`dw_1.Modify (""emp_id.dddw.name = dw_dddw_dropdown"")`
	Supported code:
	`dw_1.Modify (""emp_id.dddw.name = 'dw_dddw_dropdown'"")`
		It is unsupported to dynamically change the edit style of a column by modifying this property.

ddlb.*property*	(r/w): AllowEdit, AutoHScroll, Case, Limit, NilIsNull, Required, ShowList, Sorted, UseAsBorder, VScrollBar	It is unsupported to dynamically change the edit style of a column by modifying this property.
Edit.*property*	(r/w): AutoHScroll, AutoSelect, AutoVScroll, Case, CodeTable, DisplayOnly, Format, HscrollBar, Limit, Name, NilIsNull, Password, Required, Style, VscrollBar, FocusRectangle, ValidateCode	Modifying ValidateCode takes no effect. It is unsupported to dynamically change the edit style of a column by modifying this property.
EditMask.*property*	(r/w): AutoSkip, CodeTable, Mask, ReadOnly, Required, Spin, SpinIncr, SpinRange, UseFormat, FocusRectangle In the Web application, if the Mask of a column is set to [General], the values in the column are all	If a Mask is set as a four-digit string, the column will only retrieve the first four digits of the entered value when you use SetItem to set a value in the EditMask column which is different from what occurs in PowerBuilder.

Property	What's Supported	What's Unsupported
	displayed as integers when the column gets focus. In the Web application, if the Mask is set to General by default, the displaying data of a column with the EditMask edit style will be converted into an integer when the column gets focus.	It is unsupported to dynamically change the edit style of a column by modifying this property.
Font.*property*	(r/w): Charset, Escapement, Face, Family, Height, Italic, Pitch, Strikethrough, Underline, Weight Note: 1. When the value of Escapement is a negative number, the text will not be rotated. 2. The text rotated can be displayed out of the control in PowerBuilder, but not in the Web application. 3. After rotated, the multiple-line text still displays in multiple lines in PowerBuilder, but displays in one line on Web.	Width
Format	Edit	
Height	Set the Height property using a DataWindow expression.	
Height.AutoSize	Supported (r)	
HideSnaked	Supported (r)	
Identity	Fully supported (r/w)	
ID	Supported (r)	
Initial	Fully supported (r/w)	
Key	Fully supported (r/w)	
LineRemove	Supported (r)	
Multiline	Fully supported (r/w)	
Name	Supported (r)	
Pointer	Supported (r/w)	
Protect	This property can be set by using DataWindow expression.	

Property	What's Supported	What's Unsupported
RadioButtons .property	Fully supported (r/w): 3D, Columns, LeftText, Scale	It is unsupported to dynamically change the edit style of a column by modifying this property.
RightToLeft	Supported This property is partially supported. It only supports the right-to-left display of characters.	
SlideLeft	Supported (r/w)	
SlideUp	Supported (r/w)	
TabSequence	Fully supported (r/w)	
Tag	Fully supported (r/w)	
Type	Supported (r)	
Update	Fully supported (r/w)	
Validation	Supported (r)	This property cannot be dynamically changed.
ValidationMsg	Supported (r/w)	
Values	Supported (r/w) This property is supported for columns with the following edit styles: DropDownListBox, CheckBox, and RadioButtons.	
Visible	Supported (r/w) This property can be set by using DataWindow expression.	
Width	Fully supported (r/w) This property can be set by using DataWindow expression.	
Width.Autosize	Supported (r)	
X	Supported (r/w)	
Y	Supported (r/w)	

Unsupported

- Movable
- Resizable
- HTML.*property*

Notes:

1. Automatically adjusting the height of DropDownDataWindow and DropDownListBox objects in Column controls is unsupported.
2. Make sure that the length of data you specified for Column control in PowerBuilder exactly matches the length of data of the corresponding column in the database.
3. It is unsupported to nest a DropDownDataWindow with another DropDownDataWindow.

11.7.3.3 Computed Field control

Supported

If a property is marked as "supported (r)" or "(r)", it means that the property can be read - but not changed - in script, and properties set in the painter will not take effect in the converted application.

If a property is marked as "supported (r/w)" or "(r/w)", it means that the property can be read or changed in script, and properties set in the painter will take effect.

LineRemove Supported (r)

Table 11.16:

Property	What's Supported	What's Unsupported
Alignment	Fully supported (r/w): Left!, Center!, Right!, Justify!	None
Attributes	Supported (r)	
Background .property	Fully supported (r/w): Color, Mode	
Band	Fully supported	
Border	(r/w): NoBorder!, Box!, Lowered!, Raised!, ShadowBox!, Underline!, ResizeBorder!	
Color	Fully supported (r/w) This property can be set by using DataWindow expression.	
ColType	Supported (r)	
Expression	Supported (r/w) Note: 1. After dynamically modifying this property in CrossTab DataWindow, data will be recalculated according to the new expression on the Web. In PowerBuilder, the data will not be recalculated. 2. If the expression starts with spaces (for example, ' ' +	

Property	What's Supported	What's Unsupported
	mail_from), in the Web application the computed field value after the spaces always starts in a new line, despite the column height. This is different from PowerBuilder.	
Font.property	(r/w): Charset, Escapement, Face, Family, Height, Italic, Pitch, Strikethrough, Underline, Weight Note: 1. When the value of Escapement is a negative number, the text will not be rotated. 2. The text rotated can be displayed out of the control in PowerBuilder, but not in Web applications. 3. After rotated, the multiple-line text still displays in multiple lines in PowerBuilder, but displays in one line on Web.	Width
Format	Supported (r/w)	
Height	Fully supported (r/w) This property can be set by using DataWindow expression.	
Height.Autosize	Supported	
HideSnaked	Supported (r)	
HTML.*property*	(r): AppendedHTML, Link, LinkArgs, LinkTarget, ValueIsHTML	
Multiline	Supported	
Name	Supported	
Pointer	Supported	
SlideLeft	Supported	
SlideUp	Supported	
Tag	Supported	
Type	Supported	
Visible	Supported	

Property	What's Supported	What's Unsupported
Width	Supported	
Width.Autosize	Supported	
X	Supported	
Y	Supported	

Unsupported

- Movable
- Resizable

11.7.3.4 Graph control

Supported

If a property is marked as "supported (r)" or "(r)", it means that the property can be read - but not changed - in script, and properties set in the painter will not take effect in the converted application.

If a property is marked as "supported (r/w)" or "(r/w)", it means that the property can be read or changed in script, and properties set in the painter will take effect.

Table 11.17:

Property	What's Supported	What's Unsupported
Attributes	Fully supported	
Axis	Fully supported	
Axis.property	Fully supported	
Axis.DispAttr	Fully supported	
BackColor	Fully supported	
Border	Fully supported	
Category	Fully supported	
Category.property	Fully supported	
Category.DispAttr	Fully supported	
Color	Fully supported	
Depth	Fully supported	
Elevation	Fully supported	
GraphType	(r/w): AreaGraph!, BarGraph!, BarStackGraph!, Bar3DObjGraph!, BarStack3DObjGraph!, Col3DObjGraph!, ColStack3DObjGraph!, ColGraph!,	Area3D!, Bar3DGraph!, Col3DGraph!, Line3D!, Pie3D!

Property	What's Supported	What's Unsupported
	ColStackGraph!, LineGraph!, PieGraph!, ScatterGraph!	
Height	Fully supported	
HideSnaked	Fully supported	
Legend	Fully supported	
Legend.DispAttr.property	Fully supported	
Name	Fully supported	
OverlapPercent	Fully supported	
Perspective	Fully supported	
Pie.DispAttr.fontproperty	Fully supported	
Pointer	Fully supported	
Range	Fully supported	
Rotation	Fully supported	
Series	Fully supported	
Series.property	Fully supported	
Series.DispAttr	Fully supported	
ShadeColor	Fully supported	
SizeToDisplay	Fully supported	
SlideLeft	Fully supported	
SlideUp	Fully supported	
Spacing	Fully supported	
Tag	Fully supported	
Title	Fully supported	
Title.DispAttr.fontproperty	Fully supported	
Type	Fully supported	
Values	Fully supported	
Values.property	Fully supported	
Values.DispAttr	Fully supported	
Visible	Fully supported	
Width	Fully supported	
X	Fully supported	
Y	Fully supported	

Unsupported

- Movable

- Resizable

11.7.3.5 GroupBox control

Supported

If a property is marked as "supported (r)" or "(r)", it means that the property can be read - but not changed - in script, and properties set in the painter will not take effect in the converted application.

If a property is marked as "supported (r/w)" or "(r/w)", it means that the property can be read or changed in script, and properties set in the painter will take effect.

Table 11.18:

Property	What's Supported	What's Unsupported
Attributes	Supported (r)	
Background.*property*	Fully supported (r/w): Color, Mode	
Band	Supported (r)	
Border	Supported (r)	
Color	Fully supported (r/w)	
Font.*property*	(r/w): Charset, Escapement, Face, Family, Height, Italic, Pitch, Strikethrough, Underline, Weight Note: 1. When the value of Escapement is a negative number, the text will not be rotated. 2. The text rotated can be displayed out of the control in PowerBuilder, but not in Web applications. 3. After rotated, the multiple-line text still displays in multiple lines in PowerBuilder, but displays in one line on Web.	Width
Height	Fully supported (r/w)	
HideSnaked	Supported (r)	
Name	Supported (r)	
Pointer	Fully supported (r/w):	
SlideLeft	Supported (r/w):	
SlideUp	Supported (r/w):	

Property	What's Supported	What's Unsupported
Tag	Fully supported (r/w)	
Text	Fully supported (r/w)	
Type	Supported (r)	
Visible	Fully supported (r/w)	
Width	Fully supported (r/w)	
X	Fully supported (r/w)	
Y	Fully supported (r/w)	

Unsupported

- Movable
- Resizable

11.7.3.6 Line control

Supported

If a property is marked as "supported (r)" or "(r)", it means that the property can be read - but not changed - in script, and properties set in the painter will not take effect in the converted application.

If a property is marked as "supported (r/w)" or "(r/w)", it means that the property can be read or changed in script, and properties set in the painter will take effect.

Table 11.19:

Property	What's Supported	What's Unsupported
Attributes	Supported (r)	
Background *.property*	(r/w): Color, Mode	
Band	Supported (r)	
Name	Supported (r)	
Pen.*property*	Fully supported (r/w)	
Pointer	Fully supported (r/w)	
SlideLeft	Fully supported (r/w)	
SlideUp	Fully supported (r/w)	
Tag	Fully supported (r/w)	
Type	Supported	
Visible	Fully supported (r/w)	
X1, X2	Fully supported (r/w)	

Property	What's Supported	What's Unsupported
Y1, Y2	Fully supported (r/w)	

Unsupported

- Movable
- Resizable

11.7.3.7 Oval control

Supported

If a property is marked as "supported (r)" or "(r)", it means that the property can be read - but not changed - in script, and properties set in the painter will not take effect in the converted application.

If a property is marked as "supported (r/w)" or "(r/w)", it means that the property can be read or changed in script, and properties set in the painter will take effect.

Table 11.20:

Property	What's Supported	What's Unsupported
Attributes	Supported(r)	
Background *.property*	Fully supported (r/w): Color, Mode	
Band	Supported (r)	
Brush.*property*	(r/w): Color, Hatch	
Height	Supported (r/w)	
HideSnaked	Supported (r)	
Name	Fully supported	
Pen.*property*	(r/w): Color, Style, Width	
Pointer	Fully supported (r/w)	
SlideLeft	Supported (r/w)	
SlideUp	Supported (r/w)	
Tag	Fully supported (r/w)	
Type	Supported (r)	
Visible	Fully supported (r/w)	
X	Fully supported (r/w)	
Y	Fully supported (r/w)	

Unsupported

- Movable
- Resizable

11.7.3.8 Picture control

Supported

If a property is marked as "supported (r)" or "(r)", it means that the property can be read - but not changed - in script, and properties set in the painter will not take effect in the converted application.

If a property is marked as "supported (r/w)" or "(r/w)", it means that the property can be read or changed in script, and properties set in the painter will take effect.

Table 11.21:

Property	What's Supported	What's Unsupported
Attributes	Supported(r)	
Band	Supported (r)	
Border	Fully supported (r/w)	
Filename	Fully supported (r/w)	
Height	Fully supported (r/w)	
HideSnaked	Supported (r)	
HTML.*property*	(r): AppendedHTML, Link, LinkArgs, LinkTarget	
Invert	Supported (r)	
Name	Supported (r)	
Pointer	Supported (r/w):	
SlideLeft	Supported (r/w)	
SlideUp	Supported (r/w)	
Tag	Fully supported (r/w)	
Type	Supported (r)	
Visible	Fully supported (r/w)	
Width	Fully supported (r/w)	
X	Fully supported (r/w)	
Y	Fully supported (r/w)	

Unsupported

- Movable
- Resizable

11.7.3.9 Rectangle control

Supported

If a property is marked as "supported (r)" or "(r)", it means that the property can be read - but not changed - in script, and properties set in the painter will not take effect in the converted application.

If a property is marked as "supported (r/w)" or "(r/w)", it means that the property can be read or changed in script, and properties set in the painter will take effect.

Table 11.22:

Property	What's Supported	What's Unsupported
Attributes	Supported (r)	
Background *.property*	Fully supported (r/w): Color, Mode	
Band	Supported (r)	
Brush.*property*	(r/w): Color, Hatch	
Height	Fully supported (r/w)	
HideSnaked	Supported (r)	
Name	Supported (r)	
Pen.*property*	(r/w): Color, Style, Width	
Pointer	Fully supported (r/w)	
SlideLeft	Supported (r/w)	
SlideUp	Supported (r/w)	
Tag	Fully supported (r/w)	
Type	Supported (r)	
Visible	Fully supported (r/w)	
X	Fully supported (r/w)	
Y	Fully supported (r/w)	

Unsupported

- Movable
- Resizable

11.7.3.10 Report control

Supported

If a property is marked as "supported (r)" or "(r)", it means that the property can be read - but not changed - in script, and properties set in the painter will not take effect in the converted application.

If a property is marked as "supported (r/w)" or "(r/w)", it means that the property can be read or changed in script, and properties set in the painter will take effect.

Table 11.23:

Property	What's Supported	What's Unsupported
Attributes	Fully supported	

Property	What's Supported	What's Unsupported
Band	Fully supported	
Border	Fully supported	
DataObject	Supported (r/w)	DataWindows with the following presentation styles cannot be the DataObject: RichText, OLE and TreeView.
Height	Fully supported	
Name	Fully supported	
Nested Arguments	Fully supported	
NewPage	Supported	
Pointer	Fully supported	
Slideleft	Supported	
SlideUp	Supported	
Tag	Fully supported	
Trail footer	Supported	
Type	Fully supported	
Visible	Fully supported	
X	Fully supported	
Y	Fully supported	

Unsupported

- Criteria
- HideSnaked
- Movable
- Resizable

Note:

- In a single data retrieval, the number of report queries cannot be over 65,535.
- It is unsupported to assign a nested report to a DataWindow object.

  Unsupported example:

```
Dwobject ldwo
String  ls_1
Ldwo = Dw_1.object.dw_report[1]  // dw_report is a nested report.
Ls_1 = ldwo.object.lastname[2]
```

  Supported example:

```
Ls_1 = Dw_1.object.dw_report[1].object.lastname[2]
```

11.7.3.11 RoundRectangle control

Supported

If a property is marked as "supported (r)" or "(r)", it means that the property can be read - but not changed - in script, and properties set in the painter will not take effect in the converted application.

If a property is marked as "supported (r/w)" or "(r/w)", it means that the property can be read or changed in script, and properties set in the painter will take effect.

Table 11.24:

Property	What's Supported	What's Unsupported
Attributes	Supported (r)	
Background *.property*	(r/w): Color, Mode	
Band	Supported (r)	
Brush.*property*	(r/w): Color, Hatch	
Height	Fully supported(r/w)	
HideSnaked	Supported (r)	
Name	Supported (r)	
Pen.*property*	(r/w): Color, Style, Width	
Pointer	Fully supported(r/w)	
SlideLeft	Supported (r/w)	
SlideUp	Supported (r/w)	
Tag	Fully supported (r/w)	
Type	Supported (r)	
Visible	Fully supported (r/w)	
X	Fully supported (r/w)	
Y	Fully supported (r/w)	

Unsupported

- Movable
- Resizable

11.7.3.12 Text control

Supported

If a property is marked as "supported (r)" or "(r)", it means that the property can be read - but not changed - in script, and properties set in the painter will not take effect in the converted application.

If a property is marked as "supported (r/w)" or "(r/w)", it means that the property can be read or changed in script, and properties set in the painter will take effect.

Table 11.25:

Property	What's Supported	What's Unsupported
Alignment	Fully supported (r/w): Left!, Center!, Right!, Justify!	None
Attributes	Supported (r)	
Background *.property*	Fully supported (r/w): Color, Mode	
Band	Supported (r): Band	Background, Foreground
Border	(r/w): NoBorder!, Box!, Lowered!, Raised!, ShadowBox!, Underline!, ResizeBorder!	
Color	Supported (r/w) This property can be set by using DataWindow expression.	
Font.*property*	(r/w): Charset, Escapement, Face, Family, Height, Italic, Pitch, Strikethrough, Underline, Weight Note: 1. When the value of Escapement is a negative number, the text will not be rotated. 2. The text rotated can be displayed out of the control in PowerBuilder, but not in Web applications. 3. After rotated, the multiple-line text still displays in multiple lines in PowerBuilder, but displays in one line on Web.	Width
Height	Fully supported (r/w) s This property can be set by using DataWindow expression.	
HideSnaked	Supported (r)	
HTML.*property*	(r): AppendedHTML, Link, LinkArgs, LinkTarget, ValueIsHTML	
Name	Supported (r)	
Pointer	Fully supported(r/w)	

Property	What's Supported	What's Unsupported
SlideLeft	Supported (r/w)	
SlideUp	Supported (r/w)	
Tag	Fully supported (r/w)	
Text	Supported (r/w)	
Type	Supported (r)	
Visible	Fully supported (r/w) This property can be set by using DataWindow expression.	
Width	Fully supported (r/w) This property can be set by using DataWindow expression.	
X	Fully supported (r/w) This property can be set by using DataWindow expression.	
Y	Fully supported (r/w) This property can be set by using DataWindow expression.	

Unsupported

- Height.AutoSize
- Movable
- Resizable

11.7.3.13 Unsupported controls

The properties of all the following unsupported controls are unsupported:

- OLE
- TableBlob

11.8 DataWindow data and property expressions

Supported

- The following DataWindow data expressions are supported:
  1. Syntax for one or all data items in a named column

     ```
     dwcontrol.Object.dwcolumnname {.buffer} {.datasource}{[rownum]}
     ```

  2. Syntax for selected data in a named column

     ```
     dwcontrol.Object.dwcolumnname {.Primary}{.datasource}.Selected
     ```

3. Syntax for a range of data in a named column

```
dwcontrol.Object.columnname{.buffer}{.datasource}[startrow,endrow]
```

4. Syntax for a single data item in a DataWindow

```
dwcontrol.Object.Data {.buffer}{.datasource}[rownum, colnum]
```

5. Syntax for data in a block of rows and columns

```
dwcontrol.Object.Data {.buffer}{.datasource}[startrow,startcol, endrow,
 endcol]
```

6. Syntax for data in a single row or all rows

```
dwcontrol.Object.Data {.buffer}{.datasource}{[rownum]}
```

7. Syntax for all data from selected rows

```
dwcontrol.Object.Data {.Primary}{.datasource}.Selected
```

- The expression for accessing the text displayed in the column header is supported:

```
dwcontrol.Object.ColumnName.Text
```

- Basic syntax for DataWindow property expressions is supported:

```
dwcontrol.Object.dwcontrolname{.property}.property{=value}
```

- Setting or getting the value of a dot notation is supported, except that there is structure data in the expression.

Unsupported

- Getting the data value from a filter buffer according to the row number is unsupported. This is because data is stored in a different order in Appeon than in PowerBuilder. For example, the following script is unsupported:

```
dw_control.object.filter[startrow, startcol, endrow, endcol] //unsupported
```

- Setting the value of a dot notation to a structure array, or setting a structure array to a dot notation, is unsupported. For example, the following syntax is unsupported:

```
struct1 t[10]
t = dw_1.object.data[1]
```

  Note: The Appeon Unsupported Features Analysis tool cannot detect unsupported features in DataWindow dot notations. You must make sure the specifications about dot notations are satisfied in the application. Otherwise, it will cause errors. For more information on undetected features, refer to the Undetected Features section.

- Selected and Data are unsupported to be directly accessed by a DataWindow object via the dot notation. For example:

```
ldwo_target.Primary =
ldwo_column.selected //Unsupported
ldwo_target.Primary =
dw_1.object.id.selected //Supported
```

- It is unsupported to contain Data in a dot notation to obtain the data of a column. For example:

  Supported example:

```
string ls_an
dwobject dwo
dwo = dw_1.object.id
ls_an = string(dwo.data.primary[1])
```

  Unsupported example:

```
string ls_an
dwobject dwo
dwo = dw_1.object.id
ls_an = string(dwo.primary[1])
```

- It is unsupported to use the following syntax in obtaining data in a row of a report.

```
dw_1.Object.dw_report[1] //Unsupported
```

11.9 DataWindow constants

Supported

In PowerBuilder, constants are defined in the DataWindow control for values of properties and arguments for methods. They are sets of values associated with enumerated data types. Values for enumerated data types always end with an exclamation point. The following list is the PowerBuilder constants Appeon supports:

Table 11.26:

Constant	Supported Values	Unsupported Values
Alignment	Left! Center! Right! Justify!	None
Band	Detail! Header! Footer!	None
Border	NoBorder! Box! Lowered! Raised!	ShadowBox! Underline! ResizeBorder!
BorderStyle	StyleBox! StyleLowered! StyleRaised!	StyleShadowBox!
CharSet	CharSetAnsi! CharSetUnicode!	CharSetDBCS-Japanese!

Constant	Supported Values	Unsupported Values
	CharSetAnsiHebrew! CharSetAnsiArabic!	
DWBuffer	Primary! Delete! Filter!	None
DWItemStatus	NotModified! DataModified! New! NewModified!	None Note: It is not recommended to change status in script. Frequent or inappropriate user modification may result in unpredictable problems. The following operation is not recommended for new rows with no specified values for their columns: Changing a row's status from New! to NewModified and updating in script.
SaveAsType	Excel! Excel5! HTMLTable! PDF! Text! WMF! XML! Note: Up to 256 characters are supported and anything larger will be automatically truncated.	CSV! SYLK! WKS! WK1! DIF! dBASE2! dBASE3! SQLInsert! Clipboard! PSReport! XSLFO! Excel8! EMF!
SQLPreview Function	PreviewFunctionRetrieve!	PreviewFunctionReselectRow! PreviewFunctionUpdate!
SQLPreviewType	PreviewSelect!	PreviewInsert! PreviewDelete! PreviewUpdate!

If a constant is set to an unsupported value, Appeon will use the default enumerated value as the constant value. For example, unsupported BorderStyle will be read as NoBorder.

Unsupported

The following constants are not supported:

Table 11.27:

ConnectionSource	DriverType	DWConflictResolution	FillPattern
grColorType	grDataType	grObjectType	grSymbolType
LineStyle	RowFocusInd		

11.10 DataWindow control

11.10.1 Properties of DataWindow control

Supported

The supported properties of DataWindow controls are listed in the table below:

Table 11.28:

Border	BorderStyle*	BringToTop	ControlMenu
DataObject	DragAuto	DragIcon	Enabled
Height	HScrollBar*	LiveScroll	Icon
MaxBox	MinBox	Object	Resizable*
RightToLeft	TabOrder	Tag	Title
TitleBar	Visible	VScrollBar*	Width
X	Y		

- BorderStyle

  Three types of BorderStyle are supported: Box, 3D Raised, and 3D Lowered. Shadow Box is unsupported.

- HScrollBar

  In PowerBuilder, when the property VScrollBar or HScrollBar is changed from visible to invisible or from invisible to visible, the Resize event will be triggered. However, in the same scenario on the Web, the Resize event will not be triggered.

- Resizable

  The Resizable property cannot be dynamically changed.

  After setting this property to True, setting Border properties will not be effective.

- VScrollBar

  In PowerBuilder, when the property VScrollBar or HScrollBar is changed from visible to invisible or from invisible to visible, the Resize event will be triggered. However, in the same scenario on the Web, the Resize event will not be triggered.

Unsupported

The unsupported property of DataWindow control is ClassDefinition and HSplitScroll.

11.10.2 Events for the DataWindow control

Table 11.29:

Events for the DataWindow control	**Supported**	
	ButtonClicked	ButtonClicking
	Clicked	Constructor
	Collapsed	Collapsing
	DBError	Destructor
	DoubleClicked	DragDrop
	DragEnter	DragLeave
	DragWithin	EditChanged
	Expanded	Expanding
	GetFocus	ItemChanged
	ItemError	ItemFocusChanged
	LoseFocus	PrintPage
	RButtonDown	Resize
	RetrieveEnd	RetrieveStart
	RowFocusChanged	RowFocusChanging
	SQLPreview	ScrollHorizontal
	ScrollVertical	WSError
	DBError The sqlsyntax and row arguments do not work in the DBError event. The row argument is always considered to be 0. The GetSQLPreview function will be ignored if used in the DBError event. *SQLPreview* Only the value previewselect! can be used as the SQLPreview sqltype argument.	
	Unsupported	
	Error	Help
	Other	PrintEnd
	PrintStart	RetrieveRow
	Using instance variables or global variables to store the dwo argument (argument type is dwobject) for a DataWindow event is unsupported.	
Triggering of DataWindow events	**Supported**	
	Nearly full support is available for Triggering supported DataWindow events, except for the following limitations and differences.	

	(For **ItemChanged** event) In a Web application, if the ItemChanged event is triggered for a field whose new value does not pass validation, the focus will not go back to the column being edited. (For **Resize** event) When the Width or Height of a DataWindow control is reset, the Resize event will be triggered, regardless of whether the value is changed or remains the same. (For **RowFocusChanged** event) In PowerBuilder, RowFocusChanged will be triggered twice upon execution of the RowsCopy or RowsMove function. However, in the same scenario on the Web, the RowsFocusChanged event will only be triggered once. (For **SQLPreview** event) The SQLPreview event can only be triggered by the Retrieve function, not by Update or ReselectRow. If a DataWindow event requires interaction with the server, its subsequent event may not be triggered due to the time delay caused by the interaction with the server. For label DataWindows, clicking different lines cannot trigger the RowFocusChanging and RowFocusChanged events.

11.10.3 Functions of DataWindow control

Supported

There are some limitations or differences for calling functions marked with the "*" symbol. Refer to the Limitations and Differences for Calling Functions.

Table 11.30:

AcceptText	CategoryCount	CategoryName
ClassName	Clear	ClearValues
ClipBoard	Copy	CopyRTF
Collapse	CollapseAll	CollapseAllChildren
CollapseLevel	Cut	DataCount
DeletedCount	DeleteRow	Describe*
Drag	Expand	ExpandAll
ExpandAllChildren	ExpandLevel	Filter
FilteredCount	Find	FindCategory
FindGroupChange	FindNext	FindSeries
GetBandAtPointer	GetBorderStyle	GetChanges
GetChild	GetClickedColumn	GetClickedRow*
GetColumn	GetColumnName	GetData
GetDataPieExplode	GetDataStyle	GetDataValue
GetFormat	GetFullState*	GetItemDate
GetItemDataTime	GetItemDecimal	GetItemFormattedString

GetItemUnformattedString	GetItemNumber	GetItemStatus
GetItemString	GetItemTime	GetNextModified
GetObjectAtPointer	GetParent	GetRow
GetRowFromRowId	GetSelectedRow	GetSeriesStyle
GetSQLPreview	GetSQLSelect*	GetText
GetValidate	GetValue*	GroupCalc
Hide	ImportClipboard*	ImportFile*
InsertDocument	InsertRow	ImportString*
IsExpanded	IsSelected	LineCount
ModifiedCount	Modify*	ObjectAtPointer
Paste	PasteRTF	Position
PointerX	PointerY	PostEvent
Print*	PrintCancel	Reset*
ResetDataColors	ResetUpdate	Resize
Retrieve*	RowCount	RowsCopy*
RowsDiscard	RowsMove	SaveAs*
SaveAsAscii	ScrollNextPage	ScrollNextRow
ScrollPriorPage	ScrollPriorRow	ScrollToRow
SelectedLength	SelectedStart	SelectedText
SelectedLine	SelectRow	SelectText
SelectTextLine	SelectTextAll*	SelectTextWors
SetChanges	SeriesCount	SeriesName
SetBorderStyle	SetColumn	SetDataPieExplode
SetDataStyle	SetDetailHeight*	SetFilter
SetFocus	SetFormat	SetFullState
SetItem	SetItemStatus	SetPosition*
SetRedraw	SetRow	SetRowFocusIndicator
SetSeriesStyle	SetSort*	SetSQLPreview
SetSQLSelect*	SetTabOrder	SetText
SetTransObject	SetValidate	SetValue*
SetWSObject*	ShareData	ShowHeadFoot
ShareDataOff	Show	Sort*
TextLine	TriggerEvent*	Typeof
Undo	Update	

Unsupported

Table 11.31:

CanUnDo	CrosstabDialog	DBCancel
DBErrorCode	DBErrorMessage	GenerateHTMLForm
GenerateResultSet	GetContextService	GetMessageText
GetStateStatus	GetTrans	GetUpdateStatus
OLEActivate	ResetInk	ReselectRow
SaveInk	SaveInkPic	Scroll
ResetTransObject	SetActionCode	SetHTMLAction
SetTrans		

Limitations and Differences for Calling Functions

1. Limitations

- GetFullState

  The Retrieve argument is unsupported in the GetFullState function.

- GetSQLSelect

  If the syntax in DataWindow is PBSELECT statement, when calling the GetSQLSelect function, it will still return the PBSELECT statement but not the SQL SELECT statement, which is different from that in PowerBuilder.

- GetValue

  The *column* argument only works when the edit style of the column is CheckBox, DropDownListBox, Edit, or RadioButton.

- ImportFile, ImportString, ImportClipboard

  1. The *filename* argument of ImportFile function must be a tab-separated file (TXT) or a comma-separated file (CSV).

  2. The *importtype* arguments (e.g. Text!, CSV! and XML!) of ImportString function are unsupported.

     The *importtype* arguments (e.g. Text!, CSV!, XML!, DBase2!, DBase3!) of ImportFile function are unsupported.

  3. When using these functions, you should ensure that the data type imported matches the data type that you specified in a DataWindow control, DataStore object or graph control. Otherwise the output on the Web may be different from that in PowerBuilder. For example, if the date type specified in a DataWindow is string, when you import a file with date number (e.g., 2006-08-01), on the Web the date data will be recognized as a string (2006), however in PowerBuilder, it will be recognized as a date data (2006-08-01).

- Modify

Refer to Dynamic DataWindow.

- Print

  1. Arguments specified in the Print function will be ignored in a Web application. Even if theCancelDialog argument is set to TRUE or FALSE, no Cancel dialog will be displayed on the Web to cancel printing.

  2. When the Print function is executed for a DataWindow that contains a Text control with BorderType as None and actual length of the content larger than the length of the control, the printing result shows the content in the Text control the same way as the control is displayed in the application, although in PowerBuilder, the printing result shows the full content in the Text control.

  3. For controls, print function is only supported when called in Graph controls with the following syntax.

     integer objectname.Print(long printjobnumber, integer x, integer y{, integer width, integer height})

- Reset

  The unsupported syntax: Reset(dwcontrol).

  The supported syntax: dwcontrol.Reset.

- SaveAs

  1. On the Web, Nillable attribute is unsupported in an XML schema.

  2. The supported SaveAs syntax :

     ```
     ll_testvalue = dw_test.SaveAs(filename, saveastype, colheading, encoding)
     ```

     In the syntax:

     The value of a *saveastype* argument can be TEXT, HTMLTable, WMF (EAServer only), XML, PDF*, EXCEL, or EXCEL5. For the EXCEL format, only up to 256 characters are supported and anything larger will be automatically truncated.

     When *saveastype* is PDF, the *colheading* argument is unsupported, for example, the following syntax is unsupported:

     ```
     Integer dwcontrol.SaveAs({string filename, saveastype saveastype, boolean
      colheading{, encoding encoding}})
     ```

     The unsupported SaveAs syntax:

     ```
     dw_test.SaveAs([filename, ]graphcontrol[, saveastype, colheading])
     ```

  3. If the DataWindow without any data is saved as an HTML file or XML file, on Web the DataWindow header will not be saved, whereas it will be saved on PB.

- SetPosition

  The supported syntax:

```
dwcontrol.SetPosition (position {, precedingobject})
```

The *position* can be ToTop! and ToBottom!, but cannot be Behind!

The unsupported syntax:

```
dwcontrol.SetPosition(objectname, band, bringtofront)
```

- SetSQLSelect

  If a DataWindow uses a stored procedure as its data source, the SetSQLSelect function cannot be executed for the DataWindow.

- SetValue

  The SetValue *column* argument only works if the edit style of the column is DropDownListBox.

- SetSort

  1. If the Format argument of the SetSort contains sorting criteria for more than one column, the criteria of the columns must be separated with commas.
  2. Using desc or asc as the Order will not have any effect. Please use the standard Order value A for ascending or D for descending order.

- SetWSObject

  The SetWSObject function is used to set the authentication info for accessing Web service.

- TriggerEvent

  The following TriggerEvent syntax is unsupported:

```
object1.TriggerEvent (object2, event)
```

2. Differences

- Describe

  If there is no value for a property in the propertylist argument, Describe returns a question mark (?) in PowerBuilder, but returns the default value (for example, Arrow!) of the property on the Web.

- GetClickedRow

  In a Web application, if the user clicks or double-clicks any area within a DataWindow, the GetClickedRow function returns the same value as in PowerBuilder. If the GetClickedRow is called in some control or window for a DataWindow, the return value is different on the Web than in PowerBuilder. For example, if the *dw_1.GetClickedRow()* function is called in the Clicked event of a CommandButton, it returns -1 in PowerBuilder and 0 on the Web.

- GetItemDecimal

  The return value of the *GetItemDecimal* function will omit the zero(s) to the right of the decimal value on the Web. For example: 3000.100 will be returned as 3000.1 on the Web but 3000.100 in PowerBuilder.

- Retrieve

  1. Because Appeon Web applications cannot check whether the retrieve result set matches the DataWindow definition syntax, the DBError event will not be triggered.

  2. If the data type of an EditMask control is numeric, the value of the control cannot be set to "0".

- SetDetailHight

  Setting detail.height.autosize to true after specifying the height by this function, executing Describe returns the auto sized height on the Web. But in PowerBuilder it returns the height specified by this function.

- RowDiscard

  For Composite, Group, Label and RichText DataWindows, if the current row is the start row, calling RowDiscard will not trigger RowFocusChanged on the Web. This is different from that in PowerBuilder.

- SaveAsAscii

  When executing the SaveAsAscii function in PowerBuilder, invisible controls which lay in rows will not be saved as empty rows or " " (quotation marks). However, these invisible controls will be ignored on the Web.

- SelectTextAll

  If the focus is in the header or footer band, executing SelectTextAll (detail!) on the Web returns -1 and no text will be selected. However, in PowerBuilder, the text in the band, which the focus is in, will be selected.

- ShareData and ShareDataOff

  1. Supported:

     ```
     CONNECT USING SQLCA;
     dw_corp.SetTransObject(SQLCA)
     dw_corp.Retrieve()
     dw_corp.ShareData(dw_emp)
     dw_corp.ShareData(dw_dept)
     ... // Some processing
     dw_emp.ShareDataOff()
     ```

  2. In Web applications, if any data in two data-sharing Group DataWindows are changed, the primary Group DataWindow will be re-grouped and re-sorted automatically, while the secondary Group DataWindow will only be re-grouped.

  3. The row and column status is shared between the primary DataWindow and secondary DataWindow on the Web, although not shared in PowerBuilder.

- Sort

1. Different from the Sort in PowerBuilder, the Web Sort function considers the "~" symbol to have precedence over characters. For this reason, the sort result may be a little different in Web and PowerBuilder applications.

2. In the Web application, when the user sorts a DataWindow on a specified column, rows containing special characters (e.g. ".", "", "/") will be sorted in different order than in PowerBuilder.

11.11 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, PowerBuilder provides some events and functions for DataStores that pertain to the visual presentation of the data.

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.

11.11.1 Properties of DataStore object

Supported

The supported properties of DataStore are DataObject and Object.

Unsupported

The unsupported property of DataStore is ClassDefinition.

11.11.2 Events of DataStore object

Supported

The following DataStore events are supported:

Table 11.32:

Constructor	DBError	Destructor*	ItemChanged
ItemError	PrintPage	RetrieveEnd	RetrieveStart
SQLPreview*	UpdateEnd	UpdateStart	WSError

- Destructor

  If a non-visual object is a local variable, the Destructor event in the non-visual object cannot be triggered unless there is a Destroy statement for the non-visual object as well.

- SQLPreview

  The SQLPreview event can be triggered by Retrieve method, but cannot be triggered by the Update or ReselectRow methods.

Unsupported

The following DataStore events are unsupported:

- Error
- PrintEnd
- PrintStart
- RetrieveRow

11.11.3 Functions of DataStore object

Supported

There are some limitations or differences for calling functions marked with the "*" symbol. Refer to the Limitations and Differences for Calling Functions.

Table 11.33:

AcceptText	ClassName	ClearValues
Create	DeletedCount	DeleteRow
Describe	FindGroupChange	Filter
FilteredCount	Find	FindRequired
GenerateHTMLForm	GetChanges	GetChild
GetColumn	GetColumnName	GetFormat
GetFullState	GetItemDate	GetItemDateTime
GetItemDecimal	GetItemNumber	GetItemStatus
GetItemString	GetItemTime	GetNextModified
GetParent	GetRow	GetRowFromRowId
GetRowIdFromRow	GetSelectedRow	GetSQLSelect*
GetText	GetValidate	GetValue*
GroupCalc	ImportClipboard*	ImportFile*
ImportString*	InsertRow	IsSelected
ModifiedCount	Modify*	PostEvent
Print	Reset	ResetUpdate
Retrieve	RowCount	RowsCopy
RowsDiscard	RowsMove*	SaveAs*
SaveAsAscii	SetDetailHeight	SelectRow
SetChanges	SetColumn	SetFilter
SetFormat	SetFullState	SetItem
SetItemStatus	SetPosition	SetRow
SetSort*	SetSQLPreview	SetSQLSelect*
SetText	SetTransObject	SetValidate
SetValue*	SetWSObject*	ShareData
ShareDataOff	Sort	TriggerEvent*

TypeOf	Update	

Unsupported

Table 11.34:

CategoryCount	CategoryName	Clipboard
CopyRTF	CreateFrom	DataCount
DBCancel	FindCategory	FindSeries
GenerateResultSet	GetBorderStyle	GetContextService
GetClickedColumn	GetClickedRow	GetData
GetDataPieExplode	GetDataStyle	GetDateValue
GetSeriesStyle	GetStateStatus	GetTrans
InsertDocument	PasteRTF	PrintCancel
ReselectRow	ResetDataColors	ResetTransObject
SeriesCount	SeriesName	SetBorderStyle
SetDataPieExplode	SetDataStyle	SetHTMLAction
SetSeriesStyle	SetTrans	

Limitations and Differences for Calling Functions

- GetValue

  The *column* argument works only if the edit style of the column is CheckBox, DropDownListBox, Edit, or RadioButton.

- GetSQLSelect

  If the syntax in DataWindow is PBSELECT statement, when calling the GetSQLSelect function, it will still return the PBSELECT statement but not the SQL SELECT statement, which is different from that in PowerBuilder.

- ImportFile, ImportString, ImportClipboard

  1. The *filename* argument of ImportFile function must be a tab-separated file (TXT) or a comma-separated file (CSV).

  2. The *importtype* arguments (e.g. Text!, CSV! and XML!) of ImportString function are unsupported.

     The *importtype* arguments (e.g. Text!, CSV!, XML!, DBase2!, DBase3!) of ImportFile function are unsupported.

  3. When using the three functions, you should ensure that the data type imported matches the data type that you specified in a DataWindow control, DataStore object or graph control. Otherwise the output on the Web may be different from that in PowerBuilder. For example, if the date type specified in a DataWindow is string, when you import a

file with date number (eg, 2006-08-01), on the Web the date data will be recognized as a string (2006), however in PowerBuilder, it will be recognized as a date data (2006-08-01).

- Modify

  If the Modify function is used to modify the WHERE clause of the DataWindow object's SQL SELECT statement, please make sure that the new SQL SELECT statement is correct in syntax. Otherwise, syntax errors can occur on the Web because PowerBuilder does not validate the statement whereas JavaScript does.

- RowsMove

  Events cannot be triggered while the application moves data within the primary buffer of one DataWindow.

- SaveAs

  1. The *saveastype* argument can be TEXT, HTMLTable, WMF (EAServer only), XML, PDF*, EXCEL, or EXCEL5. For the EXCEL format, only up to 256 characters are supported and anything larger will be automatically truncated.

  2. The supported SaveAs syntax:

     ```
     ll_testvalue = ds_test.SaveAs(filename, saveastype, colheading, encoding)
     ```

     SaveAs PDF cannot support the colheading argument, for example, the following syntax is unsupported:

     ```
     Integer ds_test.SaveAs({string filename, saveastype saveastype, boolean
      colheading{, encoding encoding}})
     ```

     The unsupported SaveAs syntax:

     ```
     ds_test.SaveAs([filename, ]graphcontrol[, saveastype, colheading])
     ```

  3. On the Web, Nillable attribute is unsupported in an XML schema.

  4. If the DataStore without any data is saved as an HTML file or XML file, on Web the header will not be saved, whereas it will be saved on PB.

- SaveAsAscii

  Retainnewlinechar argument is unsupported.

- SetValue

  The *column* argument works only if the edit style of the column is DropdownListBox.

- SetSort

  Using "desc" or "asc" as the Order will not have any effect. Please use the standard Order value A for ascending or D for descending order.

- TriggerEvent

  The following TriggerEvent syntax is unsupported:

```
object1.TriggerEvent (object2, event)
```

11.12 DataWindowChild object

11.12.1 Properties of DataWindowChild object

Supported

None.

Unsupported

The unsupported property of DataWindowChild is ClassDefinition.

11.12.2 Functions for DataWindowChild

Supported

There are some limitations or differences for calling functions marked with the "*" symbol. Refer to the Limitations and Differences for Calling Functions.

Table 11.35:

AcceptText	ClassName	DeletedCount
DeleteRow	Describe	Filter
FilteredCount	Find	FindGroupChange
FindRequired	GetBandAtPointer	GetBorderStyle
GetChanges	GetChild	GetClickedColumn
GetClickedRow	GetColumn	GetColumnName
GetFormat	GetItemDate	GetItemDateTime
GetItemDecimal	GetItemNumber	GetItemStatus
GetItemString	GetItemTime	GetNextModified
GetObjectAtPointer	GetParent	GetRow
GetSelectedRow	GetSQLPreview	GetSQLSelect*
GetText	GetValidate	GetValue
GroupCalc	ImportClipboard*	ImportFile*
ImportString*	InsertRow	IsSelected
ModifiedCount	Modify*	Reset
ResetUpdate	Retrieve	RowCount
RowsCopy	RowsDiscard	RowsMove
SaveAs*	ScrollNextPage	ScrollNextRow
ScrollPriorPage	ScrollPriorRow	ScrollToRow
SetChanges	SetColumn	SetFilter
SetDetailHeight	SetFormat	SetItem
SetItemStatus	SetPosition*	SetRedraw

SetRow	SetRowFocusIndicator	SetSort*
SetSQLPreview	SetSQLSelect	SetTabOrder
SetText	SetTransObject	SetValue
SetValidate	SetWSObject*	ShareData
SharedDataOff	Sort*	TypeOf
Update		

Unsupported

Table 11.36:

CrosstabDialog	DBCancel	DBErrorCode
DBErrorMessage	GetContextService	GetRowIDFromRow
GetRowFromRowID	GetTrans	GetUpdateStatus
OLEActivate	ReselectRow	ResetTransObject
SelectRow	SetBorderStyle	SetTrans

Limitations and Differences for Calling Functions

- GetValue

  The column argument does not work only if the edit style of the column is CheckBox, DropDownListBox, Edit, or RadioButton.

- GetSQLSelect

  If the syntax in DataWindow is PBSELECT statement, when calling the GetSQLSelect function, it will still return the PBSELECT statement but not the SQL SELECT statement, which is different from that in PowerBuilder.

- ImportFile, ImportString, ImportClipboard

  1. The *filename* argument of ImportFile function must be a tab-separated file (TXT) or a comma-separated file (CSV).

  2. The *importtype* arguments (e.g. Text!, CSV! and XML!) of ImportString function are unsupported.

     The *importtype* arguments (e.g. Text!, CSV!, XML!, DBase2!, DBase3!) of ImportFile function are unsupported.

  3. When using the three functions, you should ensure that the data type imported matches the data type that you specified in a DataWindow control, DataStore object or graph control. Otherwise the output on the Web may be different from that in PowerBuilder. For example, if the date type specified in a DataWindow is string, when you import a file with date number (e.g., 2006-08-01), on the Web the date data will be recognized as a string (2006), however in PowerBuilder, it will be recognized as a date data (2006-08-01).

- Modify

  If the Modify function is used to modify the WHERE clause of the DataWindow object's SQL SELECT statement, please make sure that the new SQL SELECT statement is correct in syntax. Otherwise, syntax errors can occur because PowerBuilder does not validate the statement whereas JavaScript does.

- SaveAs

  1. The saveastype argument can be TEXT, HTMLTable, WMF (EAServer only), XML, PDF*, EXCEL, or EXCEL5. For the EXCEL format, only up to 256 characters are supported and anything larger will be automatically truncated.

  2. On the Web, Nillable attribute is unsupported in an XML schema.

  3. The supported SaveAs syntax:

     ```
     ll_testvalue = dwc_test.SaveAs(filename, saveastype, colheading, encoding)
     ```

     SaveAs PDF cannot support the colheading argument, for example, the following syntax is unsupported:

     ```
     Integer dwcontrol.SaveAs({string filename, saveastype saveastype, boolean
      colheading{, encoding encoding}})
     ```

     The unsupported SaveAs syntax:

     ```
     dwc_test.SaveAs([filename, ]graphcontrol[, saveastype, colheading])
     ```

  4. If the DataWindowChild without any data is saved as an HTML file or XML file, on Web the header will not be saved, whereas it will be saved on PB.

- SetSort

  Using "desc" or "asc" as the Order will not have any effect. Please use the standard Order value A for ascending or D for descending order.

- SetPosition

  The following syntax is supported:

  ```
  dwcontrol.SetPosition (position{, precedingobject})
  ```

  The *position* can be ToTop!, ToBottom!, but cannot be Behind!

  The following syntax is unsupported:

  ```
  dwcontrol.SetPosition(objectname, band, bringtofront)
  ```

- SetWSObject

  The SetWSObject function is used to set the authentication info for accessing Web service.

- Sort

  If the items in a Web DropDownDataWindow are double-byte (such as, Chinese, Korean, or Japanese), sorting the DropDownDataWindow has a different result from PowerBuilder. This is because the sorting method of PowerScript and JavaScript is different. For

example, if the charset is Chinese, PowerScript sorts by the spelling while JavaScript sorts by Unicode.

11.13 DataWindow performance considerations

Strong suggestion: reduce usage of DataWindow SQLPreview event

Each time the DataWindow SQLPreview event is triggered, the Web application will interact with Appeon Server twice, which costs 1-2 seconds. Therefore, Appeon recommends you minimize writing script into the SQLPreview event of the DataWindow.

Suggestion: use Describe and Modify to get and set DataWindow object properties

Dot notation is much less efficient than Describe and Modify functions. In general, the Describe and Modify functions are about two to three times faster than dot notation. Therefore, Appeon recommends you use Describe function to replace dot notation that gets the DataWindow object properties, and use the Modify function to replace dot notation that sets the DataWindow object properties.

Suggestion: minimize modifying DataWindow DataObject

It takes 0.3 seconds each time the DataObject property of DataWindow/DataStore is modified during application run time. Therefore, modifying the DataObject property frequently could slow down performance significantly. Appeon recommends you minimize modifying the DataObject property.

12 DBParm parameters in Database

Supported

The following DBParm parameters are supported: CommitOnDisconnect, CacheName, DelimitIdentifier, DateTimeAllowed, DisableBind, TrimSpaces and Identity.

1. The data source name for the CacheName parameter is case sensitive in JBoss, JEUS, WebLogic and WebSphere (but not in EAServer or IIS .NET).

2. DisableBind parameter is only supported in ESQL.

3. In PowerBuilder, the data source name only applies when a PowerBuilder NVO is deployed to EAServer.In Appeon, the data source specified applies for the connection of the deployed application to the database.Settings to the other parameters in the script will be ignored.

4. To support DelimitIdentifier parameter, you must set the "Enclose table and column names in double quotes" option in the Appeon Developer | DB Type Profile Configuration page before deployment.

5. If table and column names are keywords that must be enclosed in double quotation marks (with DelimitIdentifier set to "yes"), the SQL statement cannot be a PowerBuilder SELECT statement, otherwise, PDF Print and Save As PDF will fail.

6. The DelimitIdentifier parameter is unsupported when the updated table name of DataWindow contains the owner name. In this case, be sure to clear the "Enclose table and column names in double quotes" check box in Appeon Developer.

7. It is unsupported to dynamically set the DelimitIdentifier parameter. For all supported DBParm parameters (excluding DelimitIdentifier) you can only dynamically set them before executing Connect.

8. Supports the Identity property of Transaction DBParm, for example: Identity='@@IDENTITY'.

Unsupported

Except the parameters above, other DBParm parameters are unsupported.

13 Calling Web Service

Appeon supports to call Web Service thru the AppeonWebService object.

AppeonWebService is an Appeon customized object, provided in Chapter 2, *Appeon Workarounds PBL Reference* in *Workarounds & API Guide*. You can successfully call a Web Service via AppeonWebService object on the Web, without needing to install any add-on on end users' machine. The Web Service call will fail in client/server PowerBuilder application.

14 Undetected Unsupported Features

Appeon provides an Unsupported Features Analysis tool for detecting unsupported features in a PowerBuilder application before generating files for the application. The Unsupported Features Analysis scans each object within a PowerBuilder application for unsupported features and generates a report that, when combined with the this help, provides a foundation for bringing a PowerBuilder application to specification.

The Appeon Unsupported Features Analysis is capable of highlighting the majority of unsupported features contained in the PowerBuilder application; however, there are some features that the Unsupported Features Analysis will not detect, and thus they will not be listed in the Analysis Report. Approximately 5% of PowerBuilder features that are not supported by Appeon may not be detected in the Unsupported Features Analysis.

An application containing unsupported features can still be converted to the Web and work correctly, depending on the type and number of unsupported features. If the Web application does not work correctly but the Unsupported Features Analysis does not indicate any unsupported features that are likely to cause such a problem, the problem may be caused by an unsupported feature that has not been detected. In this case, it is strongly recommended that you carefully read the undetected features listed below and examine your application to determine if it contains any of these features.

The following are unsupported features that are not detected during Unsupported Features Analysis.

Table 14.1: Undetected features

Naming conventions	• Duplicate object names. No two objects should have the same name in an application, whether they are of the same type or not. The Unsupported Features Analysis cannot detect whether object names are duplicated in an application. • More than one application object. Having more than one application object in an application is unsupported. The Unsupported Features Analysis cannot detect whether more than one application object is present. • Applications named as objects or controls. Application names that have the same name as PowerBuilder control/object types are not supported. The Unsupported Features Analysis cannot detect whether this unsupported issue is present in an application.
Null values	• Unsupported operation • Expressions with Null values
ASCII characters	Vertical tab (~v)
Non-visual UserObject	A NonVisualObject object assigned to an autoinstantiated NVO or an autoinstantiated NVO assigned to a NonVisualObject object.
Variables and constants	• Instance variables have identical names as global variables. • longlong variable

	• PUBLIC, PROTECTED, PRIVATE qualifier in the variable declaration
Forced conversion	Forced conversion between types
Window	Multiple MDI windows in an application.
Overloading, overriding functions	• Dynamic calling for overloaded functions • Dynamic calling for overriding functions
Using the return value of some supported functions	The return value of the Open or OpenSheet functions
User functions	Remote procedure call (RPC)
Operators	The operator '^' with embedded SQL statements.
Stored procedure	• Stored procedures declared in the conditional statement • DB2 stored procedures • Stored procedures placed inside an Oracle package • Oracle stored procedures with Appeon unsupported features
Cursor declare requirement	• Cursor declared in the conditional statement; • If a cursor is declared for retrieving rows from table X, the table X (insert, delete, update) is modified during the cursor declare-close period.
Cursor statements	• UPDATE TableName SetStatement WHERE CURRENT OF CursorName; • DELETE FROM TableName WHERE CURRENT OF CursorName.
DataWindow data expressions	• Unsupported features in the DataWindow dot notations. You must make sure the specifications about dot notations are satisfied in the application. Otherwise, it will cause errors. • The return values of the DataWindow dot notations are assigned to user objects.
DataWindow expression function	lastpos | lastposw | mode | pageAcross | pageCount Across | profileint profile string | stdev | var | varp
Partially supported features	Some partially supported features (such as DataWindow Modify and Describe functions).

<table>
<tr><td>User interface interactions</td><td colspan="6">Unsupported features in the UI such as shortcut key.</td></tr>
<tr><td>Enumerated data type</td><td colspan="6">Default values of enumerate type variable</td></tr>
<tr><td>Data source</td><td colspan="6">The data sources of dynamically created DataWindows are stored procedures with input parameters.</td></tr>
<tr><td>Dynamic calls</td><td colspan="6">• Dynamically call the method of a menu object.
• Dynamically call the method that contains reference arguments.</td></tr>
<tr><td>Others</td><td colspan="6">• PBX
• PSR, for example: dw.dataobject='*.psr'
• Unsupported DBParm parameters
• Encoding parameter of the Blob functions
• filename & importtype arguments of ImportFile, ImportString, & ImportClipboard (filename can only be a text file (TXT))
• Property defined in the string variable
• Structure member has comment property</td></tr>
<tr><td>System events</td><td colspan="6">The following are unsupported and undetected system events.</td></tr>
<tr><td></td><td>cbgetcount</td><td>dderequest</td><td>gettext length</td><td>mmmom close</td><td>sbnpageup</td><td>vbxevent12</td></tr>
<tr><td></td><td>cbgetcursel</td><td>ddeter minate</td><td>globalrc change</td><td>mmmom done</td><td>sbnsetfocus</td><td>vbxevent13</td></tr>
<tr><td></td><td>cbget dropped controlrect</td><td>ddeun advise</td><td>heditctl</td><td>mmmom open</td><td>sbnthumb position</td><td>vbxevent14</td></tr>
<tr><td></td><td>cbget dropped state</td><td>deactivate</td><td>help</td><td>mmwim close</td><td>sbnthumb track</td><td>vbxevent15</td></tr>
<tr><td></td><td>cbgeteditsel</td><td>deadchar</td><td>hookrc result</td><td>mmwim data</td><td>sbntop</td><td>vbxevent16</td></tr>
<tr><td></td><td>cbgetext endedui</td><td>deleteitem</td><td>hscroll</td><td>mmwim open</td><td>setcursor</td><td>vbxevent17</td></tr>
<tr><td></td><td>cbgetitem data</td><td>destroyclip board</td><td>hscrollclip board</td><td>mmwom close</td><td>setfont</td><td>vbxevent18</td></tr>
<tr><td></td><td>cbgetitem height</td><td>devmode change</td><td>iconerase bkgnd</td><td>mmwom done</td><td>settext</td><td>vbxevent19</td></tr>
<tr><td></td><td>cbgetlbtext len</td><td>drawclip board</td><td>initdialog</td><td>mmwom open</td><td>sizeclip board</td><td>vbxevent20</td></tr>
</table>

cbinsert string	drawitem	initmenu	mouse activate	skb	vbxevent21
cblimittext	dropfiles	initmenu popup	move	spooler status	vbxevent22
cbncloseup	dwclose dropdown	keyup	ncactivate	syschar	vbxevent23
cbndblclk	dwescape	lbaddstring	nccalcsize	syscolor change	vbxevent24
cbndelayed recogfail	dwnbackt about	lbdelete string	nccreate	sys command	vbxevent25
cbndrag drop	dwndrop down	lbdir	ncdestroy	sysdead char	vbxevent26
cbndrag enter	dwngraph create	lbfindstring	nchittest	syskeydown	vbxevent27
cbndrag leave	emgetword breakproc	lbfindstring exact	nclbutton dblclk	systemerror	vbxevent28
cbndrag within	emlimittext	lbgetcaret index	nclbutton down	sysykeyup	vbxevent29
cbndrop down	emlinefrom char	lbgetcount	ncmbutton dblclk	tbnmoved	vbxevent30
cbnedit change	emlineindex	lbgetcursel	ncmbutton down	tcnclicked	vbxevent31
cbneditup date	emline length	lbgeth orizontal extent	ncmbutton up	tcndouble clicked	vbxevent32
cbnendrec	emline scroll	lbgetitem data	ncmouse move	tcndragdrop	vbxevent33
cbnerrspace	emreplace sel	lbgetitem height	ncpaint	tcndragenter	vbxevent34
cbnkillfocus	emscroll	lbgetitem rect	ncrbutton dblclk	tcndragleave	vbxevent35
cbn modified	emsetfont	mbutton down	ncrbutton down	tcndrag within	vbxevent36
cbnrcresult	emset handle	mbuttonup	ncrbuttonup	tcnenterkey	vbxevent37
cbnsel change	emset modify	mdiactivate	nextdlgctl	tcnkeydown	vbxevent38
cbnselend cancel	emsetpass wordchar	mdicascade	omnclose	tcnkillfocus	vbxevent39
cbnselend ok	emsetread only	mdicon arrange	omndata change	tcnrclicked	vbxevent40

cbnsetfocus	emsetrect	mdicreate	omndrag drop	tcnrdouble clicked	vbxevent41
cbreset content	emsetrectnp	mdidestroy	omndrag enter	tcnsel changed	vbxevent42
cbselect string	emsetsel	mdiget active	omndrag within	tcnsel changing	vbxevent43
cbsetcursel	emsettab stops	mdimaxi mize	querynew palette	tcnsetfocus	vbxevent44
cbseteditsel	emsetword break	mdinext	queryopen	timechange	vbxevent45
cbsetextend edui	emsetword breakproc	mdirestore	queuesync	tvnenterkey	vbxevent46
cbsetitem data	emundo	mdisetmenu	quit	uondrag drop	vbxevent47
cbsetitem height	enable	mditile	rcresult	uondrag enter	vbxevent48
cbshow dropdown	enchange	measure item	renderall formats	uondrag leave	vbxevent49
changecb chain	endsession	menuchar	render format	uondrag within	vbxevent50
char	enerrspace	menuselect	renhscroll	uonexternal 01	vkeytoitem
chartoitem	enhscroll	mmjoy1 buttondown	renvscroll	vbxevent01	vscroll
child activate	enmaxtext	mmjoy1 buttonup	sbnbottom	vbxevent02	vscroll clipboard
commnotify	enteridle	mmjoy1 move	sbndrag drop	vbxevent03	windowpos changed
compacting	enupdate	mmjoy1z move	sbndrag enter	vbxevent04	windowpos changing
compare item	envscroll	mmmci notify	sbndrag leave	vbxevent05	winini change
ctlcolor	erasebkgnd	mmmim close	sbndrag within	vbxevent06	Selected
ddeack	fontchange	mmmim data	sbnend scroll	vbxevent07	OMNDrag Leave
ddeadvise	getdlgcode	mmmim error	sbnkillfocus	vbxevent08	nclbuttonup
ddedata	getfont	mmmim longdata	sbnline down	vbxevent09	

	ddeexecute	getmin maxinfo	mmmim longerror	sbnlineup	vbxevent10	
	ddepoke	gettext	mmmim open	sbnpage down	vbxevent11	