Appeon Supported Features Guide

for Pure-JavaScript Deployment

Appeon® 3.1 for PowerBuilder®

For Windows

DOCUMENT ID: DC37818-01-0310-01

LAST REVISED: September 13, 2005

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

What's New since Appeon 2.8 for PowerBuilder

The following features are newly added to Pure-JavaScript deployment since Appeon 2.8 for PowerBuilder.

Unicode support

Appeon DataWindow Menu client functions

Appeon client functions

Unicode support

Appeon 3.1 is Unicode enabled - it supports the Unicode character encoding in PowerBuilder 10. This means that, beginning with Appeon 3.1, you can display characters from multiple languages on the same page of a Web application. Appeon 3.1 provides support to other Unicode related functions for backward compatibility with earlier versions of PowerBuilder.

Supported

Optional *encoding* parameter

Appeon supports the optional *encoding* parameter in the SaveAs function.

String-related functions

Appeon 3.1 supports the string-related functions in both PowerBuilder 10 as well as PowerBuilder 8 and 9. The following changes, made in PowerBuilder 10, are also supported. For more details, refer to the String functions section.

Unsupported

Calling external functions

Appeon does not support Unicode encoding when a Web application makes calls to external functions.

Overview

Sybase® PowerBuilder® applications can be constructed in many different ways, but there are general guidelines to follow and common pitfalls to avoid. An application that does not adhere to the basic requirements and architectural requirements will not convert successfully and/or will suffer from performance and scalability problems.

Before attempting to use Appeon® for PowerBuilder®, verify that your application meets the requirements outlined in this section. If your application does not meet these requirements, you must modify your application so it is compliant before beginning to use Appeon for PowerBuilder.

Requirements checklist

The following checklist outlines most of the basic and architectural requirements that your application must meet. This checklist is a quick way to assess whether there are any major high-level issues with your application. However, it is highly recommended that you carefully read all the requirements in the basic requiremetns and architectural requirements section, since you will not be able to effectively use Appeon for PowerBuilder unless your application meets all of the following requirements:

1. Are the PBLs less than 100MB?
2. Is your application coded in a relatively simple manner as follows: few deep inheritance and nesting levels, few cross-referenced objects and simple code structure?
3. Is the application code upgraded to be 100% compatible with Appeon-supported versions of PowerBuilder? Refer to Appeon Installation Guide for more details.
4. Is the application user interface (UI) and the data in English, Chinese, Korean, or Japanese?
5. Have all the objects and controls of the application been defined in the PowerBuilder painter and statically created? The exceptions are DataStore, DynamicStagingArea, Menu, Transaction Object and non-visual user objects. They can either be created in PB painter or be dynamically created.
6. Is the application free of OLE controls and external visual user objects?
7. Can the application generate .exe files (with no errors) in PowerBuilder 8.0.4 or 9.0.1?
8. Has the application been tested thoroughly to be bug-free?

If you answered yes to all of these questions above, please proceed to verify that your application meets the additional requirements outlined in the basic requirements and architectural requirements section.

In general, this checklist can aid in determining whether an application can be migrated using Appeon. However, some extremely complex frameworks will have difficulty in migrating cleanly. For example, PowerCerv PowerTool and other complex frameworks may require additional coding to migrate successfully to the Web, despite complying with the requirements listed above.

Critical architectural requirements

Application size requirement

Application complexity requirement

External independency requirement

Application size requirement

APPEON

The size of the application source code is one of the two primary factors that determines whether an application can be deployed with Appeon.

If the application is smaller than 25MB (including framework PBLs), in most cases it can be deployed with Appeon.
Successful deployment of applications between 25MB and 100MB will be a function of both size and complexity.
Generally, applications that are over 100MB in size (including framework PBLs) can be problematic.

Application complexity requirement

APPEON

Complexity of the application is another factor that determines whether the application can be deployed successfully with Appeon. Generally, the more complex an application is, the more difficult it is to deploy it to the Web. Significant time will need to be spent to work around unsupported features and fine tune performance. One method for measuring the complexity of an application is to determine the percentage of complex objects contained in the application. An object can be simple or complex depending on the number of cross-references, the number of nesting levels and the number of inheritance it has.

See below for examples of each:

- Object reference type 1: When an object is referred to in the script of an application. For example, in the script of object A, there are references to object B, object C, and object D. In this case, the number of object references in object A is three.

  A Window, Menu or UserObject is complex if the number of object references of type 1 is more than ten.

- Object reference type 2: A window or visual user object contains complex visual object(s) such as DataWindow control, TreeView control, ListView control, Tab control, or Custom visual object.

  A window or visual user object containing complex visual object(s) such as DataWindow or TreeView controls is complex if the number of object references exceeds four.

- Nesting type 1: When object A refers to object B, and object B refers to object C. In this case, the nesting level of object A is three.

  An object is considered as a complex object if the nesting level of the object is more than five.

- Nesting type 2: When object A contains object B, object B contains object C, and the script accesses to a property or function of object C via object A. The nesting level of the script is three.

  For example, a Window w_main has a Menu object m_mymenu, and m_mymenu has a File menu with an Open item. You can access the Visible property of the Open item with script w_main.m_mymenu.m_file.m_open.Visible. In this case, the nesting level of the script is four.

  An object is considered as complex if more than 1/3 of event/function calls in the object have 4+ nesting levels of type 2.

- Although inheritance is supported, increasing levels of inheritance usually makes an application more complex. An object is considered complex if it has more than four levels of inheritance.

Generally, the application complexity is considered low if the complex objects exhibiting one or more of the above features comprise only 20 percent of the entire application. If complex objects comprise of 20 to 50 percent of the application, the application complexity is considered moderate. Application complexity is considered high if complex objects in the application exceed 50 percent. Applications with high complexity may be problematic for Web migration.

External independency requirement

APPEON

The application can act upon some hardware (including the mouse and keyboard) via DLLs. For supported features about DLLs, refer to the section Program access techniques.

Although registry functions are supported, the functions are only supported to work with existing registry keys, and it is not supported to create new keys during application runtime.

The application can directly call the N-Tier NVOs deployed to EAServer/Appeon Server. The server NVOs can act as a bridge to call DLLs, other NVO components and/or EJB/Java components using IIOP. Web services can be used to integrate with other Web services-enabled applications, including J2EE and .net applications.

Coding style requirements

APPEON

Object definition requirement

Object naming requirement

Object definition requirement

APPEON

There can only be one Application object in an application.

All objects and controls in the application must be statically created (e.g. objects and controls defined in the PowerBuilder painters). The only exceptions are DataStore, DynamicStagingArea, Menu and Transaction Object, and non-visual user objects. For more details, see User Objects.

- No controls and objects in the DataWindow can be dynamically created. They can only be created in the painter.
- Distributed DataWindow/DataStores are supported by using the following workaround: Use the descendants of appeondatawindow and appeondatastore provided by Appeon, and always use the overriding GetFullState, SetFullState, GetChanges and SetChanges functions with the string

parameter.

- For Tabular, Freeform and Grid DataWindows:
  - DataWindows must be defined in the PowerBuilder painter and cannot be dynamically created (for example, using the CREATE statement).
  - Controls cannot be dynamically created, added or removed from the DataWindow.

- For Label, N-Up, Group, Composite, Graph, CrossTab and Nested DataWindows:
  - DataWindows must be defined in the PowerBuilder painter and cannot be dynamically created (for example, using the CREATE statement).
  - Controls cannot be dynamically created, added or removed from the DataWindow.
  - Only a few DataWindow functions including Print, SaveAs, and Retrieve are supported; most of the DataWindow functions, such as InsertRow, Update, GetItemValue, are unsupported.
  - If the presentation style applied to a DataStore object is not Grid, Freeform or Tabular, only the DataStore Print and SaveAs functions are supported.
  - Any limitations required by PowerBuilder SaveAs method for saving DataWindow contents to WMF must be followed.
  - If a DataWindow contains group(s), it is recommended to disable the option "New Page on Group Break". This will ensure the DataWindow is displayed correctly.

- For Grid DataWindows:
  - If any control in DataWindow is placed over a grid line, the control will not display in a deployed DataWindow.

Object naming requirement

APPEON

In PowerBuilder, two objects of the same type cannot have the same name, unless the objects are of different types or reside in different PBLs.

However, when converting an application to the Web, there are certain restrictions to name objects as follows:

- Objects within a PBL that have identical names, regardless of the object type, are unsupported.
- Objects of the same type residing in different PBLs and having identical names are unsupported.

Transaction requirements

APPEON

Cursor declare requirement

Cursor declare requirement

APPEON

Requirement: If a cursor is declared for retrieving rows from database table X, do not modify (insert, delete, update) database table X during the period from the cursor opening to the closing.

Incorrect usage example:

The following is an example of incorrect transaction:

Declare cursor --> Open cursor --> Update the database table where the cursor will retrieve data --> Retrieve data

At the fourth step, the cursor retrieves data from the newly updated database table in PowerBuilder; while in the Web application, deployed by Appeon for PowerBuilder, the cursor retrieves data from the old database table. Therefore, the Appeon result set will be incorrect.

The syntax in the following example is unsupported:

```
DECLARE cur_empl CURSOR FOR select s_emplid, s_emplname from
employee;
OPEN cur_empl;
INSERT INTO employee (s_emplid, s_emplname) VALUES (:ls_emplid,
:ls_emplname);
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
```

Correct usage example:

Appeon recommends that the code be rewritten to comply with the cursor requirement, as follows:

```
INSERT INTO employee (s_emplid, s_emplname) VALUES (:ls_emplid,
:ls_emplname);
COMMIT;
DECLARE cur_empl CURSOR FOR select s_emplid, s_emplname from
employee;
OPEN cur_empl;
FETCH cur_empl INTO :ls_emplid, :ls_emplname;
DO WHILE sqlca.sqlcode=0
      FETCH cur_empl INTO :ls_emplid, :ls_emplname;
LOOP
```

Database requirements

APPEON

DBMS requirement

Database protocol requirement

Lock database preference

DBMS requirement

APPEON

The following database systems have been tested and certified for compatibility:

- Sybase Adaptive Server® Anywhere 7.0.4, 8.0.2 and 9.x
- Sybase Adaptive Server® Enterprise 12.x
- Microsoft SQL Server 2000
- Oracle 8i or 9i
- IBM DB2 UDB 8.1

It is possible to use another JDBC-compliant DBMS, but a non-certified server is difficult to troubleshoot in the event of a technical support issue.

Database protocol requirement

APPEON

The PowerBuilder application may use any database protocol; however, when the application is deployed to the Web, it may only interface with the database using the JDBC protocol.

Even though your PowerBuilder application may use a non-JDBC protocol such as ODBC, there is no need to make any code changes to your application. All that is necessary is to set up your database and application server for JDBC connection caches. For guidance and assistance, refer to the *Appeon Migration User Guide*.

Lock database preference

APPEON

It is unsupported to control the isolation level for connections to a database in a script by setting the Lock value of a transaction object (the Lock property is unsupported) or by using the syntax EXECUTE IMMEDIATE "set transaction isolation level n".

If it is necessary to specify the isolation level, the solution is to use an ODBC data source for the application database. In the ODBC Administrator, specify the isolation level of the ODBC data source at 2 or 3 or 4 as necessary. While you are doing the pre-configuration for the Web application, please set up connection caches that use the JDBC-ODBC driver. For more information on setting up connection caches, refer to Section 3.7: *Pre-configure for the Web application* in the *Appeon Migration Guide*.

Application PBL requirement

APPEON

The application code must be 100% PowerBuilder 8.0 or 9.0 code. If the application was built with a version of PowerBuilder earlier than 8.0, the application code must be completely upgraded to PowerBuilder 8.0 or 9.0.

Non-PowerBuilder 8.0 or 9.0 code, including code that is defined as obsolete in PowerBuilder 8.0 or 9.0 help, must be replaced/upgraded before you can begin to use Appeon for PowerBuilder. In addition, supported PowerBuilder 8.0 or 9.0 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 8.0 or 9.0

- In PowerBuilder 8.0 or 9.0, 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 8.0 or 9.0, the edit styles of the DataWindows are not null.
- In the earlier versions of PowerBuilder, the user can use GetChild function to provide a reference to a child DataWindow in that DataWindow itself. However, in PowerBuilder 8.0, this is not supported.
- If an application is updated to PowerBuilder 8.0 or 9.0, some DataWindow columns may have duplicate names. Duplicate names for columns in a DataWindow are unsupported.

Upgrading Menu objects to PowerBuilder 8.0 or 9.0

- With menu inheritance, the source code in the earlier PowerBuilder versions is different from the source code in PowerBuilder 8.0 or 9.0. The user must open the menu objects with inheritance in PowerBuilder 8.0 or 9.0and upgrade the objects to PowerBuilder 8.0 or 9.0 objects.

Upgrading color specification to PowerBuilder 8.0 or 9.0

- A color value in an earlier PowerBuilder version may map to a different color of the same value in PowerBuilder 8.0 or 9.0. If you prefer the color of the earlier version, when upgrading the application to a PowerBuilder 8.0 or 9.0application, be sure to verify that the correct colors are selected.

Application language requirement

Text in the user interface (UI) of the application and application data (from the database) must be in one of the following five languages:

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

Other languages can be potentially supported, but they have not been tested.

Notes:
1) If the items in a Web DropDownDataWindow are double-byte (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.

2) For multi-byte character languages, such as Chinese, Korean, or Japanese, multi-byte characters are considered as single-byte characters in the Web application. For example, if you input three Chinese characters in a column and select the characters, when you execute the SelectedLength function, SelectedLength returns 3 in the Web application but 6 in PowerBuilder.

For PowerBuilder applications coded in languages other than English, the developer needs to switch the language setting in Appeon Developer. Detailed instructions are provided in the *Appeon Developer User Guide*.

Overview

APPEON

PowerBuilder applications can be constructed in many different ways. Appeon recommends ways to optimize the performance for deploying and running the converted Web applications. An application that does not adhere to these important recommendations outlined in the recommendations section may encounter performance problems.

Before attempting to use Appeon for PowerBuilder, verify that your application meets the recommendations discussed in the following sections. If your application does not meet these criteria, it is strongly recommended that you modify your application so it is compliant before beginning to use Appeon for PowerBuilder.

Recommendations checklist

APPEON

The following checklist outlines most of the recommendations for optimizing the performance of your application.

1. Does each window in a given application contain a combination of ten or less DataWindows, DataStores, and TreeViews? Two factors determine the heaviness of a window: the number and weight of the controls in the window. Heavy windows are an important issue that greatly affects the application conversion process and performance of the Web application at run-time. Please refer to the Window size recommendations section for details.

2. Does the application use a well-separated user object structure? If you want to see an example of a well-separated user object structure, please refer to the User object structure recommendation section.

If you answered yes to all the questions above, please proceed to verify that your application meets the additional recommendations outlined in this section. If not, review your application and make the necessary changes so that you can answer yes to all the questions above.

Architectural recommendations

APPEON

Window size recommendation

User object structure recommendation

Window size recommendation

It is recommended that an application window contains no more than ten DataWindows, DataStores or TreeViews in order to avoid becoming a Heavy window. Heavy windows slow down the Web conversion process significantly and result in the poor performance of the converted Web application at run-time. In addition, in some cases, a Window that is too heavy can cause Appeon to crash or hang during the unsupported feature analysis or deployment processes. Windows with more than 10 DataWindows, DataStores or TreeViews are considered too large.

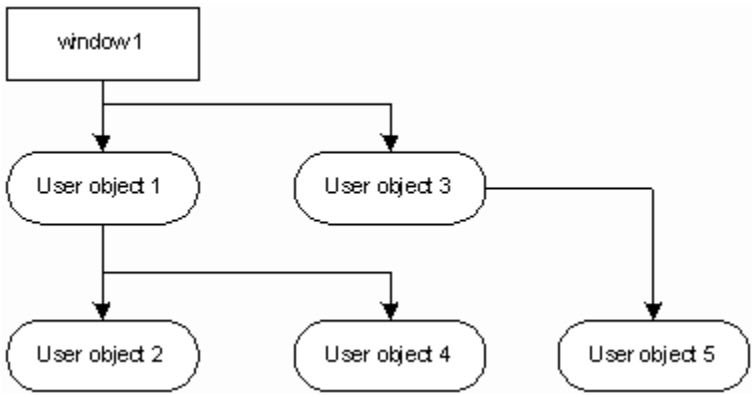
Heavy windows can be avoided by ensuring that the number of DataWindows, DataStores, and advanced data-bound controls (e.g. TreeView) in a single window does not exceed ten. Also avoid using excessive lines of PowerScript and/or user objects in the window, since each line of PowerScript also adds some weight to the application.

User object structure recommendation

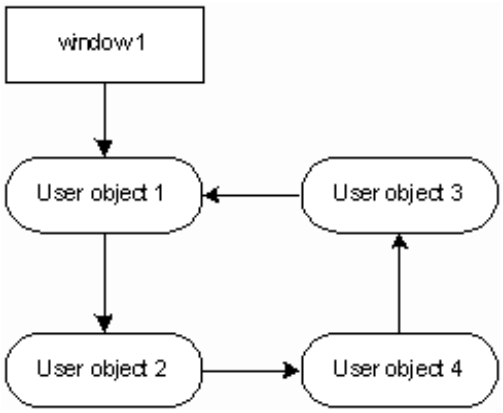
The user objects in your application should be well separated. This type of structure has several benefits:

- Reduces the need for generic coding.
- Reduces the chances of Heavy Windows.
- Makes it easy to deploy a portion of the application to the Web.
- Makes it possible to remove unsupported features without breaking-up the entire application or requiring lots of time to rework the application.

The following is an example of a well separated user object structure:



Although Appeon can convert applications with poorly structured objects that do not conform the structure depicted in the previous figure, a complex user object structure (where the user object reference paths oscillate back and forth among one another) should be avoided or reworked into the recommended structure. A complex structure typically exhibits cyclic calling: user object 1 calls to user object 2, user object 2 calls to user object 4, and user object 3 calls to user object 1 etc:



With complex or cyclical structures, the calling level can become too deep. This can cause the a stack overflow error in the converted Web application. In addition, complex structures can increase the load time or reduce the speed of executing an operation in the converted Web application.

Coding style recommendations

APPEON

Generic coding recommendations

Generic coding recommendations

APPEON

It is highly recommended that you code your application using non-generic code. Since the type of the object is not specific (for example, can be a letter, a digit, or a code), the Unsupported Features Analysis tool of Appeon Developer cannot verify whether the properties, functions and events of the object are supported. Using unsupported properties, functions, or events would cause errors when running the application on the Web .

If you code your application using generic code, you may only use the following generic-coding styles:

- Supported PowerBuilder classes - PowerObject, DragObject, GraphicObject and WindowObject.
- PowerBuilder Any data type - The Any data type is fully supported.

Transaction recommendations

APPEON

SQL statement recommendation

Commit transaction recommendation

SQL statement recommendation

APPEON

For a Web application with many concurrent users, many transactions are taking place at the same time. It is important to follow the recommendations below to effectively manage many transactions and avoid delaying server responses.

Recommendation 1: Avoid placing non-SQL statements in a transaction.
Avoid placing non-SQL statements in a transaction. If it is necessary to use non-SQL statements in a transaction, you must do a "commit" before the non-SQL statement that needs user interaction.

Incorrect usage example:

In the following example, during a single transaction, the user clicks the command button 1 and deletes a data record. Then the user clicks the command button 2, inserts a data record, and finally closes the window. The transaction locks the database server resource for a long time.

```
Window open();
//Profile appeonsample
SQLCA.DBMS = "ODBC"
SQLCA.AutoCommit = "False"
SQLCA.DBParm = "ConnectString = 'DSN=AppeonSample; UID=dba;
PWD=sql"
CONNECT;

Command Button cb_1:
String ls_emplid
ls_emplid = sle_1.text
DELETE employee WHERE s_emplid = :ls_emplid;

Command Button cb_2:
String ls_emplid, ls_emplname
ls_emplid = sle_1.text
ls_emplname = sle_2.text
INSERT INTO employee(s_emplid, s_emplname) VALUES (:ls_emplid,
:ls_emplname)

Window close ( )
COMMIT;
```

Correct usage example:

The following is an example of how the incorrect code in the previous section should be rewritten to comply with Recommendation 1, thereby reducing delay in the server response.

```
Window open();
//Profile appeonsample
SQLCA.DBMS = "ODBC"
SQLCA.AutoCommit = "False"
SQLCA.DBParm = "ConnectString = 'DSN=AppeonSample; UID=dba;
PWD=sql"
CONNECT;

Command Button cb_1:
String ls_emplid
```

```
ls_emplid = sle_1.text
DELETE employee WHERE s_emplid = :ls_emplid;
COMMIT;

Command Button cb_2:
String ls_emplid, ls_emplname
ls_emplid = sle_1.text
ls_emplname = sle_2.text
INSERT INTO employee(s_emplid, s_emplname) VALUES (:ls_emplid,
:ls_emplname)
COMMIT;
Window close ( )
```

Commit transaction recommendation

APPEON

Recommendation 2: Commit a transaction before the SELECT statements, the Cursor statements or the Procedure statements.

Incorrect usage example:

In the following example, the cursor statements fetch rows into the dropdown listbox. The statements involve time-consuming interactions with the database server. Therefore, the statements before the cursor statement must be committed at an earlier time.

```
Window open();
//Profile appeonsample
SQLCA.DBMS = "ODBC"
SQLCA.AutoCommit = "False"
SQLCA.DBParm = "ConnectString = 'DSN=AppeonSample; UID=dba;
PWD=sql"
CONNECT;

Command Button cb_1:
String ls_emplid, ls_emplname
Ls_emplid = sle_1.text
DELETE employee WHERE s_emplid = :ls_emplid;
DECLARE cur_empl CURSOR FOR SELECT s_emplid, s_emplname from
employee; ;
OPEN cur_empl;
FETCH cur_empl INTO :ls_emplid, :ls_emplname;
DO WHILE sqlca.sqlcode=0
        Ddlb_1.additem("["+ls_emplic+"]"+ls_emplname)
        FETCH cur_empl INTO :ls_emplid, :ls_emplname;

LOOP
CLOSE cur_empl;
Ddlb_1.selectitem(1)
Ddlb_1.triggerevent ("eventchanged")
COMMIT;
```

Correct usage example:

The following is an example of how the incorrect code in the previous section should be rewritten to comply with Recommendation 2. Interaction time with the database server is reduced.

```
Window open();
//Profile appeonsample
SQLCA.DBMS = "ODBC"
SQLCA.AutoCommit = "False"
SQLCA.DBParm = "ConnectString = 'DSN=AppeonSample; UID=dba;
PWD=sql"
CONNECT;

Command Button cb_1:
String ls_emplid, ls_emplname
Ls_emplid = sle_1.text
DELETE employee WHERE s_emplid = :ls_emplid;
COMMIT;

DECLARE cur_empl CURSOR FOR SELECT s_emplid, s_emplname from
employee; ;
OPEN cur_empl;
FETCH cur_empl INTO :ls_emplid, :ls_emplname;
DO WHILE sqlca.sqlcode=0
        Ddlb_1.additem("["+ls_emplic+"]"+ls_emplname)
        FETCH cur_empl INTO :ls_emplid, :ls_emplname;

LOOP
CLOSE cur_empl;
Ddlb_1.selectitem(1)
Ddlb_1.triggerevent ("eventchanged")
COMMIT;
```

Server interaction recommendation

Each server interaction has a negative impact on performance. To optimize the responsiveness of the server when a given user action is performed (such as clicking on the *Save* button to save the changes made to DataWindows, opening a Window that automatically loads data, etc) it is recommended that the number of server interactions be kept to a minimum.

The following statements, when executed, will result in server interactions:

- DataWindow.Update
- DataWindow.Retrieve
- DataStore.Update
- DataStore.Retrieve
- SQL Statements

If you place business logics into one NVO with DataStores, and then deploy the NVO to the Appeon Server, the number of interactions previously required to execute the business logics can be reduced to one.

Distributed Application Support

N-Tier PowerBuilder NVOs

N-Tier DataWindows

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, EJB, C/C++ DLLs, and ActiveX components, and so on.

The basic limitation 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 the Web deployment of the PowerBuilder client, and the component type must be a PowerBuilder NVO.

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 the Connection Object section in this book for information on supported and unsupported features.

N-Tier PowerBuilder NVOs

Supported

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

Boolean	Char	Date	DateTime
Decimal	Double	Int	Integer
Long	Real	String	Time
UInt	ULong	UnsignedInt	UnsignedInteger

UnsignedLong

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

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
- Pass arrays as parameters. However, arrays can be used in NVOs

You can use some environment-related functions. However, the return value in the Web application will be different from the return value in the PowerBuilder application. For example, the GetEnvironment function returns the system information of the client machine when executed in the PowerBuilder client, but it returns the server machine information when executed in EAServer.

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

- The proxy object for 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 parameter and return value of an NVO public interface cannot be complex data types, arrays, or structures.
- The parameter of an NVO public interface cannot be passed by reference.
- The script for an NVO or the script referred in a NVO cannot contain UI-related script or object.

N-Tier DataWindows

Supported

Appeon can support the N-Tier DataWindows and the functions GetChanges, GetFullState, SetChanges, and SetFullState, as long as the Appeon workaround for N-Tier DataWindows in the PowerBuilder application is implemented.

For more instructions on how to use the workarounds, please refer to the PowerBuilder Workarounds section in *Appeon Migration User Guide*.

Unsupported

When using the *appeondatawindow* and *appeondatastore* objects to work around the distributed DataWindow technique, there are some limitations regarding the use of Appeon GetFullState, SetFullState, GetChanges and SetChanges functions. Please be aware of these limitations:

- DataWindow style – These four Appeon functions only work with DataWindow of Grid, Freeform, and Tabular styles.
- You can only use the Appeon workaround distributed DataWindow technique to manipulate DataWindow data and data status. Manipulation of DataWindow properties, such as modifying the sort or filter criteria, is not supported.
- Dynamically create Datawindow will not be supported at runtime.
- If using the DW ImportString function in a distributed DataWindow environment, please keep the date display format the same at the client machine and Appeon Server. In addition, the date/time format configuration in AEM should keep the same as the system date/time configuration in Appeon Server.
- When calling PowerBuilder GetFullState and GetChanges, changed but unaccepted data in the DataWindow control have not been buffered and so they are treated as un-modified data. However, using the Appeon *appeondatawindow* and *appeondatastore*, changed but unaccepted data in the DataWindow control are treated as modified data.
- The return value of Appeon SetFullState may have different values from that of the PowerBuilder system SetFullState function.
- The Appeon GetChanges function always returns –1 if it fails. In PowerBuilder, the function can return more error numbers (-1, -2 and -3).
- The Appeon SetChanges can return –1 and –3, but cannot return 2 and –2.
- In PowerBuilder, the state information of a DataWindow/DataStore is initialized whenever you set its DataObject property. However, if using *appeondatawindow* and *appeondatastore*, the state information is only initialized when you change the DataObject property to a different DataWindow object.
- When applying Appeon SetChanges to a target DataWindow/DataStore, if a column of Char type in the source DataWindow/DataStore has defined more characters than its corresponding column in the target DataWindow/DataStore, characters from the source column that exceed the length limit of the target column will be truncated. In PowerBuilder the extra characters will be preserved.

Program access techniques

Top

Using OLE in an application

Using External functions

Using Run PowerScript function

Using OLE in an application Top

Appeon provides partial support for OLEObject objects: you can create an OLE object independent of an OLE control, connect to the OLEObject objects server application, and set properties for the object. The OLE server application may consist of either DLLs or separate EXE files. This support has the most functionality when used with Microsoft Word or Microsoft Excel.

The support is limited because OLE controls are currently unsupported, you can set or get properties, but cannot call functions via the server function. For more supported and unsupported details, refer to OLEObject object section.

Using External functions Top

Supported:

Data types of external function arguments (both reference and non-reference arguments) can be the following:

- Unsigned char, char, unsigned short, short, unsigned long, long, float, string, and Boolean

Data types of external function return values can be the following:

- Unsigned short, short, unsigned long, and long

It is supported to call external functions in Win32-based DLLs (including both the system DLLs and the user-defined DLLs).

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 98: The Windows system directory. Use the GetSystemDirectory function to get the path of this directory.
     or
   - Windows NT/2000: 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.

Note that the directory cannot be given an absolute path (a full path).

When the user makes changes to the environment variables, the environment variables take effect only after the user restarts the machine.

Unsupported:

For the unsigned long data type, it is unsupported to use an object handle or device handle as an argument to the External functions.

API for operations to file or directory access is not supported.

Using Run PowerScript function Top

Syntax: Run (*string* {, *windowstate* })

The value of the *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")

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 98: The Windows system directory. Use the GetSystemDirectory function to get the path of this directory.
     or
   - Windows NT/2000: 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.

Application enhancements and differences

[Enhancement: Client functions]

[Enhancement: Appeon security]

[Differences]

Enhancement: Appeon Client functions Top

Appeon provides a set of PowerBuilder global functions that can collect the information of an Internet Explorer client when a PowerBuilder application converts to the Web.

These functions are defined in the appeon_workarounds_js.pbl library. To use these functions in your PowerBuilder application, please add the library to the Library Search Path of the application.

The appeon_workarounds_js.pbl is located in the \appeon_workarounds\JS folder under the Appeon Developer directory.
(For example, C:\Program Files\Appeon\Developer\appeon_workarounds\JS)

As the following table shows, these functions, when used in a PowerBuilder application, return approximate values. When Appeon Developer converts the PowerBuilder application, these functions are identified and specially parsed into a set of JavaScript functions returning information to the Web application.

Table: Client Functions returning appropriate values

Functions	Return value when executed in PowerBuilder	Return value when executed in Web application
AppeonPopMenu (Datawindow adw_dw, Integer nx, Integer ny)	None. This function has no effect in PowerBuilder.	None. At the execution of the function, Appeon DataWindow menu is displayed at the specified position on the specified DataWindow control. AppeonPopMenu has higher priority than AppeonPopMenuOn.
AppeonPopMenuOn (Datawindow adw_dw, Boolean ab_show)	None. This function has no effect in PowerBuilder.	None. At the execution of the function, Appeon DataWindow menu pops up when you right click on the specified DataWindow.
AppeonGetAppeonUserName()	“”	The user name typed into the Appeon Web Login dialog.
AppeonGetBrowserVersion()	“”	Client Internet Explorer version
AppeonGetClientID()	Unique session identifier for PowerBuilder client	Unique session identifier for the Internet Explorer client
AppeonGetClientIP()	IP address of PowerBuilder client machine	IP address of the Internet Explorer client machine
AppeonGetClientType()	“PB”	“WEB”
AppeonGetOSType()	The type of the OS that runs the PowerBuilder client application.	The type of the OS that runs the Internet Explorer browser.

You can use these functions anywhere in your PowerBuilder application. However, the best way to use these

functions is to pass their return values into Appeon Server NVO components. Then, in the Appeon Server NVO components, the Internet Explorer client information (browser version and user name, for example) can be utilized to code more application features such as security authentication, auditing, logging, file operation, etc. This means we can write more scripts in NVO components for implementing more application features.

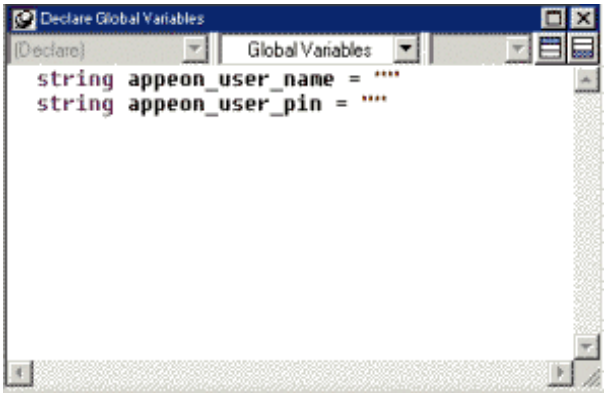
Enhancement: Incorporate Appeon security into PowerBuilder code Top

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 6.5 and 6.6 in the Appeon Enterprise Manager User Guide for more information. When using Appeon's built-in security, the user is prompted to enter a username and password in the Appeon Login Web dialog box.

It is necessary to pass the Appeon Web user name and password to the PowerBuilder application so that you can utilize them to implement script coded security features for your PowerBuilder application. To pass the Appeon Web user name and password, follow the steps below.

STEP 1: Define two global variables in your PowerBuilder application. The variables are String data type, and their names should be exactly *appeon_user_name* and *appeon_user_pin*.

When the PowerBuilder application is deployed to the Web, the Appeon system will automatically assign the values of the user name and password that the user uses to log in the Web application to *appeon_user_name* and *appeon_user_pin*.



STEP 2: Code the PowerBuilder application with the *appeon_user_name* and *appeon_user_pin* for carrying out security actions, such as saving account information in the INI upon the initial login or hiding/displaying menu items. The values of these two variables are passed to the application from the Appeon Login Web dialog box.

The Appeon Web user name and password can be set in the *Security* section of the Appeon Enterprise Manager (AEM).

Differences Top

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 measurement to Web measurement.
- Resizing does not take into account automatic changes that happen to the Web UI. For example, an automatic change can be a scrollbar added to a sheet. In this case, when the Resize event is triggered for the sheet, the result of the UI repaint does not include the scrollbar.
- The event LoseFocus will not be triggered by the focus changing between two windows that are not opened as sheets.

- When the Internet Explorer is refreshed, the Web application will restart.
- In the Web application, if the value of a property includes carriage return-newline character pairs (~r~n), the character pairs present differently than in PowerBuilder.
- When the user changes focus from the current window (non-sheet) to another window, the LoseFocus event of a control in the current window will not be triggered.
- It is unsupported to use the middle mouse key to operate the Web application.

PDF printing, image view and SaveAs

- The DataWindow PDF printing/image view/SaveAs in the PB application always prints the data that are currently displayed. The PDF printing/Image view/SaveAs in the Web application always prints the data that are specified in the Where clause of the current Table.Select property. In the same way, the DW Preview and PDF DataWindow preview display the data from the database.

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 at run-time does not work properly, 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.

Web applications and Internet Explorer

- While a Web application is being loaded, it is advised not to click the Refresh button in the Internet Explorer toolbar as this will cause some important JavaScript files not be loaded to the client. A runtime error will result.
- Clicking the “Refresh” button in the InternetExplorer browser to refresh the current page of the Web application is unsupported. By clicking the “Refresh” button, the Web application will be reloaded into the Internet Explorer.
- While a Web application is running and the TextSize in the Internet Explorer is set to Largest, Larger, Medium, Smaller, or Smallest, the menu of the Web application will change its font size accordingly, even though the other font size in the Web application remains the same.
- When users close the MDI window of the Web application, the Internet Explorer browser will close at the same time.
- If an application is an SDI application, when users close the SDI window, the Internet Explorer browser does not close at the same time.
- All Appeon system error messages appear in Internet Explorer status bar rather than in a dialog box.

Unsupported behavior

- Drag n’ Drop is not supported.
- Hot Keys, Shortcut Keys, Accelerator Keys are not supported.
- Keyboard Navigation: tab order and arrow keys are partially supported.

DataWindow enhancements and differences

Appeon Deployed DataWindows

Appeon DataWindow menu

User operation differences

Appeon deployed DataWindows

DataWindows in Appeon deployed applications include the following types:

HTML DataWindow - Implemented in XML for Grid, Tabular, and Freeform DataWindow Presentation Styles.

Image DataWindow - Implemented in GIF Graphic for DataWindow Presentation Styles other than Grid, Tabular, and Freeform, and for all DataWindow Presentation Styles via the DataWindow RMB Image View menu item.

PDF DataWindow - Implemented in PDF that can be printed or saved to a computer or LAN for emailing and later reference for all DataWindow Presentation Styles.

Exported DataWindow - Implemented in Microsoft Excel, Text, HTML and in the future PowerBuilder 9.0 XML for all DataWindow Presentation Styles.

Appeon DataWindow menu

Appeon DataWindow menu client functions

Starting from Appeon 3.0, Appeon DataWindow Menu cannot be enabled through AEM (Appeon Enterprise Manager). You can enable the Appeon DataWindow Menu by calling two Appeon client functons that are defined in *appeon_workarounds_js.pbl*:

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

Syntax: AppeonPopMenu (datawindow adw, Integer x, Integer y)

Argument Description:
adw The DataWindow control on which you want to pop up the Appeon DataWindow menu.
x The instance from the left edge of the DataWindow control
y 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.

AppeonPopMenuOn

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

Syntax: AppeonPopMenuOn (datawindow adw, Boolean bShow)

Argument Description:
adw The DataWindow control on which you want to show the Appeon DataWindow menu.
bShow 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.

Appeon DataWindow Menu Items for different types of DataWindow

Appeon DataWindow Menu Items for HTML DataWindows

Among the Appeon DataWindow Menu items for HTML DataWindows, Print View, Image View and Save As require a round trip to Appeon Server, and the others do not. Also, the implementation of the three menus Print View, Image View and Save As requires a transaction object to get data from the database.

Appeon DataWindow 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 View	Prints the DataWindow to a PDF file with PDFPrinter on Appeon Server. • At the click of the Print View button, a Print Setting dialog is displayed. Specify the paper size and orientation. • PDF printing is asynchronous, which means that after the user sends a printing request to the server (and before the server creates the PDF file), the user can continue with other operations. Meanwhile, users can view PDF printing results any time they wish. • A PDF Repository window displays at the right bottom corner of the Internet Explorer window that lists all the documents printed by Appeon Server PDFPrinter, unless the documents are deleted by the user from the Repository window.
Image View	View the DataWindow data as an image.
Save As	Saves the DataWindow data to a file in the specified format (Text, Excel, HTML Table).

Appeon DataWindow Menu items for Image DataWindows

Among the Appeon DataWindow Menu items for Image DataWindows, Save As and Print View require a round trip to Appeon Server, and the others do not. Also, the implementation of the two menu items Save As and Print View requires a transaction object to get data from the database.

Appeon DataWindow Menu Items	Functionality
Save As...	Saves the DataWindow data to a file in the specified format (Text, Excel, HTML Table).
Print View	Prints the DataWindow to a PDF file with PDFPrinter on Appeon Server. • Viewing the PDF format of an Image DataWindow is recommended if the display of the Image DataWindow is not perfect (i.e. if it has overlapping text or hidden lines). The PDF view can be more faithful to the original DataWindow. • Print View is asynchronous, which means that after the user sends a printing request to the server (and before the server creates the PDF file), the user can continue with other operations. Meanwhile, users can view PDF printing results any time they wish. • A PDF Repository window is displayed at the right bottom corner of the Internet Explorer window that lists all the documents printed by Appeon Server PDFPrinter, unless the documents are deleted by the user from the Repository window.
Zoom in	Zooms in on the DataWindow image.
Zoom out	Zooms out on the DataWindow image.

Other DataWindow enhancements include
Enlarge/decrease Grid DataWindow column size in the Web application.
Enlarge/decrease DropDownDataWindow column size in the Web application.

DataWindow user operation differences

APPEON

User interface differences

- *(This difference does not exist in versions earlier than Appeon 2.8 for PowerBuilder)* If a control in a DataWindow is placed above or behind another control in the DataWindow, the display of these controls will be different on the Web from in PowerBuilder. For example, if there is a Text control behind a DropDownDataWindow column in the DataWindow, the DropDownDataWindow field cannot be pulled down correctly.

- 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. Workaround: For the affected DataWindow, resize the UI layout in PowerBuilder to allow slightly more space for the items that are not fully displaying, until the corresponding Web UI layout is desirable. Use the Window or DataWindow preview functionality of the Appeon Developer to quickly view the changes and fine-tune the UI without the need to re-deploy the application.

- The number of data records displayed per page in deployed DataWindows may be different from that in PowerBuilder.

- If the AutoHScroll property of a DataWindow control is set to TRUE and the width of the DataWindow control is slightly larger than the DataObject, the DataWindow UI will be slightly distorted when the last column of the DataWindow gets focus. The UI display will revert back to its original setting once the focus changes to another column.

- In the Web application, controls or objects outside the displaying area of the band will still display in the DataWindow. This is will not happen in PowerBuilder.

- If the value of the argument is an array, the user cannot use dot notation to pass the argument to the function. Instead, the user must assign the array to another variable and then pass the variable to the function.

- The control(s) in a Web Image DataWindow do/does not display until data is retrieved in the DataWindow.

- In a deployed DataWindow, if empty space is updated into a column without any content, DataWindow will fail to update other data that the user re-inputs.

- In a deployed DataWindow, if empty spaces are updated into a column without any content, afterwards the DataWindow will fail to update another non-empty data entry that the user inputs. In addition, it is highly recommended that the user does not input data containing any empty spaces proceeding or following the data.

PDF printing/image view/SaveAs

- The PDF printing/Image view/SaveAs in the Web application always prints the data that is specified in Where clause of the current Table.Select property. This also applies to the Appeon Developer DataWindow Preview and PDF DataWindow Preview functionalities.

- Image DataWindows rely on the PowerBuilder DataWindow SaveAs function to save the data of a DataWindow data into a WMF file. The generated WMF image is resized before being loaded into the IE browser. As a result of the image resize process, image DataWindows may not display perfectly on the Web. Some characters may overlap each other or be truncated. For PDF DataWindows printing, DataWindow prints out what is displayed.

DataWindow Retrieve

- When a Specify Retrieval Arguments Dialog box displays and prompts the user to enter parameters for DataWindow retrieve, if a null value is used as the retrieval argument, Appeon will fail to retrieve the DataWindow data and will require the user to input some non-null parameter.

DataWindow modify/update

- If a cell in a DataWindow contains a value that is composed of only spaces and no other characters, the update to the database fails (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.

DropDownDataWindow and DropDownListBox

- In PowerBuilder, if the user presses the left mouse down in an expanded DropDownDataWindow, no item in the DropDownDataWindow is selected before the mouse button is released. The DropDownDataWindow remains in expanded status until the mouse button is released. However, in the Web application, no matter whether the user releases the mouse or not, the DropDownDataWindow is folded and an item is selected immediately when the user presses the left mouse down.

- If the user pulls down the DropDown ListBox or DropDown DataWindow and finds that the window is not large enough to fully display the dropdown area, the window will shift its content to display most of the dropdown area. The part that cannot be displayed will be truncated.

More behavioral differences

- In the deployed DataWindow, if the user clicks the UpArrow/DownArrow in the scrollbar, the DataWindow data scrolls one page upward/downward. If it is in PowerBuilder, the DataWindow data scrolls one row upward/downward.
- If there is no DataObject for a DataWindow, the GetFocus and LoseFocus events will still be triggered when the DataWindow is selected by clicking the Tab key. If the mouse clicked on the DataWindow, in PowerBuilder, no event is triggered. However, in the Web application, the Clicked event of the DataWindow container will be trigger.
- If the vertical scroll bar of an HTML DataWindow is dragged downward/upward, the DataWindow scrolls page by page instead of row by row as in PowerBuilder. This design is mainly for performance considerations: if an HTML DataWindow scrolls data row by row, the UI consistently refreshes, slowing down performance.
- In Grid DataWindow, the user cannot select all items in a column by clicking the column header.
- An Aggregate function executed in an Appeon Image DataWindow returns a calculation value based on the data in each page. In PowerBuilder, the calculation is based on all pages.

Object/control enhancements and differences

Enhancements

User operation differences

Object/control enhancements

In the Web MultiLineEdit control, the user can drag and drop the selected text.

Object/control user operation differences

User interface and operation differences

- In Web applications, the descendant controls are not always in front of the other non-descendant controls, and the graphical controls (RoundRectangle, Oval, Rectangle and Line) are not always at the bottom. This behavior is different in the PowerBuilder application.
- When the Pointer property of a control or an object is set to NULL or a non-cursor value, and the user moves the mouse over an Edit control, the cursor remains an arrow in PowerBuilder. However, it will change into an "I" in the Web application.
- In the Web application, if the pointer is specified as UpArrow!, the pointer displays SizeNS! when the user moves the pointer over a control.
- In PowerBuilder, while a DataWindow field is being edited, if the user drags the scroll bar, AcceptText is not executed for the field. In a Web DataWindow, AcceptText is executed for the field.

- In a container such as a UserObject, DataWindow, or Window, if the Enable property of a control in the container is set to FALSE, when the user clicks the control, a corresponding event of the container will be triggered in PowerBuilder. However, no event is triggered for the container.
- In a Web application, if control A has a parent (control B), it is unsupported to use control B in control A.
- The property for a control or an object cannot be NULL.

Open and close windows

- If the MDI window for a Web application closes, the whole Web application closes. This is different behavior from PowerBuilder. With a PowerBuilder application, if a Main window is opened before the MDI window is opened, the application returns to the Main window when the MDI window is closed.
- 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.
- It usually takes longer time to execute an event in Web applications than in C/S application. After clicking a button but before receiving a result, you may wonder whether the click was effective. You can check this using the following method: click the title bar of the current window and drag the bar with the mouse. If the bar can be dragged, no event is taking place and you should re-perform the action. If the bar cannot be dragged, the event is taking place and you must wait until the server responds.

Window behavioral differences

- It is unsupported to scroll a window by clicking PageUp/PageDown, UpArrow/downArrow, or LeftArrow/RightArrow keys.
- It is unsupported to switch between windows by pressing the Ctrl + Tab key.
- In PowerBuilder, if the user right-clicks a sheet window that is not the currently active window, the sheet will not be activated. However, in the Web application, the sheet can be activated.
- If the user double clicks within a PowerBuilder window, the DoubleClicked event is triggered once and the Clicked event is triggered twice. If the user double clicks in a deployed Web application, only the DoubleClicked event will be triggered.
- When the Enabled property for a control or object is FALSE, there may be different behaviors between the PowerBuilder application and the Web application. For example, when the Enabled property for a control is set to FALSE, some window events will be triggered in PowerBuilder but not in the Web application.
- When you perform a window preview from the Appeon Developer toolbar, you may find different behaviors on the Web from PowerBuilder. For example, the window size in the preview is different from that in PowerBuilder. It is normal behavior with the preview. In the Web application, the window size will be identical to the window in PowerBuilder.
- In a Web application, if a Response window is opened, the Timer event of other windows cannot be triggered. In PowerBuilder, the Timer event can still be triggered.
- The following sequence is not well-supported:

  Step 1: In response to a user operation in a response window, a sheet opens.
  Step 2: The application allows the response window and the sheet to be open simultaneously.
  Step 3: During a user operation in the sheet, the response window is closed.

  It is recommended that the sequence be changed as follows: In response to a user operation in a response window, a sheet opens. Simultaneously, the response window is closed.

Focus change between controls

- When the focus is passed from control A to control B and then back to control A, the previously selected text in control A will be unselected even though the the window has focus again.
- There can only be one selected control at a given time. It is because in the Internet Explorer browser, a control can only be displayed as selected once it has focus.
- In PowerBuilder, when a control does not have focus, the execution of the AddItem or DeleteItem function on the control will make the control have focus. In the Web application, the control will not get focus by executing the AddItem or DeleteItem function.
- In PowerBuilder, when the Enabled or Visible property of a control or an object that has focus is changed from TRUE to FALSE, the focus will move to the next control or object. However, in a Web application, the focus will still remain with the control or object.
- If there is any script in AddItem or InsertItem function for changing focus, the script will have no effect.

CommandButton

- When right-mouse clicking over a CommandButton, the border of the control will be a dotted line, but this does not mean that the control has focus.

DropDownDataWindow

- Filter criteria for retrieving data into a DataWindow or DropDownDataWindow are case-sensitive, while in PowerBuilder, filter criteria for retrieving data into a DropDownDataWindow are case-insensitive.

- When the AllowEdit property is set to YES for a DropDownDataWindow, if the listbox of the DropDownDataWindow is folded and a value is entered in the field, the value should match one of the items in the list. If not, no item will be selected in the DropDownDataWindow.

DropdownListBox, DropdownPictureListBox, ListBox or PictureListBox

- If a DropdownListBox or DropdownPictureListBox has focus, using the up/down arrow does not scroll through the entries in the list, and pressing a letter does not advance to the first value that begins with that letter. The user must mouse click and select the entry.

- 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.

- In a Web application, when the ExtendedSelect property or the MultiSelect property is set to TRUE, the user can select multiple items by using click/ shift+ click to select a sequential group of items or using ctrl+ click on multiple items. However, the user cannot select multiple items with only keystrokes (for example, ctrl + arrow keys) or mouse.

- If the HScrollBar is set to TRUE for the ListBox control, the user scrolls right or left in the ListBox control with the arrow keys in PowerBuilder. However, in a Web application, the user can scroll right or left only by clicking. The left/right arrow keys can only be used to select items in the ListBox.

EditMask

- Support for user interaction is limited with the EditMask control or the EditMask edit style. For example, if the user selects all the text in the edit box and presses the Back Space key or the Delete key in an EditMask control, the text will not be cleared as expected.

- In the Web application, the user has to click twice in an EditMask control to locate where to input data.

- In a Web application, if the user clicks on an arrow in a spin and then drags elsewhere, the spin stays in the dropdown state and does not rise.

- If the mask of an EditMask control is set to “####”, the length of the string in EditMask is set to four digits. When the user inputs data that has fewer than four digits, EditMask does not accept the data. However, in a Web application, extra empty spaces will be added to the left of the data. As a result, AcceptedText is unable to detect the illegal data.

- When the mask of an EditMask control contains "!", "^", "#", or "a", the value input in the control will not be displayed.

- In a Web application, if the mask of an EditMask control is set to “####0”, when the user clicks on the control, the control displays a “0”, but in PowerBuilder, it does not display anything.

Menu

- Only one menu can be displayed at any given time. Multiple toolbars can be displayed simultaneously, but the docking bar is not supported. In addition, toggle is not supported for the toolbars.

MultiLineEdit

- If the user right clicks the scrollbar in the Web MultiLineEdit control, the action triggers the RbuttonDown event. In PowerBuilder, no event is triggered.
- In PowerBuilder, when scrolling is disabled and the Limit property is set to 0, the user can only type in a maximum number of characters in the control. However, in the Web application, the user can type in unlimited number of characters in the control.

RadioButton

- The user cannot change the value of a RadioButton control by pressing the Left/Right key.

StaticText

- For StaticText control, if the FocusRectangle is set to TRUE, the focus rectangle will not display in the Web application unless the user clicks the Tab key to activate the rectangle first.
- If the StaticText control is longer than the specified control width, the text will automatically display in more than one line in the Web application.

Tab

- If the user clicks the blank area in the header of a Web Tab, the Clicked or RightClicked event will not be triggered. The user must click the tab page area to trigger the event. This is different from PowerBuilder.
- The tab page headers will only display in one line in the Web application even though they are displayed in multiple lines the PowerBuilder application.

TreeView

- With the CheckBoxes property of a TreeView control set to TRUE, if the highlighted TreeView item does not display in the UI and the user clicks on the checkbox of an item, this action does not change the value of the checkbox. In PowerBuilder, the value of the checkbox does change; the Web TreeView will scroll upward or downward until the highlighted item is displayed. In such a case, in order to change the checkbox value of the TreeView item, it is recommended to select the item first by clicking on the item, and then clicking the checkbox of the item.
- If the property FullRowSelect or Haslines of a Web TreeView is set to TRUE, when the user performs a right mouse click at or near one of the selected items, the text color of all the currently-selected items in the TreeView turns black.
- If an item in a Web TreeView control has a picture assigned to it, the size of the picture varies with the TextSize property of the item.

Controls

Supported

The following list is the controls Appeon supports:

CheckBox	CommandButton	DataWindow	DropDownListBox
DropDownPictureListBox	EditMask	GroupBox	Line
ListBox	ListView	MultiLineEdit	Picture
PictureButton	PictureHyperLink	PictureListBox	RadioButton
Rectangle	SingleLineEdit	StaticHyperLink	StaticText
Tab	TreeView		

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

- For DataWindow control, please refer to the DataWindow section.

Unsupported

The following list is the controls Appeon does not support:

Graph	HProgressBar	HScrollBar	HTrackBar
OLEControl	OLECustomControl	Oval	RichTextEdit
RoundRectangle	VProgressBar	VScrollBar	VTrackBar

- The 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!")
- It is unsupported to get the classname of a system class type variable with the function ClassName. The return value will be incorrect on the Web. For example:
  Transaction gtr_trans //gtr_trans is a system class type variable
  String ls_arg
  gtr_trans = create transaction
  ls_arg = gtr_trans.ClassName() // Unsupported

Important Requirements

1. Supported controls and features of supported controls must be created in a PowerBuilder painter.
2. Supported controls must be static. They cannot be dynamically created (e.g. using the CREATE statement).
3. It is unsupported to create a new control in a child object by copying and pasting an existing control in the PowerBuilder painter, if the existing control is inherited from the parent object.
4. Note the front to back order of controls: The GroupBox control will always be behind all other controls in the container where it is placed.

CheckBox control

Properties | Events | Functions

Important Requirements

The CheckBox control must be created in a PowerBuilder painter.
The CheckBox control must be static. It cannot be dynamically created (for example, using the CREATE statement).

Properties for CheckBox control

Properties | Events | Functions

Property	Support Level	Example Code
BackColor	Supported	cbx_1.BackColor = ll_value cbx_1.BackColor = RGB(192,192,192)
BorderStyle	Supported	cbx_1.BorderStyle=StyleLowered! cbx_1.BorderStyle=StyleRaised!
Checked	Supported	cbx_1.Checked=TRUE cbx_1.Checked = lb_value
Enabled	Supported	cbx_1.Enabled=TRUE cbx_1.Enabled = lb_value
FaceName	Supported	cbx_1.FaceName = "Arial" cbx_1.FaceName = ls_value
Height	Supported	cbx_1.Height=889 cbx_1.Height = li_value
Italic	Supported	cbx_1.Italic=TRUE cbx_1.Italic = lb_value
Pointer	Supported	cbx_1.Pointer=ls_value The Pointer property is partially supported: • Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow! • Unsupported values of the Pointer property: AppStarting!, Help!, Icon!, No!, and SizeNS! • User-defined pointers are not supported. • All unsupported pointers are replaced with Arrow! in the Web application.
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
Underline	Supported	cbx_1.Underline=TRUE cbx_1.Underline = lb_value
Visible	Supported	cbx_1.Visible=TRUE cbx_1.Visible = lb_value cbx_1.Y = li_value

		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
Automatic	Unsupported	Note: the Automatic property cannot be changed, and will always be set to TRUE.
BringToTop	Unsupported	
ClassDefinition	Unsupported	
DragAuto	Unsupported	
DragIcon	Unsupported	
FontCharSet	Unsupported	
FontFamily	Unsupported	
FontPitch	Unsupported	
LeftText	Unsupported	
RightToLeft	Unsupported	
ThirdState	Unsupported	
ThreeState	Unsupported	

Events for CheckBox control

Properties | Events | Functions

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

DragLeave	Unsupported
DragWithin	Unsupported
Help	Unsupported
Other	Unsupported

Functions for CheckBox control

Properties | Events | Functions

Function	Support Level	Example Code
ClassName	Supported	ls_return = cbx_test.ClassName()
GetParent	Supported	lobj_parent = this.GetParent()
Hide	Supported	li_return = cbx_test.Hide()
Move	Supported	cbx_1.Move(100,100)] li_return = cbx_test.Move(li_x,li_y)
PostEvent	Supported	cbx_1.PostEvent(RButtonDown!) lb_return = cbx_test.PostEvent(event)
Resize	Supported	cbx_1.Resize(100,20) li_return = cbx_test.Resize(li_x, li_y)
SetFocus	Supported	cbx_1.SetFocus() li_return = cbx_test.SetFocus() Unsupported: SetFocus(cbx_test)
SetPosition	Supported	li_return = cbx_test.SetPosition(position)
Show	Supported	li_return = cbx_test.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

Drag	Unsupported	
GetContextService	Unsupported	
PointerX	Unsupported	
PointerY	Unsupported	
Print	Unsupported	
SetRedraw	Unsupported	

CommandButton control

Properties | Events | Functions

Important Requirements

The CommandButton control must be created in PowerBuilder painter.
The CommandButton control must be static. It cannot be dynamically created (for example, using the CREATE statement).

Properties for CommandButton control

Properties | Events | Functions

Property	Support Level	Example Code
Enabled	Supported	cb_test.Enabled = lb_value
FaceName	Supported	cb_test.FaceName = "Arial" cb_test.FaceName = "Courier" cb_test.FaceName = "Fixedsys"
Height	Supported	cb_test.Height = li_value
Italic	Supported	cb_test.Italic = lb_value
Pointer	Supported	cb_1.Pointer = ls_value The Pointer property is partially supported: • Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow! • Unsupported values of the Pointer property: AppStarting!, Help!, Icon!, No!, and SizeNS! • User-defined pointers are unsupported. • All unsupported pointers are replaced with Arrow! in the Web application.
TabOrder	Supported	cb_test.TabOrder = 20
Tag	Supported	cb_test.Tag = ls_value
Text	Supported	cb_test.Text = ls_value
TextSize	Supported	cb_test.TextSize = li_value
Underline	Supported	cb_test.Underline = TRUE cb_test.Underline = lb_value
Visible	Supported	cb_test.Visible = TRUE cb_test.Visible = lb_value

Weight	Supported	cb_test.Weight = 700 cb_test.Weight = li_value
Width	Supported	cb_test.Width = 750 cb_test.Width = li_value
X	Supported	cb_test.X = 280 cb_test.X = li_value
Y	Supported	cb_test.Y = 1280 cb_test.Y = li_value
BringToTop	Unsupported	
Cancel	Unsupported	
ClassDefinition	Unsupported	
Default	Unsupported	
DragAuto	Unsupported	
DragIcon	Unsupported	
FontCharSet	Unsupported	
FontFamily	Unsupported	
FontPitch	Unsupported	

Events for CommandButton control

Properties | Events | Functions

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

Functions for CommandButton control

Properties | Events | Functions

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

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

Drag	Unsupported	
GetContextService	Unsupported	
PointerX	Unsupported	
PointerY	Unsupported	
Print	Unsupported	
SetRedraw	Unsupported	

DropDownListBox control

Properties | Events | Functions

Important Requirements

The DropDownListBox control must be created in PowerBuilder painter.
The DropDownListBox control must be static. It cannot be dynamically created (for example, using the CREATE statement).

For the DropDownListBox, the current selected item in the textbox will never change focus when the user pageup or pagedown.

Properties for DropDownListBox control

Properties | Events | Functions

Property	Support Level	Example Code
AllowEdit	Supported	ddlb_1.AllowEdit = TRUE ddlb_1.AllowEdit = lb_value The cursor over the DropDownListBox control will always shapes in IBeam, no matter the AllowEdit property is enabled or not.
BackColor	Supported	ddlb_1.BackColor = RGB(0, 0, 255) ddlb_1.BackColor = ll_value
Border	Supported	ddlb_1.Border = TRUE ddlb_1.Border = lb_value
BorderStyle	Supported	ddlb_1.BorderStyle = StyleLowered!
Enabled	Supported	ddlb_1.Enabled=TRUE ddlb_1.Enabled = lb_value
FaceName	Supported	ddlb_1.FaceName = "Arial" ddlb_1.FaceName = ls_value
Height	Supported	ddlb_1.Height = 750 ddlb_1.Height = li_value If the height of a ListBox is set to a value greater than necessary (for displaying the items), the user may find extra space in the Web ListBox.
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 = ls_value The Pointer property is partially supported: • Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow! • Unsupported values of the Pointer property: AppStarting!, Help!, Icon!, No!, and SizeNS! • User-defined pointers are not supported.

- All unsupported pointers are replaced with Arrow! in the Web application.

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)
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
Accelerator	Unsupported	
AutoHScroll	Unsupported	
BringToTop	Unsupported	
ClassDefinition	Unsupported	
DragAuto	Unsupported	
DragIcon	Unsupported	
FontCharSet	Unsupported	
FontFamily	Unsupported	
FontPitch	Unsupported	
ImeMode	Unsupported	
RightToLeft	Unsupported	
ShowList	Unsupported	

Note: 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.

Events for DropDownListBox control

Properties | Events | Functions

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

Functions for DropDownListBox control

Properties | Events | Functions

Function	Support Level	Example Code
AddItem	Supported	li_return = ddlb_1.AddItem(item)
ClassName	Supported	ls_return = ddlb_1.ClassName()
DeleteItem	Supported	li_return = ddlb_1.DeleteItem(index)
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)
PostEvent	Supported	ddlb_1.PostEvent(Clicked!)
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)
SetFocus	Supported	li_return = ddlb_1.SetFocus() Unsupported: SetFocus(ddlb_test)
SetPosition	Supported	ddlb_1.SetPosition(ToTop!) li_return = ddlb_1.SetPosition(position)
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 this.typeof()=DropDownListBox! then
		  messagebox("show"," DropDownListBox!")
		end if

Clear	Unsupported	
Copy	Unsupported	
Cut	Unsupported	
DirList	Unsupported	
DirSelect	Unsupported	
Drag	Unsupported	
GetContextService	Unsupported	
Paste	Unsupported	
PointerX	Unsupported	
PointerY	Unsupported	
Position	Unsupported	
Print	Unsupported	
ReplaceText	Unsupported	

SelectedLength	Unsupported
SelectedStart	Unsupported
SelectedText	Unsupported
SelectText	Unsupported
SetRedraw	Unsupported

DropDownPictureListBox control

Properties | Events | Functions

APPEON

Important Requirements

The DropDownPictureListBox control must be created in PowerBuilder painter.
The DropDownPictureListBox control must be static. It cannot be dynamically created (for example, using the CREATE statement).

For the DropDownPictureListBox, the current selected item in the textbox will never change focus when the user uses pageup and pagedown.

Properties for DropDownPictureListBox control

Properties | Events | Functions

APPEON

Property	Support Level	Example Code
AllowEdit	Supported	ddplb_1.AllowEdit = TRUE ddplb_1. AllowEdit=lb_value
BackColor	Supported	ddplb_1.BackColor = ll_value ddplb_1.BackColor = RGB (0,0,255)
Border	Supported	ddplb_1.Border = TRUE ddplb_1.Border = lb_value
BorderStyle	Supported	ddplb_1.BorderStyle = StyleLowered!
Enabled	Supported	ddplb_1.Enabled = TRUE ddplb_1.Enabled = lb_value
FaceName	Supported	ddplb_1.Facename = ls_value
Height	Supported	ddplb_1.Height = li_value Note: if the Height is more than necessary for displaying the items in the control, the Web UI may differ to that in PowerBuilder.
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 Note: this property cannot be dynamically changed.
PictureName[ ]	Supported	ls_value = ddplb_1.PictureName[1] Note: PictureName cannot contain any double-byte characters.
Pointer	Supported	Ddplb_1.Pointer = "Beam!" The Pointer property is partially supported: ● Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow! ● Unsupported values of the Pointer property: AppStarting!, Help!, Icon!, No!, and SizeNS! ● User-defined pointers are not supported. ● All unsupported pointers are replaced with Arrow! in the Web application.
Sorted	Supported	ddplb_1.Sorted = lb_value
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
Accelerator	Unsupported	
AutoHScroll	Unsupported	
BringToTop	Unsupported	
ClassDefinition	Unsupported	
DragAuto	Unsupported	
DragIcon	Unsupported	
FontCharSet	Unsupported	
FontFamily	Unsupported	
FontPitch	Unsupported	
ImeMode	Unsupported	
PictureHeight	Unsupported	
PictureMaskColor	Unsupported	
PictureWidth	Unsupported	
RightToLeft	Unsupported	
ShowList	Unsupported	

Events for DropDownPictureListBox control

Properties | Events | Functions

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

Functions for DropDownPictureListBox control

Properties | Events | Functions

Function	Support Level	Example Code
AddItem	Supported	li_return = ddplb_1.AddItem(item[, pictureindex])
AddPicture	Supported	li_return = ddplb_1.AddPicture(picturename)
ClassName	Supported	ls_return = ddplb_1.ClassName()
DeleteItem	Supported	li_return = ddplb_1.DeleteItem(index)
DeletePicture	Supported	li_return =ddplb_1.DeletePicture(index)
DeletePictures	Supported	li_return =ddplb_1.DeletePictures()
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	li_return = ddplb_1.InsertItem(item[, pictureindex], index)
Move	Supported	li_return = ddplb_1.Move(li_x,li_ y)
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)
SetFocus	Supported	li_return = ddplb_1.SetFocus() Unsupported: SetFocus(ddplb_test)
SetPosition	Supported	li_return = ddplb_1.SetPosition(position)
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 this.typeof()=DropDownPictureListBox! Then
		ls_return ='DropDownPictureListBox!'
		else
		ls_return ='isvalid'
		end if

Clear	Unsupported	
Copy	Unsupported	
Cut	Unsupported	
DirList	Unsupported	
DirSelect	Unsupported	
Drag	Unsupported	
GetContextService	Unsupported	
Paste	Unsupported	
PointerX	Unsupported	
PointerY	Unsupported	
Position	Unsupported	
Print	Unsupported	
ReplaceText	Unsupported	
SelectedLength	Unsupported	
SelectedStart	Unsupported	
SelectedText	Unsupported	
SelectText	Unsupported	
SetRedraw	Unsupported	

EditMask control

Properties | Events | Functions

APPEON

Important Requirements

The EditMask control must be created in PowerBuilder painter.
The EditMask control must be static. It cannot be dynamically created (for example, using the CREATE statement).

EditMask mask settings

Appeon supports all the EditMask mask settings except for the ones shown in the following table:

MaskDataType	Mask (unsupported format)	Different display in Web from PowerBuilder
DateMask!	yyyy~ymm~mdd~d	Input: 1998/01/30 Web: 01/30/1998 PowerBuilder: 1998y01m30d
	mmm-dd-yyyy	Input: 1998/01/30 Web: 30-01-1998

		PowerBuilder: JAN-30-1998
DateTimeMask!	yyyy~ymm~mdd~d hh~h	Input: 1998/01/30 12:58:59 Web: 01/30/1998 12:58:59:000 PowerBuilder: 1998y01m30d 12h
TimeMask!	hh~hmm~mss~s	Input: 12:58:59 Web: 12:58:59:000 PowerBuilder: 12h:58m:59s
StringMask!	xxx~xxxx~xxxx	

Note: (1) It is unsupported to use plus sign (+) operator as delimiter in a date or datetime value. For example, if “yyyy-mm-dd" is set as the mask, value "004-01+01" is interpreted as 2004-01-01 in PowerBuilder but 0000-00-00 on the Web. (2) NumericMask should not contain parenthesis or spaces.

Properties for EditMask control

Properties | Events | Functions

APPEON

Property	Support Level	Example Code
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 = StyleShadowBox!
DisplayOnly	Supported	em_1.DisplayOnly = TRUE em_1.DisplayOnly = lb_value
Enabled	Supported	em_1.Enabled = TRUE em_1.Enabled = lb_value
FaceName	Supported	em_1.FaceName = ls_value
Height	Supported	em_1.Height = li_value
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 that the Limit property cannot be dynamically modified.
Mask	Supported	em_1.MaskDataType =StringMask em_1.Mask = '@@@,@@' Note that the Mask property is partially supported. The following restrictions are: • In PowerBuilder, "~x" is the same as "x". On the Web, if

		the format is set to "~x", data will be displayed in the default string format. • For the data format "###0;($###0)": "-5", in PowerBuilder, the display is similar to "($0005)"; on the Web, the display is similar to "($5)". • For the data format "###0.00;($###0.00)": "-n", in PowerBuilder, the display is similar "($0005.00)"; on the Web, the display is similar to "($5.00)". • The Mask property for EditMask can only be set in the PowerBuilder Painter. • If the format for Date, Time, DateTime, Currency mask is not specified, PowerBuilder applications will read the local settings from the client machine, while Web applications will read the settings in AEM (Application Enterprise Manager). • It is unsupported to set the mask directly as “[Date]”, “[Time]” or “[DateTime]”. • If the Mask is set to “####-##-####” and the MaskDataType is StringMask!, when the user types “1234” in the SingleLineEdit column and then presses Delete after moving the cursor in front of “3”, the user cannot type in any data as the cursor is in front of the separator. However, on the Web, the user can still type in a data which will replace the number followed. • For more information, refer to the table in the EditMask mask settings in Edit Mask important requirements. • When the spin control and the text cannot each fully display within the EditMask control at the same time, the spin control will display fully and text will display partially display in PowerBuilder. However, in the Web application, the text will fully display and the spin control will partially display.
MaskDataType	Supported	em_1.MaskDataType =StringMask! em_1.MaskDataType =DateTimeMask! m_1.MaskDataType =NumericMask! Note: this property can be set only in painter.
MinMax	Supported	em_1.MinMax = ("100 ~~ 10000")
Pointer	Supported	em_1.Pointer = ls_value Note that the Pointer property is partially supported. The following are limitations: • Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow!

		• Unsupported values of the Pointer property: AppStarting!, Help!, Icon!, No!, and SizeNS! • User-defined pointers are not supported. • All unsupported pointers are replaced with Arrow! in the Web application.
Spin	Supported	em_1.Spin = TRUE em_1.Spin = FALSE The height of the spin for an EditMask control should be no larger than the height of the control. Otherwise, the spin will be cut off by the border of the EditMask control (on the Web, the EditMask control is actually a container for the spin).
TabOrder	Supported	em_1.TabOrder = 50
Tag	Supported	em_1.Tag = ls_value
Text	Supported	em_1.Text = ls_value Note: it is unsupported to modify text that contains ESC character.
TextColor	Supported	em_1.TextColor = ll_value em_1.TextColor = RGB(192,192,192)
TextCase	Supported	em_1.TextCase=AnyCase! 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
Visible	Supported	em_1.Visible = lb_value
Weight	Supported	em_1.Weight = li_value
Width	Supported	em_1.Width = li_value Note: if the Text property is not null, the setting for the Width property may not reach the desired effect.
X	Supported	em_1.X = li_value
Y	Supported	em_1.Y = li_value
Accelerator	Unsupported	
Alignment	Unsupported	
AutoHScroll	Unsupported	
AutoSkip	Unsupported	

AutoVScroll	Unsupported
BringToTop	Unsupported
ClassDefinition	Unsupported
DisplayData	Unsupported
DragAuto	Unsupported
DragIcon	Unsupported
FontCharSet	Unsupported
FontFamily	Unsupported
FontPitch	Unsupported
HScrollBar	Unsupported
HideSelection	Unsupported
IgnoreDefaultButton	Unsupported
ImeMode	Unsupported
RightToLeft	Unsupported
TabStop[ ]	Unsupported
UseCodeTable	Unsupported
VScrollBar	Unsupported

Events for EditMask control

Properties | Events | Functions

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor
GetFocus	Supported	GetFocus
LoseFocus	Supported	LoseFocus
Modified	Supported	Modified Note: unlike in PowerBuilder, in JavaScript the event cannot be triggered for the Web control if (1) the text in the control has not been changed; or (2) the user tried to trigger the event by pressing the Enter key.
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
DragDrop	Unsupported	
DragEnter	Unsupported	
DragLeave	Unsupported	
DragWithin	Unsupported	
Help	Unsupported	
Other	Unsupported	

Functions for EditMask control

Properties | Events | Functions

Function	Support Level	Example Code
ClassName	Supported	ls_returnvalue = em_1.ClassName()
Clear	Supported	li_returnvalue = em_1.Clear()
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)
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.
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)
SetMask	Supported	li_returnvalue = em_1.SetMask(maskdatatype, mask)
SetPosition	Supported	li_returnvalue = em_1.SetPosition(position)

Show	Supported	li_returnvalue = em_1.Show()
TextLine	Supported	ls_returnvalue = em_1.TextLine()
TriggerEvent	Supported	li_returnvalue = em_1.TriggerEvent()

TypeOf	Supported	if this.typeof()=editmask! Then
		  ls_returnvalue ='editmask!'
		else
		  ls_returnvalue ='invalid!'
		end if

CanUndo	Unsupported	
Copy	Unsupported	
Cut	Unsupported	
Drag	Unsupported	
GetContextService	Unsupported	
Paste	Unsupported	
PointerX	Unsupported	
PointerY	Unsupported	
Position	Unsupported	
Print	Unsupported	
Scroll	Unsupported	
SelectedLine	Unsupported	
SetRedraw	Unsupported	
Undo	Unsupported	

GroupBox control

Properties | Events | Functions

Important Requirements

The GroupBox control must be created in PowerBuilder painter.
The GroupBox control must be static. It cannot be dynamically created (for example, using the CREATE statement).

The GroupBox control will always be behind all other controls in the container where it is placed. If a GroupBox has a RadioButton control placed in it, the GroupBox and the RadioButton control cannot be placed behind another GroupBox.

Properties for GroupBox control

Properties | Events | Functions

Property	Support	Example Code

	Level	
Enabled	Supported	gb_test.Enabled = lb_value
FaceName	Supported	gb_test.FaceName = ls_value
Height	Supported	gb_test.Height = li_value
Italic	Supported	gb_test.Italic = lb_value
Pointer	Supported	gb_test.Pointer= 'Beam!' //Specifies the name of the stock pointer or the file containing the pointer used for the control. The Pointer property is partially supported: • Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow! • Unsupported values of the Pointer property: AppStarting!, Help!, Icon!, No!, and SizeNS! • User-defined pointers are not supported. • All unsupported pointers are replaced with Arrow! in the Web application.
TabOrder	Supported	gb_test.TabOrder = 50
Tag	Supported	gb_test.Tag = ls_value
Text	Supported	gb_test.Text = ls_value
TextColor	Supported	gb_test.TextColor = ll_value gb_test.TextColor = RGB(192,192,192)
TextSize	Supported	gb_test.TextSize = li_value
Underline	Supported	gb_test.Underline = lb_value
Visible	Supported	gb_test.Visible = lb_value
Weight	Supported	gb_test.Weight = li_value
Width	Supported	gb_test.Width = li_value
X	Supported	gb_test.X = li_value
Y	Supported	gb_test.Y = li_value
BackColor	Unsupported	
BorderStyle	Unsupported	
BringToTop	Unsupported	

ClassDefinition	Unsupported
DragAuto	Unsupported
DragIcon	Unsupported
FontCharSet	Unsupported
FontFamily	Unsupported
FontPitch	Unsupported
RightToLeft	Unsupported

Events for GroupBox control

Properties | Events | Functions

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

Functions for GroupBox control

Properties | Events | Functions

Function	Support Level	Example Code
ClassName	Supported	ls_returnvalue = gb_test.ClassName()
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = gb_test.GetParent()
Hide	Supported	li_returnvalue = gb_test.Hide()
Move	Supported	gb_test.Move(200,500) li_returnvalue = gb_test.Move(li_xpos, li_ypos)
PostEvent	Supported	lb_returnvalue = gb_test.PostEvent(event)
Resize	Supported	li_returnvalue = gb_test.Resize(li_width,li_height)
SetFocus	Supported	li_returnvalue = gb_test.SetFocus( ) Unsupported: SetFocus(gb_text)
SetPosition	Supported	li_returnvalue = gb_test.SetPosition(position)
Show	Supported	li_returnvalue = gb_test.Show()
TriggerEvent	Supported	li_returnvalue = gb_test.TriggerEvent(event)

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

Drag	Unsupported	
GetContextService	Unsupported	
PointerX	Unsupported	
PointerY	Unsupported	
Print	Unsupported	
SetRedraw	Unsupported	

Line control

Properties | Events | Functions

Important Requirements

The Line control must be created in PowerBuilder painter.
The Line control must be static. It cannot be dynamically created (for example, using the CREATE statement).

Properties for Line control

Properties | Events | Functions

Property	Support Level	Example Code
BeginX	Supported	ln_test.BeginX = li_value
BeginY	Supported	ln_test.BeginY = li_value
EndX	Supported	ln_test.EndX = li_value
EndY	Supported	ln_test.EndY = li_value
LineColor	Supported	ln_test.LineColor = ll_value ln_test.LineColor = RGB(192,192,192) It is unsupported to set the value of the LineColor property to Desktop.
LineStyle	Supported	ln_test.LineStyle = Continuous! ln_test.LineStyle = Dash! ln_test.LineStyle = DashDot! ln_test.LineStyle = DashDotDot! ln_test.LineStyle = Dot! ln_test.LineStyle = Transparent!
LineThickness	Supported	ln_test.LineThickness = li_value

Tag	Supported	ln_test.Tag = ls_value
Visible	Supported	ln_test.Visible = TRUE ln_test.Visible = lb_value
ClassDefinition	Unsupported	

Events for Line control

Properties | Events | Functions

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor

Functions for Line control

Properties | Events | Functions

Function	Support Level	Example Code
ClassName	Supported	ls_returnvalue = ln_test.ClassName()
GetParent	Supported	lobj_returnvalue = ln_test.GetParent( )
Hide	Supported	li_returnvalue = ln_test.Hide()
Move	Supported	li_returnvalue = ln_test.Move(li_xpos, li_ypos)
Show	Supported	li_returnvalue = ln_test.Show()

TypeOf	Supported	if ln_test.typeof() = Line! Then
		  ls_returnvalue = 'Line!'
		else
		  ls_returnvalue = 'Invalid'
		end if

GetContextService	Unsupported	
Resize	Unsupported	

ListBox control

Properties | Events | Functions

Important Requirements

The ListBox control must be created in PowerBuilder painter.
The ListBox control must be static. It cannot be dynamically created (for example, using the CREATE statement).

For the ListBox, the current selected item in the textbox will never change focus when the user uses pageup or pagedown.

Properties for ListBox control

Properties | Events | Functions

Property	Support Level	Example Code
BackColor	Supported	lb_1.BackColor = ll_value lb_1.BackColor = RGB(192,192,192)
Border	Supported	lb_1.Border = lb_value
BorderStyle	Supported	lb_1.BorderStyle = StyleLowered!
DisableNoScroll	Supported	lb_1.DisableNoScroll = FALSE lb_1.DisableNoScroll = lb_value
Enabled	Supported	lb_1.Enabled = lb_value
ExtendedSelect	Supported	lb_1.ExtendedSelect = lb_value
FaceName	Supported	lb_1.FaceName = ls_value
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]
MultiSelect	Supported	lb_1.MultiSelect = lb_value
Pointer	Supported	lb_1.Pointer = ls_value The Pointer property is partially supported: • Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow! • Unsupported values of the Pointer property: AppStarting!, Help!, Icon!, No!, and SizeNS! • User-defined pointers are not supported. • All unsupported pointers are replaced with Arrow! in the Web application.
Sorted	Supported	lb_1.Sorted = TRUE Note: there is a difference when comparing the items. In PowerBuilder, a is less than A, but in the converted Web application, a is greater than A.

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
Accelerator	Unsupported	
BringToTop	Unsupported	
ClassDefinition	Unsupported	
DragAuto	Unsupported	
DragIcon	Unsupported	
FontCharSet	Unsupported	
FontFamily	Unsupported	
FontPitch	Unsupported	
RightToLeft	Unsupported	
TabStop[ ]	Unsupported	

Events for ListBox control

Properties | Events | Functions

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor
DoubleClicked	Supported	DoubleClicked
GetFocus	Supported	GetFocus
LoseFocus	Supported	LoseFocus

RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
SelectionChanged	Supported	SelectionChanged(index)
DragDrop	Unsupported	
DragEnter	Unsupported	
DragLeave	Unsupported	
DragWithin	Unsupported	
Help	Unsupported	
Other	Unsupported	

Functions for ListBox control

Properties | Events | Functions

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)
Find Item	Supported	li_returnvalue = lb_1.FindItem(text, index) Note: in PowerBuilder this function is not case sensitive, but it is case sensitive in JavaScript.
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = lb_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))
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)
SetState	Supported	li_returnvalue = lb_1.SetState(index, lb_y)
SetTop	Supported	li_returnvalue = lb_1.SetTop(index) Note: the SetTop function is supported only when the specified item is scrolled into the view of the listbox.
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 this.typeof() = listbox! Then
		ls_returnvalue ='listbox!'
		else
		ls_returnvalue ='isvalid'
		end if

DirList	Unsupported	
DirSelect	Unsupported	
Drag	Unsupported	
GetContextService	Unsupported	
PointerX	Unsupported	
PointerY	Unsupported	
Print	Unsupported	
SetRedraw	Unsupported	

ListView control

Properties | Events | Functions

APPEON

Important Requirements

The ListView control must be created in PowerBuilder painter.
The ListView control must be static. It cannot be dynamically created (for example, using the CREATE statement).

Properties for ListView control

Properties | Events | Functions

APPEON

Property	Support Level	Example Code
BackColor	Supported	lv_test.BackColor = ll_value lv_test.BackColor = RGB(192,192,192)
Border	Supported	lv_test.Border = lb_value
BorderStyle	Supported	lv_test.BorderStyle = StyleBox!
CheckBoxes	Supported	lv_test.CheckBoxes = lb_value
DeleteItems	Supported	lv_test.DeleteItems = lb_value
EditLabels	Supported	lv_test.EditLabels = lb_value
Enabled	Supported	lv_test.Enabled = lb_value
FaceName	Supported	lv_test.FaceName =ls_value
FullRowSelect	Supported	lv_test.FullRowSelect = TRUE
Height	Supported	lv_test.Height = li_value
Italic	Supported	lv_test.Italic = lb_value
Item[ ]	Supported	lv_test.item[intx] = ls_value
ItemPictureIndex[ ]	Supported	lv_test. ItemPictureIndex [li_x] = ls_value
LargePictureHeight	Supported	lv_test.LargePictureHeight = li_value Note: this value cannot be set to 0 or negative.
LargePictureName[ ]	Supported	lv_test. LargePictureName [li_x] = ls_value
LargePictureWidth	Supported	lv_test.LargePictureWidth = li_value Note: this value cannot be set to 0 or negative.
Pointer	Supported	lv_test.Pointer = ls_value The Pointer property is partially supported: ● Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow! ● Unsupported values of the Pointer property: AppStarting!, Help!, Icon!, No!, and SizeNS! ● User-defined pointers are not supported. ● All unsupported pointers are replaced with Arrow! in the Web application.
SmallPictureHeight	Supported	lv_test.SmallPictureHeight = li_value Note: this value cannot be set to 0 or negative.

SmallPictureName[ ]	Supported	lv_test. SmallPictureName [li_x] = ls_value
SmallPictureWidth	Supported	lv_test.SmallPictureWidth = li_value Note: this value cannot be set to 0 or negative.
StatePictureName[ ]	Supported	lv_test. StatePictureName [li_x] = ls_value
TabOrder	Supported	lv_test.TabOrder = 70
Tag	Supported	lv_test.Tag = ls_value
TextColor	Supported	lv_test.TextColor = ll_value lv_test.TextColor = RGB(192,192,192)
TextSize	Supported	lv_test.TextSize = li_value
Underline	Supported	lv_test.Underline = lb_value
View	Supported	lv_test.View = ListViewLargeIcon! Operations with arrow keys only have effect on the ListView control in the report style.
Visible	Supported	lv_test.Visible = lb_value
Weight	Supported	lv_test.Weight = li_value
Width	Supported	lv_test.Width = li_value
X	Supported	lv_test.X = li_value
Y	Supported	lv_test.Y =li_value
Accelerator	Unsupported	
AutoArrange	Unsupported	
BringToTop	Unsupported	
ButtonHeader	Unsupported	
ClassDefinition	Unsupported	
DragAuto	Unsupported	
DragIcon	Unsupported	
ExtendedSelect	Unsupported	
FixedLocations	Unsupported	
FontCharSet	Unsupported	
FontFamily	Unsupported	
FontPitch	Unsupported	
GridLines	Unsupported	
HeaderDragDrop	Unsupported	

HideSelection	Unsupported
ImeMode	Unsupported
LabelWrap	Unsupported
LargePictureMaskColor	Unsupported
OneClickActivate	Unsupported
Scrolling	Unsupported
ShowHeader	Unsupported
SmallPictureMaskColor	Unsupported
SortType	Unsupported
StatePictureHeight	Unsupported
StatePictureMaskColor	Unsupported
StatePictureWidth	Unsupported
TrackSelect	Unsupported
TwoClickActivate	Unsupported
UnderlineCold	Unsupported
UnderlineHot	Unsupported

Events for ListView control

Properties | Events | Functions

Event	Support Level	Example Code
Clicked	Supported	Clicked(index)
ColumnClick	Supported	ColumnClick
Constructor	Supported	Constructor
DeleteAllItems	Supported	DeleteAllItems
DeleteItem	Supported	DeleteItem(index)
Destructor	Supported	Destructor
DoubleClicked	Supported	DoubleClicked(index)
GetFocus	Supported	GetFocus
InsertItem	Supported	InsertItem(index)
ItemChanged	Supported	ItemChanged(index, focuschanged, hasfocus, selectionchange, selected, otherchange) Note: when the user selects an item and then click on the blank area, the CheckBox of the item will become unchecked. In addition, the ItemChanging and ItemChanged events will be triggered. However, in the Web application, the item will still remain checked, and only the ItemChanging event will be triggered.

ItemChanging	Supported	ItemChanging(index, focuschange, hasfocus, selectionchange, selected, otherchange) 1) If the return value of the event is 0, the ItemChanged Event will be triggered. Otherwise, stop processing. 2) When the user selects an item and then click on the blank area, the state of the item's checkbox will become unchecked. In addition, the ItemChanging and ItemChanged events will be triggered. However, in the Web application, the item will still remain checked, and only the ItemChanging event will be triggered.
LoseFocus	Supported	LoseFocus
BeginDrag	Unsupported	
BeginLabelEdit	Unsupported	
BeginRightDrag	Unsupported	
DragDrop	Unsupported	
DragEnter	Unsupported	
DragLeave	Unsupported	
DragWithin	Unsupported	
EndLabelEdit	Unsupported	
Help	Unsupported	
ItemActivate	Unsupported	
Key	Unsupported	
Other	Unsupported	
RightClicked	Unsupported	
RightDoubleClicked	Unsupported	
Sort	Unsupported	

Functions for ListView control

Properties | Events | Functions

Function	Support Level	Example Code
AddColumn	Supported	li_value = lv_test.AddColumn(ls_x,Center!,li_y)
AddItem	Supported	li_value= lv_test.AddItem(ls_x,li_y)
AddLargePicture	Supported	li_value = lv_test.AddLargePicture(ls_prop)
AddSmallPicture	Supported	li_value = lv_test.AddSmallPicture(ls_prop)
AddStatePicture	Supported	li_value = lv_test.AddstatePicture(ls_prop)

ClassName	Supported	ls_value = lv_test.ClassName( )
DeleteColumn	Supported	li_value = lv_test.DeleteColumn(1)
DeleteColumns	Supported	li_value = lv_test.DeleteColumns( )
DeleteItem	Supported	li_value = lv_test.DeleteItem(li_x)
DeleteItems	Supported	li_value = lv_list.DeleteItems()
DeleteLargePicture	Supported	li_value = lv_test.DeleteLargePicture(li_x)
DeleteLargePictures	Supported	li_value = lv_test.DeleteLargePictures( )
DeleteSmallPicture	Supported	li_value = lv_test.DeleteSmallPicture(li_x)
DeleteSmallPictures	Supported	li_value = lv_test.DeleteSmallPictures( )
DeleteStatePicture	Supported	li_value = lv_test.DeleteStatePicture(li_x)
DeleteStatePictures	Supported	li_value = lv_test.DeleteStatePictures( )
EditLabel	Supported	li_value= lv_test. EditLabel()
FindItem	Supported	li_Value = lv_test.FindItem (startindex, label, partial, wrap) The following syntax is unsupported: listviewname.FindItem(startindex, direction, focused, selected, cuthighlighted, drophighlighted).
GetColumn	Supported	li_strValue = lv_test.GetColumn (index, label, alignment, width)
GetItem	Supported	li_value = lv_test.GetItem(li_x,lv1)
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = lv_test.GetParent()
Hide	Supported	li_value = lv_test.Hide( )
InsertColumn	Supported	li_value = lv_test.InsertColumn(li_x,ls_x,Center!,li_y)
InsertItem	Supported	li_value = lv_test.InsertItem(2, "Insert String")
Move	Supported	li_value = lv_test.Move(li_x,li_y)
PostEvent	Supported	lb_value = lv_test.PostEvent(Clicked!)
Resize	Supported	li_value = lv_test.Resize(li_x,li_y)
SelectedIndex	Supported	li_value = lv_test.SelectedIndex( )

SetColumn	Supported	li_value = lv_test.SetColumn(li_x,ls_x,Center!,li_y)
SetFocus	Supported	li_value = lv_test.SetFocus( ) Unsupported: SetFocus(lv_test)
SetItem	Supported	li_value = lv_test.SetItem(li_x[],li_y[],lv1)
SetPosition	Supported	li_value = lv_test.SetPosition(Behind!, lv_three)
Show	Supported	li_value = lv_test.Show( )
Sort	Supported	li_value = lv_test.Sort(Ascending!) li_value = lv_test.Sort(Descending!) li_value = lv_test.Sort(Unsorted!) Note: three types, ascending!, descending! and unsorted!, are supported. The sort type userdefinesort! is not supported.
TotalColumns	Supported	li_cols = lv_list.TotalColumns( )
TotalItems	Supported	li_value = lv_test.TotalItems( )
TriggerEvent	Supported	li_value = lv_test.TriggerEvent("Clicked")

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

Arrange	Unsupported	
Drag	Unsupported	
GetContextService	Unsupported	
GetOrigin	Unsupported	
PointerX	Unsupported	
PointerY	Unsupported	
Print	Unsupported	
SetOverlayPicture	Unsupported	
SetRedraw	Unsupported	
TotalSelected	Unsupported	

MultiLineEdit control

Properties | Events | Functions

Important Requirements

- The MultiLineEdit control must be created in PowerBuilder painter.

- The MultiLineEdit control must be static. It cannot be dynamically created (for example, using the CREATE statement).
- In order to get the text display of the MultiLineEdit control identical on the Web as in PowerBuilder, use "~r~n" to control the start and end of each line.
- If the user right-clicks on the scrollbar for the MultiLineEdit control, the RButtonDown event will be triggered in the Web application, but not in PowerBuilder.

Properties for MultiLineEdit control

Properties | Events | Functions

Property	Support Level	Example Code
Alignment	Supported	mle_test.Alignment = Center!
BackColor	Supported	mle_test.BackColor = ll_value mle_test.BackColor = RGB(192,192,192)
Border	Supported	mle_test.Border = lb_value
BorderStyle	Supported	mle_test.BorderStyle = StyleBox!
DisplayOnly	Supported	mle_test.DisplayOnly =lb_value
Enabled	Supported	mle_test.Enabled = lb_value 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_test.FaceName = ls_value
Height	Supported	mle_test.Height = li_value
HScrollBar	Supported	mle_test.HScrollBar = TRUE
Italic	Supported	mle_test.Italic = lb_value
Limit	Supported	mle_test.Limit = li_value
Pointer	Supported	mle_pointer= 'Beam!' //Specifies the name of the stock pointer or the file containing the pointer used for the control. The Pointer property is partially supported: • Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow! • Unsupported values of the Pointer property: AppStarting!, Help!, Icon!, No!, and SizeNS! • User-defined pointers are not supported.

- All unsupported pointers are replaced with Arrow! in the Web application.

TabOrder	Supported	mle_test.TabOrder = 80
Tag	Supported	mle_test.Tag = ls_value
Text	Supported	mle_test.Text = ls_value
TextCase	Supported	mle_test.TextCase = AnyCase! mle_test.TextCase = Lower! mle_test.TextCase = Upper!
TextColor	Supported	mle_test.TextColor = ll_value
TextSize	Supported	mle_test.TextSize = li_value
Underline	Supported	mle_test.Underline = lb_value
Visible	Supported	mle_test.Visible = lb_value
VScrollBar	Supported	mle_test.VScrollBar = TRUE
Weight	Supported	mle_test.Weight = li_value
Width	Supported	mle_test.Width = li_value
X	Supported	mle_test.X = li_value
Y	Supported	mle_test.Y = li_value
Accelerator	Unsupported	
AutoHScroll	Unsupported	
AutoVScroll	Unsupported	
BringToTop	Unsupported	
ClassDefinition	Unsupported	
DragAuto	Unsupported	
DragIcon	Unsupported	
FontCharSet	Unsupported	
FontFamily	Unsupported	
FontPitch	Unsupported	
HideSelection	Unsupported	
IgnoreDefaultButton	Unsupported	
ImeMode	Unsupported	
RightToLeft	Unsupported	
TabStop[ ]	Unsupported	

Events for MultiLineEdit control

Properties | Events | Functions

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor
GetFocus	Supported	GetFocus
LoseFocus	Supported	LoseFocus
Modified	Supported	Modified Note: unlike in PowerBuilder, in JavaScript the event cannot be triggered for the Web control if (1) the text in the control has not been changed; or (2) the user tried to trigger the event by pressing the Enter key. A Modified event is followed by a LoseFocus event.
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
DragDrop	Unsupported	
DragEnter	Unsupported	
DragLeave	Unsupported	
DragWithin	Unsupported	
Help	Unsupported	
Other	Unsupported	

Functions for MultiLineEdit control

Properties | Events | Functions

Function	Support Level	Example Code
ClassName	Supported	ls_value = mle_test.ClassName()
Clear	Supported	ll_value = mle_test.Clear() Note: the function executes successfully only when the control has focus.
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = mle_test.GetParent()
Hide	Supported	li_value = mle_test.Hide()
Move	Supported	li_value = mle_test.Move(li_x,li_y)
PostEvent	Supported	lb_value = mle_test.PostEvent(Constructor!)
ReplaceText	Supported	li_strexevalue = mle_test.ReplaceText(ls_x) Note: the function executes successfully only when the

		control has focus.
Resize	Supported	li_value = mle_test.Resize(li_x,li_y)
SelectedLength	Supported	li_value = mle_test.SelectedLength() Note: if the control is off screen, the execution of the function may have an incorrect return result.
SelectedStart	Supported	li_value = mle_test.SelectedStart() Note: if the control is off screen, the execution of the function may have an incorrect return result.
SelectedText	Supported	ls_value = mle_test.SelectedText() Note: if the control is off screen, the execution of the function may have an incorrect return result.
SelectText	Supported	li_value = mle_test.SelectText(li_x,li_y) Note: if the control is off screen, the execution of the function may have an incorrect return result.
SetFocus	Supported	li_value = mle_test.SetFocus() Unsupported: SetFocus(mle_test)
SetPosition	Supported	mle_two.SetPosition(Behind!, mle_three) li_value = mle_test.SetPosition(ToTop!) li_value = mle_test.SetPosition(ToBottom!)
Show	Supported	li_value = mle_test.Show()
TriggerEvent	Supported	li_value = mle_test.TriggerEvent(Constructor!)

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

CanUndo	Unsupported	
Copy	Unsupported	
Cut	Unsupported	
Drag	Unsupported	
GetContextService	Unsupported	
LineCount	Unsupported	
LineLength	Unsupported	
Paste	Unsupported	
PointerX	Unsupported	
PointerY	Unsupported	
Position	Unsupported	
Print	Unsupported	
Scroll	Unsupported	
SelectedLine	Unsupported	

SetRedraw	Unsupported
TextLine	Unsupported
Undo	Unsupported

Picture control

Properties | Events | Functions

Important Requirements

The Picture control must be created in PowerBuilder painter.
The Picture control must be static. It cannot be dynamically created (for example, using the CREATE statement).
Picture 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

Pictures in an application

- The name of the picture file cannot be a string that is ended 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".
- Appeon identifies picture names case-insensitively. The picture name cannot contain any non ASCII character.
- Specify which folder contains the pictures in the Appeon Developer Configuration | Application Profile Properties. All the pictures used in the application should be placed in the folder. In this way, pictures can still display if the names of the picture files are from Database or are dynamically generated.

Properties for Picture control

Properties | Events | Functions

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

Enabled	Supported	p_test.Enabled = lb_value
Height	Supported	p_test.Height = li_value
OriginalSize	Supported	p_test.OriginalSize = lb_value
PictureName	Supported	p_test.PictureName = ls_value Note: it is unsupported that the PictureName property contains any double-byte character.
Pointer	Supported	p_pointer= 'size!' //Specifies the name of the stock pointer or the file containing the pointer used for the control. The Pointer property is partially supported: • Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow! • Unsupported values of the Pointer property: AppStarting!, Help!, Icon!, No!, and SizeNS! • User-defined pointers are not supported. • All unsupported pointers are replaced with Arrow! in the Web application.
TabOrder	Supported	p_test.TabOrder = 100
Tag	Supported	p_test.Tag = ls_value
Visible	Supported	p_test.Visible = lb_value
Width	Supported	p_test.Width = li_value
X	Supported	p_test.X = li_value
Y	Supported	p_test.Y = li_value
BringToTop	Unsupported	
ClassDefinition	Unsupported	
DragAuto	Unsupported	
DragIcon	Unsupported	
FocusRectangle	Unsupported	
Invert	Unsupported	
Map3DColors	Unsupported	
PowerTipText	Unsupported	

Events for Picture control

Properties | Events | Functions

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

Functions for Picture control

Properties | Events | Functions

Function	Support Level	Example Code
ClassName	Supported	ls_value = p_test.ClassName( )
Draw	Supported	li_value = p_test.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_test.GetParent( )
Hide	Supported	li_value = p_test.Hide( )
Move	Supported	li_value = p_test.Move(li_x,li_y)
PostEvent	Supported	lb_value = p_test.PostEvent(Clicked!)
Resize	Supported	li_value = p_test.Resize(li_x,li_y) Note: in PowerBuilder, to resize a picture will only change the size of the picture control, not the picture itself. After conversion, the size of the picture itself will be changed.

SetFocus	Supported	li_value = p_test.SetFocus( ) Unsupported: SetFocus(p_test)
SetPosition	Supported	li_value = pic_test.SetPosition( )
Show	Supported	li_value = p_test.Show( )
TriggerEvent	Supported	li_value = p_test.TriggerEvent(Clicked!)

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

Drag	Unsupported	
GetContextService	Unsupported	
PointerX	Unsupported	
PointerY	Unsupported	
Print	Unsupported	
SetPicture	Unsupported	
SetRedraw	Unsupported	

PictureButton control

Properties | Events | Functions

APPEON

Important Requirements

The PictureButton control must be created in PowerBuiler painter.
The PictureButton control must be static. It cannot be dynamically created (for example, using the CREATE statement).

Properties for PictureButton control

Properties | Events | Functions

APPEON

Property	Support Level	Example Code
DisabledName	Supported	pb_1.DisabledName = "d:\pbhelp\controls.bmp" //Specifies the name of the picture (bitmap image) that displays when the control is disabled. If the string has no extension, then PowerBuilder adds an appropriate extension.
Enabled	Supported	pb_test.Enabled = TRUE
FaceName	Supported	pb_test.FaceName = ls_value
Height	Supported	pb_test.Height = li_value

HTextAlign	Supported	pb_test.HtextAlign = "Left!" //Specifies how the text in the control is aligned. Values are: Center! Justify! Left! Right!
Italic	Supported	pb_test.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_test.PictureName = ls_value
Pointer	Supported	pb_1.pointer= 'Arrow!' //Specifies the name of the stock pointer or the file containing the pointer used for the control. The Pointer property is partially supported: • Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow! • Unsupported values of the Pointer property: AppStarting!, Help!, Icon!, No!, and SizeNS! • User-defined pointers are not supported. • All unsupported pointers are replaced with Arrow! in the Web application.
TabOrder	Supported	pb_test.TabOrder = 10
Tag	Supported	pb_test.Tag = ls_value
Text	Supported	pb_test.Text = ls_value
TextSize	Supported	pb_test.TextSize = li_value
Underline	Supported	pb_test.Underline = lb_value
Visible	Supported	pb_test.Visible = lb_value
VTextAlign	Supported	pb_test.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_test.Weight = li_value
Width	Supported	pb_test.Width = li_value
X	Supported	pb_test.X = li_value

Y	Supported	pb_test.Y = li_value
BringToTop	Unsupported	
Cancel	Unsupported	
ClassDefinition	Unsupported	
Default	Unsupported	
DragAuto	Unsupported	
DragIcon	Unsupported	
FontCharSet	Unsupported	
FontFamily	Unsupported	
FontPitch	Unsupported	
Map3DColors	Unsupported	
PowerTipText	Unsupported	

Events for PictureButton control

Properties | Events | Functions

APPEON

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

Functions for PictureButton control

Properties | Events | Functions

APPEON

Function	Support Level	Example Code
ClassName	Supported	ls_returnvalue = pb_test.ClassName()
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = pb_test.GetParent()

Hide	Supported	li_returnvalue = pb_test.Hide()
Move	Supported	li_returnvalue = pb_test.Move(li_xpos, li_ypos)
PostEvent	Supported	lb_returnvalue = pb_test.PostEvent(event)
Resize	Supported	li_returnvalue = pb_test.Resize(li_width, li_height)
SetFocus	Supported	li_returnvalue = pb_test.SetFocus() Unsupported: SetFocus(pb_test)
SetPosition	Supported	li_returnvalue = pb_test.SetPosition(position)
Show	Supported	li_returnvalue = pb_test.Show()
TriggerEvent	Supported	li_returnvalue = pb_test.TriggerEvent(event)

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

Drag	Unsupported	
GetContextService	Unsupported	
PointerX	Unsupported	
PointerY	Unsupported	
Print	Unsupported	
SetRedraw	Unsupported	

PictureHyperLink control

Properties | Events | Functions

Important Requirements

The PictureHyperLink control must be created in a PowerBuilder painter.

The PictureHyperLink control must be static. It cannot be dynamically created (for example, using the CREATE statement).

The PictureHyperLink controls can contain images in the following formats:

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

- Cursor files with .CUR extensions

Properties for PictureHyperLink control

Properties | Events | Functions

Property	Support Level	Example Code
Border	Supported	phl_test.Border = lb_value
BorderStyle	Supported	phl_test.BorderStyle = StyleBox!
Enabled	Supported	phl_test.Enabled = lb_value
Height	Supported	phl_test.Height = li_value
OriginalSize	Supported	phl_test.OriginalSize = lb_value
PictureName	Supported	phl_test.PictureName = ls_value Note: it is unsupported that the PictureName property contains any double-byte character.
Pointer	Supported	phl_pointer= 'size!' //Specifies the name of the stock pointer or the file containing the pointer used for the control. The Pointer property is partially supported: • Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow! • Unsupported values of the Pointer property: AppStarting!, Help!, Icon!, No!, and SizeNS! • User-defined pointers are not supported. • All unsupported pointers are replaced with Arrow! in the Web application.
TabOrder	Supported	phl_test.TabOrder = 100
Tag	Supported	phl_test.Tag = ls_value
URL	Supported	phl_1.URL = "http://www.appeon.net"
Visible	Supported	phl_test.Visible = lb_value
Width	Supported	phl_test.Width = li_value

X	Supported	phl_test.X = li_value
Y	Supported	phl_test.Y = li_value
BringToTop	Unsupported	
ClassDefinition	Unsupported	
DragAuto	Unsupported	
DragIcon	Unsupported	
FocusRectangle	Unsupported	
Invert	Unsupported	
Map3DColors	Unsupported	
PowerTipText	Unsupported	

Events for PictureHyperLink control

Properties | Events | Functions

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

Functions for PictureHyperLink control

Properties | Events | Functions

Function	Support Level	Example Code
ClassName	Supported	ls_value = phl_test.ClassName( )
Draw	Supported	li_value = phl_test.Draw(li_x,li_y)
GetParent	Supported	PowerObject lpo_value lpo_value = phl_test.GetParent()

Hide	Supported	li_value = phl_test.Hide( )
Move	Supported	li_value = phl_test.Move(li_x,li_y)
PostEvent	Supported	lb_value = phl_test.PostEvent(Clicked!)
Resize	Supported	li_value = phl_test.Resize(li_x,li_y)
SetFocus	Supported	li_value = phl_test.SetFocus( ) Unsupported: SetFocus(phl_test)
SetPosition	Supported	li_value = pic_test.SetPosition( )
Show	Supported	li_value = phl_test.Show( )
TriggerEvent	Supported	li_value = phl_test.TriggerEvent(Clicked!)

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

Drag	Unsupported	
GetContextService	Unsupported	
PointerX	Unsupported	
PointerY	Unsupported	
Print	Unsupported	
SetPicture	Unsupported	
SetRedraw	Unsupported	

PictureListBox control

Properties | Events | Functions

APPEON

Important Requirements

The PictureListBox control must be created in PowerBuilder painter.
The PictureListBox control must be static. It cannot be dynamically created (for example, using the CREATE statement).
Setting a PictureListBox as a Dropdown PictureListBox is not supported.

Properties for PictureListBox control

Properties | Events | Functions

APPEON

Property	Support Level	Example Code
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!'
DisableNoScroll	Supported	plb_1. DisableNoScroll = lb_value
Enabled	Supported	plb_1.Enabled = lb_value
ExtendedSelect	Supported	plb_1. ExtendedSelect = lb_value The user cannot select multiple items by clicking an item and dragging the mouse up or down to select items, even when the ExtendedSelect property is enabled.
FaceName	Supported	plb_1.FaceName = ls_value
Height	Supported	plb_1.Height = li_value
HScrollBar	Supported	plb_1.Hscrollbar = lb_value
Italic	Supported	plb_1.Italic = lb_value
Item[ ]	Supported	plb_actions.Item[] = ls_value
ItemPictureIndex [ ]	Supported	plb_actions.ItemPictureIndex[ ] = li_value[]
MultiSelect	Supported	plb_actions.MultiSelect = lb_value
PictureName[ ]	Supported	ls_result[] = plb_1.PictureName[li_x[]] Note: it is unsupported that the PictureName property contains any double-byte character.
Pointer	Supported	plb_1.pointer = ‘Beam!’ The Pointer property is partially supported: ● Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow! ● Unsupported values of the Pointer property: AppStarting!, Help!, Icon!, No!, and SizeNS! ● User-defined pointers are not supported. ● All unsupported pointers are replaced with Arrow! in the Web application.
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
Accelerator	Unsupported	
AllowEdit	Unsupported	
BringToTop	Unsupported	
ClassDefinition	Unsupported	
DragAuto	Unsupported	
DragIcon	Unsupported	
FontCharSet	Unsupported	
FontFamily	Unsupported	
FontPitch	Unsupported	
ImeMode	Unsupported	
PictureHeight	Unsupported	
PictureMaskColor	Unsupported	
PictureWidth	Unsupported	
RightToLeft	Unsupported	
ShowList	Unsupported	
TabStop[ ]	Unsupported	

Events for PictureListBox control

Properties | Events | Functions

Event	Support Level	Example Code
Constructor	Supported	Constructor

Destructor	Supported	Destructor
DoubleClicked	Supported	DoubleClicked
GetFocus	Supported	GetFocus
LoseFocus	Supported	LoseFocus
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
SelectionChanged	Supported	SelectionChanged(index)
DragDrop	Unsupported	
DragEnter	Unsupported	
DragLeave	Unsupported	
DragWithin	Unsupported	
Help	Unsupported	
Other	Unsupported	

Functions for PictureListBox control

Properties | Events | Functions

Function	Support Level	Example Code
AddItem	Supported	li_position = plb_birds.AddItem(ls_name, li_pic)
AddPicture	Supported	li_pic = plb_birds.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 ()
FindItem	Supported	li_index = plb_actions.FindItem ("Delete", 1)
GetParent	Supported	PowerObject lpo_value lpo_value = plb1.GetParent()
Hide	Supported	li_value = plb_1.Hide()
InsertItem	Supported	plb_actions.InsertItem("Run Application", 5) plb_actions.InsertItem("Run Application", 5, 4)
Move	Supported	li_value = plb_1.Move(li_x[],li_y[])
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_actions.SelectedIndex()
SelectedItem	Supported	li_item = plb_Shortcuts.SelectedItem()
SelectItem	Supported	li_value = plb_1.SelectItem (li_x[],li_y[])
SetFocus	Supported	li_value = plb_1.SetFocus() Unsupported: SetFocus(plb_test)
SetPosition	Supported	li_returnvalue = plb_1.SetPosition(position)
SetState	Supported	plb_Actions.SetState(6, TRUE)
SetTop	Supported	plb_Actions.SetTop(6)
Show	Supported	li_value = plb_1.Show()
State	Supported	li_item = plb_Contact.State(3)
Text	Supported	ls_return = plb_actions.Text(2)
Top	Supported	li_num = plb_Contacts.Top()
TotalItems	Supported	li_value= plb_1.TotalItems()
TotalSelected	Supported	integer SelectedTotal SelectedTotal = ls_Actions.TotalSelected()
TriggerEvent	Supported	li_value = plb_1.TriggerEvent(Constructor!)

TypeOf	Supported	if ddplb_1.typeof()=picturelistbox! Then
		  ls_value='picturelistbox!'
		else
		  ls_value='isvalid'
		end if

Clear	Unsupported	
Copy	Unsupported	
Cut	Unsupported	
DirList	Unsupported	
DirSelect	Unsupported	
Drag	Unsupported	
GetContextService	Unsupported	
Paste	Unsupported	
PointerX	Unsupported	

PointerY	Unsupported
Position	Unsupported
Print	Unsupported
SetRedraw	Unsupported

RadioButton control

Properties | Events | Functions

APPEON

Important Requirements

The RadioButton control must be created in PowerBuilder painter.
The RadioButton control must be static. It cannot be dynamically created (for example, using the CREATE statement).

Properties for RadioButton control

Properties | Events | Functions

APPEON

Property	Support Level	Example Code
BackColor	Supported	rb_test.BackColor = ll_value rb_test.BackColor = RGB(192,192,192)
BorderStyle	Supported	rb_test.BorderStyle = StyleLowered! rb_test.BorderStyle = StyleRaised!
Checked	Supported	rb_test.Checked = lb_value
Enabled	Supported	rb_test.Enabled = lb_value
FaceName	Supported	rb_test.FaceName = ls_value
Height	Supported	rb_test.Height = li_value
Italic	Supported	rb_test.Italic = lb_value
Pointer	Supported	rb_test.Pointer = ls_value The Pointer property is partially supported: • Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow! • Unsupported values of the Pointer property: AppStarting!, Help!, Icon!, No!, and SizeNS! • User-defined pointers are not supported. • All unsupported pointers are replaced with Arrow! in the

		Web application.
TabOrder	Supported	rb_test.TabOrder = 120
Tag	Supported	rb_test.Tag = ls_value
Text	Supported	rb_test.Text = ls_value
TextColor	Supported	rb_test.TextColor = ll_value
TextSize	Supported	rb_test.TextSize = li_value
Underline	Supported	rb_test.Underline = lb_value
Visible	Supported	rb_test.Visible = lb_value
Weight	Supported	rb_test.Weight = li_value
Width	Supported	rb_test.Width = li_value
X	Supported	rb_test.X = li_value
Y	Supported	rb_test.Y = li_value
Automatic	Unsupported	
BringToTop	Unsupported	
ClassDefinition	Unsupported	
DragAuto	Unsupported	
DragIcon	Unsupported	
FontCharSet	Unsupported	
FontFamily	Unsupported	
FontPitch	Unsupported	
LeftText	Unsupported	
RightToLeft	Unsupported	

Events for RadioButton control

Properties | Events | Functions

Event	Support Level	Example Code
Clicked	Supported	Clicked
Constructor	Supported	Constructor
Destructor	Supported	Destructor
GetFocus	Supported	GetFocus
LoseFocus	Supported	LoseFocus

RButtonDown	Supported	RButtonDown(flags, xpos, ypos)
DragDrop	Unsupported	
DragEnter	Unsupported	
DragLeave	Unsupported	
DragWithin	Unsupported	
Help	Unsupported	
Other	Unsupported	

Functions for RadioButton control

Properties | Events | Functions

Function	Support Level	Example Code
ClassName	Supported	ls_value = rb_test.ClassName( )
GetParent	Supported	PowerObject lpo_value lpo_value = rb_test.GetParent( )
Hide	Supported	li_value = rb_test.Hide( )
Move	Supported	li_value = rb_test.Move(li_x,li_y)
PostEvent	Supported	lb_value = rb_test.PostEvent(Clicked!)
Resize	Supported	li_value = rb_test.Resize(li_x,li_y)
SetFocus	Supported	li_value = rb_test.SetFocus( ) Unsupported: SetFocus(rb_test)
SetPosition	Supported	li_value = rb_test.SetPosition(ToTop!) li_value = rb_test.SetPosition(ToBottom!)
Show	Supported	li_value = rb_test.Show( )
TriggerEvent	Supported	li_value = rb_test.TriggerEvent(Clicked!)

TypeOf	Supported	if this.TypeOf( ) = RadioButton! Then
		  ls_value = "RadioButton!"
		end if

Drag	Unsupported	
GetContextService	Unsupported	
PointerX	Unsupported	
PointerY	Unsupported	
Print	Unsupported	
SetRedraw	Unsupported	

Rectangle control

Properties | Events | Functions

Important Requirements

The Rectangle control must be created in PowerBuilder painter.
The Rectangle control must be static. It cannot be dynamically created (for example, using the CREATE statement).

Properties for Rectangle control

Properties | Events | Functions

Property	Support Level	Example Code
FillColor	Supported	r_test.FillColor = ll_value It is unsupported to set the value of the FillColor property to Desktop.
Height	Supported	r_test.Height = li_value
LineColor	Supported	r_test.LineColor = ll_value It is unsupported to set the value of the LineColor property to Desktop.
LineStyle	Supported	r_test.LineStyle = Continuous! r_test.LineStyle = Dash! r_test.LineStyle = DashDot! r_test.LineStyle = DashDotDot! r_test.LineStyle = Dot! r_test.LineStyle = Transparent! Note: the LineStyle Dash!, Dashdot!, Dashdotdot! and Dot! all set the appearance of the border to dotted line.
LineThickness	Supported	r_test.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_test.Tag = ls_value
Visible	Supported	r_test.Visible = lb_value
Width	Supported	r_test.Width = li_value
X	Supported	r_test.X = li_value
Y	Supported	r_test.Y = li_value
ClassDefinition	Unsupported	
FillPattern	Unsupported	

Events for Rectangle control

Properties | Events | Functions

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor

Functions for Rectangle control

Properties | Events | Functions

Function	Support Level	Example Code
ClassName	Supported	ls_value = r_test.ClassName( )
GetParent	Supported	PowerObject lpo_value lpo_value = r_test.GetParent( )
Hide	Supported	li_value = r_test.Hide( )
Move	Supported	li_value = rb_test.Move(li_x,li_y)
PostEvent	Supported	lb_value = rb_test.PostEvent(Clicked!)
Resize	Supported	lb_value = rb_test.Resize(Clicked!)
Show	Supported	li_value = rb_test.Show( )
TriggerEvent	Supported	li_value = rb_test.TriggerEvent(Clicked!)

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

GetContextService	Unsupported	

SingleLineEdit control

Properties | Events | Functions

Important Requirements

The SingleLineEdit control must be created in PowerBuilder painter.
The SingleLineEdit control must be static. It cannot be dynamically created (for example, using the CREATE statement).

Properties for SingleLineEdit control

Properties | Events | Functions

Property	Support Level	Example Code
BackColor	Supported	sle_test.BackColor = ll_value
Border	Supported	sle_test.Border = lb_value
BorderStyle	Supported	sle_test.BorderStyle = StyleRaised!
DisplayOnly	Supported	sle_test.DisplayOnly = lb_value Note: if a SingleLineEdit control is set to DisplayOnly, the control cannot get focus.
Enabled	Supported	sle_test.Enabled = lb_value
FaceName	Supported	sle_test.FaceName = ls_value
Height	Supported	sle_test.Height = li_value
Italic	Supported	sle_test.Italic = lb_value
Limit	Supported	sle_test.Limit = li_value
Password	Supported	sle_test.Password = lb_value
Pointer	Supported	sle_test.pointer= 'Arrow!' //Specifies the name of the stock pointer or the file containing the pointer used for the control The Pointer property is partially supported: • Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow! • Unsupported values of the Pointer property: AppStarting!, Help!, Icon!, No!, and SizeNS! • User-defined pointers are not supported. • All unsupported pointers are replaced with Arrow! in the Web application.
TabOrder	Supported	sle_test.TabOrder = 130
Tag	Supported	sle_test.Tag = ls_value
Text	Supported	sle_test.Text = ls_value
TextCase	Supported	sle_test.TextCase = AnyCase! sle_test.TextCase = Lower! sle_test.TextCase = Upper!
TextColor	Supported	sle_test.TextColor = ll_value

TextSize	Supported	sle_test.TextSize = li_value
Underline	Supported	sle_test.Underline = lb_value
Visible	Supported	sle_test.Visible = lb_value
Weight	Supported	sle_test.Weight = li_value
Width	Supported	sle_test.Width = li_value
X	Supported	sle_test.X = li_value
Y	Supported	sle_test.Y = li_value
Accelerator	Unsupported	
AutoHScroll	Unsupported	
BringToTop	Unsupported	
ClassDefinition	Unsupported	
DragAuto	Unsupported	
DragIcon	Unsupported	
FontCharSet	Unsupported	
FontFamily	Unsupported	
FontPitch	Unsupported	
HideSelection	Unsupported	
ImeMode	Unsupported	
RightToLeft	Unsupported	

Events for SingleLineEdit control

Properties | Events | Functions

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor
GetFocus	Supported	GetFocus
LoseFocus	Supported	LoseFocus
Modified	Supported	Modified Note: unlike in PowerScript, in JavaScript the event cannot be triggered for the Web control if (1) the text in the control has not been changed; or (2) the user tried to trigger the event by pressing the Enter key.
RButtonDown	Supported	RButtonDown(flags, xpos, ypos)

DragDrop	Unsupported
DragEnter	Unsupported
DragLeave	Unsupported
DragWithin	Unsupported
Help	Unsupported
Other	Unsupported

Functions for SingleLineEdit control

Properties | Events | Functions

Function	Support Level	Example Code
ClassName	Supported	ls_returnvalue = sle_test.ClassName()
Clear	Supported	li_returnvalue = sle_test.Clear()
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = sle_test.GetParent()
Hide	Supported	li_returnvalue = sle_test.Hide()
Move	Supported	li_returnvalue = sle_test.Move(li_x,li_y)
PostEvent	Supported	lb_returnvalue = sle_test.PostEvent(event)
ReplaceText	Supported	li_returnvalue = sle_test.ReplaceText(text)
Resize	Supported	li_returnvalue = sle_test.Resize(li_x,li_y)
SelectedLength	Supported	li_returnvalue = sle_test.SelectedLength() If the control is off screen, the execution of the function may have an incorrect return result.
SelectedStart	Supported	li_returnvalue = sle_test.SelectedStart() If the control is off screen, the execution of the function may have an incorrect return result.
SelectedText	Supported	ls_returnvalue = sle_test.SelectedText() If the control is off screen, the execution of the function may have an incorrect return result.
SelectText	Supported	li_returnvalue = sle_test.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_test.SetFocus() Unsupported: SetFocus(sle_test)

SetPosition	Supported	li_returnvalue = sle_test.SetPosition(position)
Show	Supported	li_returnvalue = sle_test.Show()
TriggerEvent	Supported	li_returnvalue = sle_test.TriggerEvent(event)

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

CanUndo	Unsupported	
Copy	Unsupported	
Cut	Unsupported	
Drag	Unsupported	
GetContextService	Unsupported	
Paste	Unsupported	
PointerX	Unsupported	
PointerY	Unsupported	
Position	Unsupported	
Print	Unsupported	
SetRedraw	Unsupported	
Undo	Unsupported	

StaticHyperLink control

Properties | Events | Functions

Important Requirements

The StaticHyperLink control must be created in PowerBuilder painter.
The StaticHyperLink control must be static. It cannot be dynamically created (for example, using the CREATE statement).

Properties for StaticHyperLink control

Properties | Events | Functions

Property	Support Level	Example Code
Alignment	Supported	shl_test.Alignment = ls_value
BackColor	Supported	shl_test.BackColor = ll_value shl_test.BackColor = RGB(192,192,192)
Border	Supported	shl_test.Border = lb_value
BorderColor	Supported	shl_test.BorderColor = ll_value

BorderStyle	Supported	shl_test.BorderStyle = StyleBox! shl_test.BorderStyle = StyleLowered! shl_test.BorderStyle = StyleRaised! Note: StyleShadowBox! is not supported.
Enabled	Supported	shl_test.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. However, if the Enabled property is false, the text of the StaticHyperLink control will be gray in the Web application.
FaceName	Supported	shl_test.FaceName = ls_value
FocusRectangle	Supported	shl_test.FocusRectangle = TRUE Note: when the FocusRectangle property is enabled, the dotted rectangle (the focus rectangle) will not be activated unless the user clicks the Tab key.
Height	Supported	shl_test.Height = li_value
Italic	Supported	shl_test.Italic = lb_value
Pointer	Supported	shl_test.Pointer = ls_value The Pointer property is partially supported: • Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow! • Unsupported values of the Pointer property: AppStarting!, Help!, Icon!, No!, and SizeNS! • User-defined pointers are not supported. • All unsupported pointers are replaced with Arrow! in the Web application.
TabOrder	Supported	shl_test.TabOrder = 140
Tag	Supported	shl_test.Tag = ls_value
Text	Supported	shl_test.Text = ls_value
TextColor	Supported	shl_test.TextColor = ll_value
TextSize	Supported	shl_test.TextSize = li_value
Underline	Supported	shl_test.Underline = lb_value
URL	Supported	shl_1.URL = "http://www.appeon.com"

Visible	Supported	shl_test.Visible = lb_value
Weight	Supported	shl_test.Weight = li_value
Width	Supported	shl_test.Width = li_value
X	Supported	shl_test.X = li_value
Y	Supported	shl_test.Y = li_value
BringToTop	Unsupported	
ClassDefinition	Unsupported	
DisabledLook	Unsupported	
DragAuto	Unsupported	
DragIcon	Unsupported	
FillPattern	Unsupported	
FontCharSet	Unsupported	
FontFamily	Unsupported	
FontPitch	Unsupported	
RightToLeft	Unsupported	

Events for StaticHyperLink control

Properties | Events | Functions

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

Functions for StaticHyperLink control

Properties | Events | Functions

Function	Support Level	Example Code
ClassName	Supported	ls_returnvalue = shl_test.ClassName()
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = shl_test.GetParent()
Hide	Supported	li_returnvalue = shl_test.Hide()
Move	Supported	li_returnvalue = shl_test.Move(li_x,li_y)
PostEvent	Supported	lb_returnvalue = shl_test.PostEvent(event)
Resize	Supported	li_returnvalue = shl_test.Resize(li_x,li_y)
SetFocus	Supported	li_returnvalue = shl_test.SetFocus() Unsupported: SetFocus(shl_test)
SetPosition	Supported	li_returnvalue = shl_test.SetPosition(position)
SetRedraw	Supported	li_returnvalue = shl_test.SetRedraw(lb_x)
Show	Supported	li_returnvalue = shl_test.Show()
TriggerEvent	Supported	li_returnvalue = shl_test.TriggerEvent(event)

TypeOf	Supported	if shl_test.typeof() = StaticHyperLink! Then
		ls_returnvalue = 'StaticHyperLink!'
		end if

Drag	Unsupported	
GetContextService	Unsupported	
PointerX	Unsupported	
PointerY	Unsupported	
Print	Unsupported	

StaticText control

Properties | Events | Functions

Important Requirements

The StaticText control must be created in PowerBuilder painter.
The StaticText control must be static. It cannot be dynamically created (for example, using the CREATE statement).

Properties for StaticText control

Properties | Events | Functions

Property	Support Level	Example Code
Alignment	Supported	st_test.Alignment = ls_value
BackColor	Supported	st_test.BackColor = ll_value st_test.BackColor = RGB(192,192,192)
Border	Supported	st_test.Border = lb_value
BorderColor	Supported	st_test.BorderColor = ll_value
BorderStyle	Supported	st_test.BorderStyle = StyleBox! st_test.BorderStyle = StyleLowered! st_test.BorderStyle = StyleRaised! Note: the border style cannot be StyleShadowBox!.
Enabled	Supported	st_test.Enabled = lb_value Notes: 1) If the Enabled property is false and the user clicks the control, the Clicked event will be triggered. In PowerBuilder, the Click event will not be triggered. 2) In PowerBuilder, setting the Enabled property of StaticText to TRUE or FALSE has no effect. The text always displays as it is. However, if the Enabled property is false, the text of the StaticText control will be gray in the Web application.
FaceName	Supported	st_test.FaceName = ls_value
FocusRectangle	Supported	st_test.FocusRectangle = TRUE Note: when this property is set to TRUE, the focus rectangle will not display on the Web unless the user clicks the Tab key to activate the rectangle first.
Height	Supported	st_test.Height = li_value
Italic	Supported	st_test.Italic = lb_value
Pointer	Supported	st_test.Pointer = ls_value The Pointer property is partially supported: • Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow! • Unsupported values of the Pointer property: AppStarting!, Help!, Icon!, No!, and SizeNS! • User-defined pointers are not supported. • All unsupported pointers are replaced with Arrow! in the Web application.
TabOrder	Supported	st_test.TabOrder = 140

Tag	Supported	st_test.Tag = ls_value
Text	Supported	st_test.Text = ls_value
TextColor	Supported	st_test.TextColor = ll_value
TextSize	Supported	st_test.TextSize = li_value
Underline	Supported	st_test.Underline = lb_value
Visible	Supported	st_test.Visible = lb_value
Weight	Supported	st_test.Weight = li_value
Width	Supported	st_test.Width = li_value
X	Supported	st_test.X = li_value
Y	Supported	st_test.Y = li_value
BringToTop	Unsupported	
ClassDefinition	Unsupported	
DisabledLook	Unsupported	
DragAuto	Unsupported	
DragIcon	Unsupported	
FillPattern	Unsupported	
FontCharSet	Unsupported	
FontFamily	Unsupported	
FontPitch	Unsupported	
RightToLeft	Unsupported	

Events for StaticText control

Properties | Events | Functions

Event	Support Level	Example Code
Clicked	Supported	Clicked
Constructor	Supported	Constructor
Destructor	Supported	Destructor
DoubleClicked	Supported	DoubleClicked
GetFocus	Supported	GetFocus
LoseFocus	Supported	LoseFocus
RButtonDown	Supported	RButtonDown

DragDrop	Unsupported
DragEnter	Unsupported
DragLeave	Unsupported
DragWithin	Unsupported
Help	Unsupported
Other	Unsupported

Functions for StaticText control

Properties | Events | Functions

Function	Support Level	Example Code
ClassName	Supported	ls_returnvalue = st_test.ClassName()
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = st_test.GetParent()
Hide	Supported	li_returnvalue = st_test.Hide()
Move	Supported	li_returnvalue = st_test.Move(li_x,li_y)
PostEvent	Supported	lb_returnvalue = st_test.PostEvent(event)
Resize	Supported	li_returnvalue = st_test.Resize(li_x,li_y)
SetFocus	Supported	li_returnvalue = st_test.SetFocus() Unsupported: SetFocus(st_test)
SetPosition	Supported	li_returnvalue = st_test.SetPosition(position)
SetRedraw	Supported	li_returnvalue = st_test.SetRedraw(lb_x)
Show	Supported	li_returnvalue = st_test.Show()
TriggerEvent	Supported	li_returnvalue = st_test.TriggerEvent(event)

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

Drag	Unsupported	
GetContextService	Unsupported	
PointerX	Unsupported	
PointerY	Unsupported	
Print	Unsupported	

Tab control

Properties | Events | Functions

Important Requirements

The Tab control must be created in PowerBuilder painter.
The Tab control must be static. It cannot be dynamically created (for example, using the CREATE statement).
Placing a tab control in a custom visual object is not supported.

Properties for Tab control

Properties | Events | Functions

Property	Support Level	Example Code
Alignment	Supported	tab_test. Alignment = 'Center!'
BackColor	Supported	tab_test.BackColor = ll_value tab_test.BackColor = RGB(192,192,192)
BoldSelectedText	Supported	tab_1.BoldSelectedText = TRUE
Control[]	Supported	Userobject luo_1 luo_1 = tab_test.Control[1] Note: this property cannot be dynamically changed.
CreateOnDemand	Supported	tab_test.CreateOnDemand = TRUE
Enabled	Supported	Tab_test.Enabled = FALSE Note: when the Enabled property is set to FALSE, all the controls/objects in the Tab control will turn into gray.
FaceName	Supported	tab_test.FaceName = ls_value
FixedWidth	Supported	tab_test.FixedWidth = lb_value
Height	Supported	tab_test.Height = li_value Note: the height cannot be set to a value lower than 186 in PowerBuilder units.
Italic	Supported	tab_test.Italic = lb_value
Pointer	Supported	tab_test.Pointer= ls_value The Pointer property is partially supported: • Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow! • Unsupported values of the Pointer property:

		AppStarting!, Help!, Icon!, No!, and SizeNS! • User-defined pointers are not supported. • All unsupported pointers are replaced with Arrow! in the Web application.
PowerTips	Supported	tab_test.PowerTips = lb_value
SelectedTab	Supported	tab_test.SelectedTab = li_value
ShowText	Supported	tab_test.ShowText = lb_value
TabOrder	Supported	tab_test.TabOrder = 150
Tag	Supported	tab_test.Tag = ls_value
TextSize	Supported	tab_test.TextSize = li_value
Underline	Supported	tab_test.Underline = lb_value
Visible	Supported	tab_test.Visible = lb_value
Weight	Supported	tab_test.Weight = li_value
Width	Supported	tab_test.Width = li_value
X	Supported	tab_test.X = li_value
Y	Supported	tab_test.Y = li_value
BringToTop	Unsupported	
ClassDefinition	Unsupported	
DragAuto	Unsupported	
DragIcon	Unsupported	
FocusOnButtonDown	Unsupported	
FontCharSet	Unsupported	
FontFamily	Unsupported	
FontPitch	Unsupported	
MultiLine	Unsupported	
PerpendicularText	Unsupported	
PictureOnRight	Unsupported	
RaggedRight	Unsupported	
ShowPicture	Unsupported	
TabPosition	Unsupported	

Events for Tab control

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

Functions for Tab control

Properties | Events | Functions

Function	Support Level	Example Code
ClassName	Supported	ls_returnvalue = tab_test.ClassName()
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = tab_test.GetParent()
Hide	Supported	li_returnvalue = tab_test.Hide()
Move	Supported	li_returnvalue = tab_test.Move(li_x,li_y)
MoveTab	Supported	li_returnvalue = tab_test.MoveTab(li_x,li_y)
PostEvent	Supported	li_returnvalue = tab_test.PostEvent(Clicked!)
Resize	Supported	li_returnvalue = tab_test.Resize(li_x,li_y)
SelectTab	Supported	li_returnvalue = tab_test.SelectTab(tabidentifier) Note: the Tab Identifier can be integer or string type, or DragObject type, but not UserObject type.

SetFocus	Supported	li_returnvalue = tab_test.SetFocus() Unsupported: SetFocus(tab_test)
SetPosition	Supported	li_returnvalue = tab_test.SetPosition(position)
Show	Supported	li_returnvalue = tab_test.Show()
TriggerEvent	Supported	li_returnvalue = tab_test.TriggerEvent(event)

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

CloseTab	Unsupported	
Drag	Unsupported	
GetContextService	Unsupported	
OpenTab	Unsupported	
OpenTabWithParm	Unsupported	
PointerX	Unsupported	
PointerY	Unsupported	
Print	Unsupported	
SetRedraw	Unsupported	
TabPostEvent	Unsupported	
TabTriggerEvent	Unsupported	

Properties for TabPage object

Properties | Events | Functions

APPEON

Property	Support Level	Example Code
BackColor	Supported	tab_test.tabpage_1.BackColor = ll_value
Enabled	Supported	tab_test.tabpage_1.Enabled = lb_value Note: when the Enabled property is set to FALSE, all the controls/objects in the TabPage object will turn to gray.
Pointer	Supported	tab_test.tabpage_1.Pointer= ls_value
PowerTipText	Supported	tab_test.tabpage_1.PowerTipText = ls_value
TabBackColor	Supported	tab_test.tabpage_1.TabBackColor = ll_value
TabTextColor	Supported	tab_test.tabpage_1.TabTextColor = ll_value
Tag	Supported	ls_value = tab_test. tabpage_1.Tag

Text	Supported	ls_value = tab_test. tabpage_1.Text
Visible	Supported	tab_test. tabpage_1.Visible = lb_value
PictureMaskColor	Unsupported	
PictureName	Unsupported	

Events for Tabpage object

Properties | Events | Functions

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

Functions for Tabpage object

Properties | Events | Functions

Function	Support Level	Example Code
ClassName	Supported	ls_value = tabpage_1.ClassName()
GetParent	Supported	lobj_returnvalue = tabpage_1.GetParent()
Hide	Supported	li_returnvalue = tabpage_1.Hide()
PostEvent	Supported	lb_returnvalue = tabpage_1.PostEvent(event)
Show	Supported	li_returnvalue = tabpage_1.Show()
TriggerEvent	Supported	li_returnvalue = tabpage_1.TriggerEvent(event)

TypeOf	Supported	if tabpage_1.TypeOf() = tabpage! Then
		ls_returnvalue='tabpage!'
		end if

CreatePage	Unsupported	
Drag	Unsupported	
GetContextService	Unsupported	
PageCreated	Unsupported	

PointerX	Unsupported
PointerY	Unsupported
SetRedraw	Unsupported

TreeView control

Properties | Events | Functions

APPEON

Important Requirements

The TreeView control must be created in PowerBuilder painter.
The TreeView control must be static. It cannot be dynamically created (for example, using the CREATE statement).
The length of the converted TreeView labels has minor differences to those in PowerBuilder.

Properties for TreeView control

Properties | Events | Functions

APPEON

Property	Support Level	Example Code
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!
CheckBoxes	Supported	tv_1.CheckBoxes = lb_value
DeleteItems	Supported	tv_1.DeleteItems = lb_value
EditLabels	Supported	tv_1.EditLabels = lb_value
Enabled	Supported	tv_1.Enabled = lb_value
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
Italic	Supported	tv_1.Italic = lb_value

LinesAtRoot	Supported	tv_1.LinesAtRoot = lb_value
PictureName	Supported	tv_1.PictureName = ls_value
Pointer	Supported	tv_1.Pointer = ls_value The Pointer property is partially supported: ● Supported values of the Pointer property: Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, and UPArrow! ● Unsupported values of the Pointer property: AppStarting!, Help!, Icon!, No!, and SizeNS! ● User-defined pointers are not supported. ● All unsupported pointers are replaced with Arrow! in the Web application.
StatePictureName[ ]	Supported	ls_value[] = tv_1.StatePictureName[]
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
Underline	Supported	tv_1.Underline = lb_value
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
Accelerator	Unsupported	
BringToTop	Unsupported	
ClassDefinition	Unsupported	
DisableDragDrop	Unsupported	
DragAuto	Unsupported	
DragIcon	Unsupported	
FontCharSet	Unsupported	
FontFamily	Unsupported	

FontPitch	Unsupported
HideSelection	Unsupported
ImeMode	Unsupported
Indent	Unsupported
PictureHeight	Unsupported
PictureMaskColor	Unsupported
PictureWidth	Unsupported
SingleExpand	Unsupported
SortType	Unsupported
StatePictureHeight	Unsupported
StatePictureMaskColor	Unsupported
StatePictureWidth	Unsupported
ToolTips	Unsupported
TrackSelect	Unsupported

Events for TreeView control

Properties | Events | Functions

Event	Support Level	Example Code
Clicked	Supported	Clicked
Constructor	Supported	Constructor
DeleteItem	Supported	DeleteItem(handle)
Destructor	Supported	Destructor
DoubleClicked	Supported	DoubleClicked(handle)
GetFocus	Supported	GetFocus
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)
SelectionChanged	Supported	SelectionChanged(oldhandle,newhandle)
SelectionChanging	Supported	SelectionChanging(oldhandle,newhandle)
BeginDrag	Unsupported	
BeginLabelEdit	Unsupported	
BeginRightDrag	Unsupported	

DragDrop	Unsupported
DragEnter	Unsupported
DragLeave	Unsupported
DragWithin	Unsupported
EndLabelEdit	Unsupported
Help	Unsupported
Key	Unsupported
Other	Unsupported
RightDoubleClicked	Unsupported
Sort	Unsupported

Functions for TreeView control

Properties | Events | Functions

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) Note: the *itemhandle* argument can only be 0.
DeleteItem	Supported	li_returnvalue = tv_1.DeleteItem(itemhandle) Note: it is unsupported to directly specify a value as the function argument, because the handle of the item might be named differently in PowerScript than in JavaScript. However, it is supported to use 0 as the function argument (e.g. DeleteItem(0)). Using the return value of the function FindItem as the function argument is supported.
DeletePicture	Supported	li_returnvalue = tv_1.DeletePicture(index)
DeletePictures	Supported	li_returnvalue = tv_1.DeletePictures()
DeleteStatePicture	Supported	li_returnvalue = tv_1.DeleteStatePicture(index)
DeleteStatePictures	Supported	li_returnvalue = tv_1.DeleteStatePictures()
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)
GetParent	Supported	lobj_returnvalue = tv_1.GetParent()
Hide	Supported	li_returnvalue = tv_1.Hide()
InsertItem	Supported	ll_returnvalue = tv_1.InsertItem (handleparent,handleafter,item) 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)
Move	Supported	li_returnvalue = tv_1.Move(li_xpos,li_ypos)
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)
SetFocus	Supported	li_returnvalue = tv_1.SetFocus() Unsupported: SetFocus(tv_test)
SetItem	Supported	li_returnvalue = tv_1.SetItem(itemhandle,item)
SetPosition	Supported	li_returnvalue = tv_1.SetPosition(position)
Show	Supported	li_returnvalue = tv_1.Show()
TriggerEvent	Supported	li_returnvalue = tv_1.TriggerEvent(event)
TypeOf	Supported	if tv_1.TypeOf() = TreeView! Then ls_returnvalue='TreeView!'

end if

Drag	Unsupported
GetContextService	Unsupported
GetItemAtPointer	Unsupported
InsertItemSort	Unsupported
PointerX	Unsupported
PointerY	Unsupported
Print	Unsupported
SetDropHighlight	Unsupported
SetFirstVisible	Unsupported
SetLevelPictures	Unsupported
SetOverlayPicture	Unsupported
SetRedraw	Unsupported
Sort	Unsupported
SortAll	Unsupported

Graph control

APPEON

This control, including its events, properties and functions, is not supported.

HProgressBar control

APPEON

This control, including its events, properties and functions, is not supported.

HScrollBar control

APPEON

This control, including its events, properties and functions, is not supported.

HTrackBar control

APPEON

This control, including its events, properties and functions, is not supported.

OLEControl control

APPEON

This control, including its events, properties and functions, is not supported.

OLECustomControl control

APPEON

This control, including its events, properties and functions, is not supported.

Oval control

APPEON

This control, including its events, properties and functions, is not supported.

RichTextEdit control

APPEON

This control, including its events, properties and functions, is not supported.

RoundRectangle control

APPEON

This control, including its events, properties and functions, is not supported.

VProgressBar control

APPEON

This control, including its events, properties and functions, is not supported.

VScrollBar control

APPEON

This control, including its events, properties and functions, is not supported.

VTrackBar control

APPEON

This control, including its events, properties and functions, is not supported.

System objects

APPEON

Supported

The following are the system objects that Appeon supports:

Application	DataStore	Connection	CORBAObject
DataWindowChild	DragObject	DWObject	DynamicStagingArea
GraphicObject	ListViewItem	MDIClient	Menu
Message	NonVisualObject	OLEObject	PowerObject
Transaction Object	TreeViewItem	UserObject	Window
WindowObject			

- Application, MDIClient, Message, and Window objects cannot be dynamically created (for example, 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. For more information, see the Object naming requirement section.

- It is unsupported to equate the class name of a system class type variable with the function ClassName. The return value will be incorrect on the Web. For example:
```
Transaction      gtr_trans      //gtr_trans is a system class type variable
String           ls_arg
gtr_trans = create transaction
ls_arg = gtr_trans.ClassName()     // Unsupported
```

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

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	CORBAUserException
CPlusPlus	DivideByZeroError	DWRuntimeError	DynamicDescriptionArea
EnumerationDefinition	EnumerationItemDefinition	Environment	Error
ErrorLogging	Exception	ExtObject	Graph

grAxis	grDispAttr	Inet	InternetResult
JaguarORB	mailFileDescription	mailMessage	mailRecipient
mailSession	MenuCascade	NullObjectError	OLECustomControl
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	Timing	TraceActivityNode	TraceBeginEnd
TraceError	TraceESQL	TraceFile	TraceGarbageCollect
TraceLine	TraceObject	TraceRoutine	TraceTree
TraceTreeError	TraceTreeESQL	TraceTreeGarbageCollect	TraceTreeLine
TraceTreeNode	TraceTreeObject	TraceTreeRoutine	TraceTreeUser
TraceUser	TransactionServer	Transport	TypeDefinition
VariableCardinalityDefinition	VariableDefinition		

Application object

Properties | Events | Functions

Important Requirements

There can only be one Application object in an application.

A commandline argument in the application Open event is supported.

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.

An application object cannot be dynamically created (for example, using the CREATE statement); it must be defined as a static object, created in PowerBuilder painter.

It is recommended that the use of Application object is limited to the following well-supported actions:

- Launching login screens.
- Opening and closing MDI, SDI, Main, Response and Popup windows.
- Creating instances of one or more transaction objects.

Additional properties of Application object:

- The type of SQLCA can be a transaction user object.
- The type of SQLSA can be a DynamicStagingArea user object.
- The variable type MESSAGE can use the default value "message" or a user-defined message object.
- The user can set a global variable Message object as a user object in the application painter as follows: through the Properties window --> General tab page --> Additional Properties button --> Variable Types tab page.
  The user can also customize the Message object by defining a class user object inherited from the built-in Message object.
- The specification of application icon will be ignored .
- The specification of SQLDA and Error will be ignored.

Properties for Application object

Properties | Events | Functions

APPEON

Property	Support Level	Example Code
AppName	Supported	String ls_AppName ls_AppName = this.AppName //Get the AppName property
DisplayName	Supported	String ls_DisplayName ls_DisplayName = this.DisplayName
MicroHelpDefault	Supported	
ToolbarFrameTitle	Supported	ls_value = this.ToolbarFrameTitle
ToolbarSheetTitle	Supported	ls_value = this.ToolbarSheetTitle
ToolbarTips	Supported	lb_value = this.ToolbarTips
ClassDefinition	Unsupported	
DDETimeOut	Unsupported	
DWMessageTitle	Unsupported	
FreeDBLibraries	Unsupported	
RightToLeft	Unsupported	
ToolbarPopMenuText	Unsupported	
ToolbarText	Unsupported	
ToolbarUserControl	Unsupported	

Events for Application object

Properties | Events | Functions

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

Functions for Application object

Properties | Events | Functions

Function	Support Level	Example Code
ClassName	Supported	ls_class = l_app.ClassName()
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	
GetParent	Unsupported	
SetLibraryList	Unsupported	
SetTransPool	Unsupported	

Note: in PowerBuilder 8.0 and subsequent versions, GetLibrary and SetLibrary are global functions.

Connection object

Properties | Events | Functions

Important Requirements

- Connection object can be created dynamically, using the CREATE statement.

Properties for Connection object

Properties | Events | Functions

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

Events for Connection object

Properties | Events | Functions

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

Functions for Connection object

Properties | Events | Functions

Function	Support Level	Example Code
ClassName	Supported	ls_name = Myconnect.ClassName()

ConnectToServer	Supported	ll_rc = Myconnect.ConnectToServer()
CreateInstance	Supported	ll_rc = Myconnect.CreateInstance (ln_tools,"simpletest/nvo_tools")
DisconnectServer	Supported	ll_rc = Myconnect.disconnectserver()
GetParent	Supported	lobj_parent = Myconnect.GetParent()
PostEvent	Supported	lb_rc = Myconnect.PostEvent(event)
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	
RemoteStopConnection	Unsupported	
RemoteStopListening	Unsupported	

CORBAObject object

Properties | Events | Functions

Important Requirements

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

Properties for CORBAObject object

Properties | Events | Functions

Property	Support Level	Example Code
ClassDefinition	Unsupported	

Events for CORBAObject object

Properties | Events | Functions

Event	Support Level	Example Code

Constructor	Supported	Constructor
Destructor	Supported	Destructor

Functions for CORBAObject object

Properties | Events | Functions

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”
		END IF

_Is_A	Unsupported	
_Narrow	Unsupported	
GetContextService	Unsupported	

DynamicStagingArea object

Properties | Events | Functions

Important Requirements

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

Properties for DynamicStagingArea object

Properties | Events | Functions

Property	Support Level	Example Code
ClassDefinition	Unsupported	

Events for DynamicStagingArea object

Properties | Events | Functions

Event	Support Level	Example Code
Constructor	Unsupported	
Destructor	Unsupported	

Functions for DynamicStagingArea object

Properties | Events | Functions

Function	Support Level	Example Code
ClassName	Supported	ls_class = dsa_1.ClassName()
GetParent	Supported	lobj_parent_name = this.GetParent ()
PostEvent	Supported	this.PostEvent (Clicked!) this.PostEvent ("Clicked!")
TriggerEvent	Supported	this.TriggerEvent (Clicked!) this.TriggerEvent ("Clicked!")
TypeOf()	Supported	SQLSA.TypeOf()
GetContextService	Unsupported	

ListViewItem object

Properties | Functions

Properties for ListViewItem object

Properties | Functions

Property	Support Level	Example Code
Data	Supported	any_value = lvi_item1.Data
Label	Supported	ls_value = lvi_item1.Label In PowerBuilder, 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.
PictureIndex	Supported	li_value = lvi_item1.PictureIndex
Selected	Supported	lb_value = lvi_item1.Selected
StatePictureIndex	Supported	li_value = lvi_item1.StatePictureIndex
ClassDefinition	Unsupported	
CutHighlighted	Unsupported	
DropHighlighted	Unsupported	
HasFocus	Unsupported	

ItemX	Unsupported
ItemY	Unsupported
OverlayPictureIndex	Unsupported

Functions for ListViewItem object

Properties | Functions

APPEON

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

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

GetContextService	Unsupported	
GetParent	Unsupported	

MDIClient object

Properties | Functions

APPEON

Important Requirements

The MDIClient object cannot be dynamically created (for example, using the CREATE statement); it must be defined as a static object, created in a PowerBuilder painter.

The MDI can only have one MDIClient without controls on top of it.

Multiple MDI windows are supported.

Defining the size of the client area in a 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.

Properties for MDIClient object

Properties | Functions

APPEON

Property	Support Level	Example Code
BackColor	Supported	w_main.mdi_1.BackColor = ll_value
Height	Supported	w_main.mdi_1.Height = li_value
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	
MicroHelpHeight	Unsupported	
Visible	Unsupported	

Functions for MDIClient object

Properties | Functions

APPEON

Function	Support Level	Example Code
ClassName	Supported	ls_classname = w_main.mdi_1.Classname()
GetParent	Supported	PowerObject lpo_returnvalue lpo_returnvalue = this.GetParent()
TypeOf	Supported	ls_type = w_main.mdi_1.TypeOf()
GetContextService	Unsupported	
Hide	Unsupported	
Move	Unsupported	
Resize	Unsupported	
SetRedraw	Unsupported	
Show	Unsupported	

OLEObject object

Properties | Events | Functions

APPEON

Important Requirements

OLEObject objects are supported. OLEControl controls remain unsupported.

OLEObject object does not support cascaded calling. The following example is not supported: OleObject.func1.func2.

It is required to use parentheses when calling an OLEObject method.

Properties for OLEObject object

Properties | Events | Functions

APPEON

No properties are supported.

Events for OLEObject object

Properties | Events | Functions

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

Functions for OLEObject object

Properties | Events | Functions

Function	Support Level
ClassName	Supported
ConnectToNewObject	Supported
DisconnectObject	Supported
GetParent	Supported
IsAlive	Supported
TriggerEvent	Supported
TypeOf	Supported
ConnectToNewRemoteObject	Unsupported
ConnectToObject	Unsupported
ConnectToRemoteObject	Unsupported
GetAutomationNativePointer	Unsupported
GetContextService	Unsupported
PostEvent	Unsupported
ReleaseAutomationNativePointer	Unsupported
SetAutomationLocale	Unsupported
SetAutomationPointer	Unsupported
SetAutomationTimeout	Unsupported

Menu object

Properties | Events | Functions

Important Requirements

- Menus must be defined in the PowerBuilder painter, and 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

Toolbar features

Properties for Menu object

Properties | Events | Functions

Property	Support Level	Example Code
Checked	Supported	m_function.m_testitemforfunction.Checked = lb_value If a menu item has submenu item(s), when the Check property of the submenu item(s) is/are set to TRUE during the Web application is running, a checked sign will appear to the left of the item. However, this does not happen in PowerBuilder.
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
Tag	Supported	m_function.m_testitemforfunction.Tag = ls_value
Text	Supported	m_function.m_testitemforfunction.Text = ls_value The Text property cannot be an empty string (“”). The Text property is unsupported for root menu item.
ToolbarItemBar Index	Supported	m_function.m_testitemforfunction.ToolbarItemBarIndex = li_value
ToolbarItemDown Name	Supported	m_function.m_testitemforfunction.ToolbarItemDownName = ls_value

ToolbarItemName	Supported	m_function.m_testitemforfunction.ToolbarItemName = ls_value
ToolbarItemOrder	Supported	m_function.m_testitemforfunction.ToolbarItemOrder = li_value
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	
Default	Unsupported	
MenuItemType	Unsupported	
MergeOption	Unsupported	
ShiftToRight	Unsupported	
Shortcut	Unsupported	
ToolbarItemDown	Unsupported	
ToolbarItemSpace	Unsupported	

Events for Menu object

Properties | Events | Functions

Event	Support Level	Example Code
Clicked	Supported	Clicked
Selected	Supported	Selected
Help	Unsupported	

Functions for Menu object

Properties | Events | Functions

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) Notes: 1. The popup menu cannot have the same items as the top menu. 2. It is unsupported to pop menu in a LeftClick event. 3. In PowerBuilder, it is possible to popup a menu first and then destroy it in an event. Because the execution of PopMenu is immediately followed by the execution of the subsequent script on the Web, there will be no popup menu in this case. 4. When a window is not maximized in the Web application, the popup menu displays at the cursor position. This is the same as in PowerBuilder. However, in a maximized window, the height of the window toolbar will be added to the value of the PointerY property. As a result, the popup menu will display at a certain distance from the bottom of the cursor position.
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	

Menu features

Properties | Events | Functions

APPEON

Menu Features

Supported

- Menu object may be implemented with the following functionalities:

    - Modifying menu items based on user rights using the enable and visible properties.
    - 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 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 object can use the ChangeMenu function to dynamically change the menu. See the ChangeMenu function for Window object.

- 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.

- If the width of the window is not enough to fully display the menu items, the menu items will still be displayed. However, the PowerBuilder application will automatically display the items in two or more lines, while the Web application will show a vertical scrollbar for the user to view the items that cannot be displayed in one line.

- Append the names of open sheets (window list) to a menu item.

Unsupported

- Using the UpArrow/DownArrow/LeftArrow/RightArrow to change the currently selected menu item is not supported.

- Shortcut keys for menu items are not supported.

- It is unsupported 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.
  In PowerBuilder, the following two lines of code in the w_citysheet script will have the same effect:
  m_main.m_file.enabled = FALSE //Unsupported by Appeon
  m_city.m_file.enabled = FALSE //Supported by Appeon

- 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.

- Appeon does not support the ShiftToRight property but has the following rule for displaying the menu items on the Web: when other menu objects are added in a descendent menu, the menu objects that are inherited shift up and to the left, and the menu objects newly added shift down and to the right. More levels of dropdown menus under a menu bar item are supported.

- 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.

Toolbar features

Properties | Events | Functions

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.
- Clicking the toolbar item triggers the same event as clicking the relevant menu item.
- Multiple toolbars are supported.
- Multiple toolbars can be displayed at the same time, but docked toolbar or toolbar toggle are not supported.
- If the width of the window is not enough to fully display the full toolbar, the toolbar items will still be displayed. The PowerBuilder application will automatically display the items in two or more lines, while the Web application will show a horizontal or vertical scrollbar for the user to view the items that cannot be displayed in one line.

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.
- A single window can only have a single toolbar line. If multiple toolbars are defined for a window, the multiple toolbars will be displayed at one line.
- 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.

Message object

Properties | Events | Functions

Important Requirements

The Message object cannot be dynamically created (for example, using the CREATE statement); it must be defined in a PowerBuilder painter.

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

Properties for Message object

Properties | Events | Functions

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

Events for Message object

Properties | Events | Functions

APPEON

Event	Support Level	Example Code
Constructor	Unsupported	
Destructor	Unsupported	

Functions for Message object

Properties | Events | Functions

APPEON

Function	Support Level	Example Code
ClassName	Supported	
PostEvent	Supported	
TriggerEvent	Supported	
TypeOf	Supported	
GetContextService	Unsupported	
GetParent	Unsupported	

Transaction object

Properties | Events | Functions

APPEON

Important Requirements

The Transaction object can be global or local.

The user-defined Transaction object is supported.

If passing parameters to transaction objects, parameters must be passed by reference. If passing parameters by value, it is treated as local, which is not supported.

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.

Properties for Transaction object

Properties | Events | Functions

Appeon ignores the dynamic property settings for the connection cache and Transaction Object except for the AutoCommit property.

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
DBParm	Supported	i_sqlca.DBParm= "ConnectionString= 'DSN=AppeonSample;UID=dba;PWD=sql'" //DBMS-specific parameters Supported parameters: CommitOnDisconnect and CacheName. In PowerBuilder, the connection cache name only applies when a PowerBuilder NVO is deployed to EAServer. In Appeon, the connection cache specified applies for the connection of the deployed application to the database. Settings to the other parameters in the script will be ignored.
LogID	Supported	ls_value = sqlca.LogID
LogPass	Supported	ls_value = sqlca.LogPass
ServerName	Supported	ls_value = sqlca.ServerName
SQLCode	Supported	ll_value = sqlca.SQLCode

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 a Web application, SQLDBCode returns a different value from PowerBuilder. It is 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	
DBPass	Unsupported	
Lock	Unsupported	
SQLReturnData	Unsupported	

Events for Transaction object

Properties | Events | Functions

Event	Support Level	Example Code
Constructor	Supported	Constructor
Destructor	Supported	Destructor

Functions for Transaction object

Properties | Events | Functions

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

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

DBHandle	Unsupported	
GetContextService	Unsupported	
GetParent	Unsupported	
PostEvent	Unsupported	
SyntaxFromSQL	Unsupported	
TriggerEvent	Unsupported	

TreeViewItem object

Properties | Functions

Properties for TreeViewItem object

Properties | Functions

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.
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
StatePictureIndex	Supported	li_value = tv_treeviewitem.StatePictureIndex
ClassDefinition	Unsupported	
CutHighLighted	Unsupported	
DropHighLighted	Unsupported	
OverlayPictureIndex	Unsupported	

Functions for TreeViewItem object

Properties | Functions

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	

UserObject object

Properties | Events | Functions

Important Requirements

The UserObject object cannot be dynamically created (for example, using the CREATE statement); it must be defined in the PowerBuilder painter.

Properties for UserObject object

Properties | Events | Functions

Property	Support Level	Example Code
BackColor	Supported	uo_test.BackColor = ll_value
Border	Supported	uo_test.Border = lb_value
BorderStyle	Supported	uo_test.BorderStyle = StyleLowered!
ClassName	Supported	ls_value = uo_test.ClassName
Control[]	Supported	WindowObject lobj_control[] lobj_control[] = uo_test.control[]
Enabled	Supported	uo_test.Enabled = TRUE Note: when the Enabled property of a UserObject is set to FALSE, the controls or objects in the UserObject will be grayed.
Height	Supported	uo_test.Height = 300
Pointer	Supported	uo_test.Pointer=ls_value
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
Width	Supported	uo_test.Width = li_value
X	Supported	uo_test.X = li_value
Y	Supported	uo_test.Y = li_value
BringToTop	Unsupported	
ClassDefinition	Unsupported	
ColumnsPerPage	Unsupported	
DragAuto	Unsupported	
DragIcon	Unsupported	
HScrollBar	Unsupported	
LibraryName	Unsupported	
LinesPerPage	Unsupported	
ObjectType	Unsupported	
PictureMaskColor	Unsupported	
PictureName	Unsupported	
Style	Unsupported	
UnitsPerColumn	Unsupported	
UnitsPerLine	Unsupported	
VScrollBar	Unsupported	

Events for UserObject object

Properties | Events | Functions

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

DragWithin	Unsupported
Help	Unsupported
Other	Unsupported

Note: the Constructor and Destructor events cannot be triggered for a UserObject Control[] array.

Functions for UserObject object

Properties | Events | Functions

Function	Support Level	Example Code
ClassName	Supported	uo_test.classname()
GetParent	Supported	lobj_parent_name = this.GetParent()
Hide	Supported	uo_test.Hide()
Move	Supported	uo_test.Move(x1,y1)
PostEvent	Supported	uo_test.PostEvent(Clicked!)
Resize	Supported	uo_test.Resize(x1,y1)
SetPosition	Supported	uo_test.SetPosition(ToTop!) uo_two.SetPosition(Behind!, uo_three)
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	
Drag	Unsupported	
GetContextService	Unsupported	
InsertItem	Unsupported	
PageCreated	Unsupported	
PointerX	Unsupported	
PointerY	Unsupported	

Print	Unsupported
SetFocus	Unsupported
SetRedraw	Unsupported

Window object

Properties | Events | Functions

APPEON

Important Requirements

There are six window types: Child, Main, MDI, MDIHelp, Popup, and Response. Refer to the Window types page for more information on window types.

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

Controls in a window

Window variables

Opening and closing windows

User operation in windows

Properties for Window object

Properties | Events | Functions

APPEON

Property	Support Level	Example Code
BackColor	Supported	w_1.BackColor = RGB(0, 0, 255)
Border	Supported	w_test.Border = TRUE
Center	Supported	w_test.Center = TRUE Note: this property is unsupported for MDI and MDIHelp windows.
Control [ ]	Supported	Control[] graphicobject lobj_control lobj_control = w_city.control[1] Note: this property cannot be dynamically changed.
ControlMenu	Supported	w_test.ControlMenu = TRUE

		Note: this property can be dynamically changed only if the window is Sheet window or Child window. The ControlMenu property of MDI, MDIHelp, Main, PopUp, Response window will always be TRUE in Appeon.
Height	Supported	w_1.Height = 750 //Specifies the height of the window w_1 Notes: 1) The Height property is unsupported for an MDI frame window. 2) For a Web Popup, Main or Response window, the minimum height is 400 in PowerBuilder units, and the minimum width is 457. 3) For a Web sheet or a Child window, the minimum height is 108 in PowerBuilder units, and the minimum width is 731.
HScrollBar	Supported	Parent.HScrollBar = TRUE Note: HScrollBar is unsupported for Popup and Main windows.
Icon	Supported	w_city.Icon = 'Rectangle!' Note: only the sheet window and the child window are supported for this property.
MaxBox	Supported	Parent.MaxBox = TRUE Note: the MaxBox property of MDI, MDIHelp, Main, PopUp, Response Window will always be TRUE in Appeon.
MenuID	Supported	Menu menuvar = w_test.MenuID
MenuName	Supported	ls_value = w_test.MenuName
MinBox	Supported	parent. MinBox = TRUE Note: the MinBox property of MDI, MDIHelp, Main, PopUp, Response Window will always be TRUE in Appeon.
Resizable	Supported	parent. Resizable = TRUE Note: for MDI frame window or Main window, the Resizable property will always be TRUE. 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.
Tag	Supported	w_frame.SetMicroHelp(This.Tag) //Set or get the Tag property
Title	Supported	w_mdi.Title = "Monthly Report" Note: the Title property of MDI, MDIHelp, Main, PopUp, Response Window cannot be dynamically changed.
ToolbarVisible	Supported	parent.ToolbarVisible = TRUE Note: the ToolBarVisible is unsupported for Popup and Main windows.

Visible	Supported	w_city.Visible = TRUE Note: this property is unsupported for sheets whose window type is Response, Popup or Main windows.
VScrollBar	Supported	Parent. VScrollBar = TRUE
Width	Supported	dw_1.Width = 750 //Set or get the Width property Note: the Width property is unsupported for an MDI frame window. For a Web Popup, Main or Response window, the minimum height is 400 in PowerBuilder units, and the minimum width is 457. For a Web sheet or a Child window, the minimum height is 108 in PowerBuilder units, and the minimum width is 731.
WindowState	Supported	W_city.WindowState = Maximized! Note: it is supported to set the WindowState property in script. However, it is unsupported to set the WindowState property in the painter. The initial state for all the windows is always Normal! It is possible for WindowState to control the window state but not Internet Explorer state.

WindowType	Supported	IF This.WindowType=Main! THEN
		MessageBox("This is a main window.")
		END IF
		Note: the MDI WindowType is not supported. If the window is MDI, the return value will be MDIHelp.

X	Supported	dw_1.X = 215 //Set or get the X property Note: the X property is not supported for the MDI frame window. It is unsupported to move a window out of the screen by modifying its X property.
Y	Supported	dw_1.Y = 215 //Set or get the Y property Note: the Y property is not supported for the MDI frame window. It is unsupported to move a window out of the screen by modifying its Y property.
BringToTop	Unsupported	
ClassDefinition	Unsupported	
ClientEdge	Unsupported	
ColumnsPerPage	Unsupported	
ContextHelp	Unsupported	
Enabled	Unsupported	
KeyboardIcon	Unsupported	
LinesPerPage	Unsupported	

PaletteWindow	Unsupported
Pointer	Unsupported
RightToLeft	Unsupported
TitleBar	Unsupported
ToolbarAlignment	Unsupported
ToolbarHeight	Unsupported
ToolbarWidth	Unsupported
ToolbarX	Unsupported
ToolbarY	Unsupported
UnitsPerColumn	Unsupported
UnitsPerLine	Unsupported

Events for Window object

Properties | Events | Functions

Event	Support Level	Example Code
Activate	Supported	Activate Notes: 1) With the Web applications, the Activate/Deactivate events will not triggered if the user shifts the focus from a window in the current application to a place outside the application. 2) Activate and Deactivate events are unsupported for dialogue boxes.
Clicked	Supported	Clicked Appeon does not support triggering the Clicked event twice by double clicking the window title bar. The Clicked event is only triggered once.
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 Note: in PowerBuilder, if the CloseQuery event returns 1, the closing of the Window is aborted. With Web applications, this rule is only effective for sheets.
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. 2) Activate and Deactivate events are unsupported for dialog boxes.
DoubleClicked	Supported	DoubleClicked
Hide	Supported	Hide Note: with the Window types Main, Popup, and Response, the event cannot be automatically triggered.
Key	Supported	Key
MouseDown	Supported	MouseDown
MouseMove	Supported	MouseMove
MouseUp	Supported	MouseUp
Open	Supported	Open Note: in Appeon, the window creation and UI-painting occur before the Open event is triggered.
RButtonDown	Supported	RButtonDown
Resize	Supported	Resize When the Internet Explorer window is resized, no Resize event is triggered for the windows in the Web application.
Show	Supported	Show
Timer	Supported	Timer
DragDrop	Unsupported	
DragEnter	Unsupported	
DragLeave	Unsupported	
DragWithin	Unsupported	
Help	Unsupported	
HotLinkAlarm	Unsupported	
Other	Unsupported	
RemoteExec	Unsupported	

RemoteHotLinkStart	Unsupported
RemoteHotLinkStop	Unsupported
RemoteRequest	Unsupported
RemoteSend	Unsupported
SystemKey	Unsupported
ToolbarMoved	Unsupported

Functions for Window object

Properties | Events | Functions

Function	Support Level	Example Code
ArrangeSheets	Supported	w_mdi.ArrangeSheets(arrangetype)
ChangeMenu	Supported	Parent.ChangeMenu(m_test_menu2) Notes: 1) Specifying the value of the *position* argument will be ignored. 2) It is unsupported to use a menu type variable as the *menuname* argument in the ChangeMenu function. For example, the following syntax is unsupported: m_menu_b lm_var //lm_var is a menu variable ChangeMenu(lm_var) //Unsupported 3) It is unsupported to use a menu item as the *menuname* argument. 4) 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()
GetActiveSheet	Supported	w_test. GetActiveSheet()
GetFirstSheet	Supported	w_test. GetFirstSheet()
GetNextSheet	Supported	w_test. GetNextSheet()
Hide	Supported	Parent.Hide() Note: this function is unsupported for Response, Popup and Main windows because Dialog window is unable to show/hide in the Internet Explorer browser.
Move	Supported	w_test.Move(X,Y)
ParentWindow	Supported	w_parent = w_test .parentwindow()

PointerX	Supported	li_dist = w_city.PointerX() Note: it is unsupported to execute the PointerX function for a window that does not have focus. In a Web application, it is unsupported for a sheet to call the PointerX function of an MDI frame that contains the sheet.
PointerY	Supported	li_dist = w_city.PointerY() Note: it is unsupported to execute the PointerY function for a window that does not have focus. In a Web application, it is unsupported for a sheet to call the PointerY function of an MDI frame that contains the sheet.
PostEvent	Supported	W_main.PostEvent(Clicked!)
Resize	Supported	w_test.Resize(300,250)
SetFocus	Supported	w_test.SetFocus() Unsupported: SetFocus(w_test)
SetMicroHelp	Supported	W_test.SetMicroHelp (“Preview”)
SetPosition	Supported	w_city.SetPosition(position [, precedingwindow]) Note: the window with property TopMost may not always be the top one in Appeon.
Show	Supported	Parent.Show() Note: this function is unsupported for Popup and Main windows because Dialog window is unable to show/hide in the Internet Explorer browser.
TriggerEvent	Supported	w_test.TriggerEvent(close!) The following TriggerEvent syntax is unsupported: *object1.*TriggerEvent (*object2, event).*

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()
CloseChannel	Unsupported	
CloseUserObject	Unsupported	
ExecRemote	Unsupported	
GetCommandDDE	Unsupported	

GetCommandDDEOrigin	Unsupported
GetContextService	Unsupported
GetDataDDE	Unsupported
GetDataDDEOrigin	Unsupported
GetParent	Unsupported
GetRemote	Unsupported
GetToolbar	Unsupported
GetToolbarPos	Unsupported
OpenChannel	Unsupported
OpenUserObject	Unsupported
OpenUserObjectWithParm	Unsupported
Print	Unsupported
RespondRemote	Unsupported
SetDataDDE	Unsupported
SetRedraw	Unsupported
SetRemote	Unsupported
SetToolbar	Unsupported
SetToolbarPos	Unsupported
StartHotLink	Unsupported
StartServerDDE	Unsupported
StopHotLink	Unsupported
StopServerDDE	Unsupported

Window types

Properties | Events | Functions

APPEON

Window Types

1) MDI sheet

- MDI sheet refers to the window that is opened with the function OpenSheet or OpenSheetWithParm.
- With Web applications, an MDI sheet is more like a separate window type, and the supported features for MDI sheet are more comprehensive than the supported features for the specific window type. For example, if a Main window is opened through the function OpenSheet, the MinBox and MaxBox properties that are unsupported for Main window are supported for the MDI sheet.
- With Web applications, a MDI sheet will be always behind the windows that are opened with the function Open.

2) MDI and MDIHelp

- The window type MDIHelp is treated the same as the window type MDI. The status bar in the

MDIHelp window will not display.

- It is supported to place any controls in an MDIHelp or MDI window.
- The initial state of the window is always Normal!.
- If the MDI window for a Web application is closed, the whole Web application is closed. This is different from PowerBuilder. With the PowerBuilder application, if there is a Main window open before the MDI window is opened, the application returns to the Main window when the MDI window is closed.

3) Main

- The MinBox and MaxBox properties are unsupported for Main windows.
- The Title of a Main window cannot be dynamically specified.
- The setting of ControlMenu for Main window will be ignored.
- The initial state for the window is always Normal!.
- In PowerBuilder, the MDI menu is displayed in front of Main window. In Web applications, the MDI menu will be displayed behind the Main window. This may be inconvenient for the user to select menu items.
- The user cannot trigger any event (including Activate event and Deactivate event) that will be triggered by clicking the title bar of a Main window.

4) Child

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

5) Popup

- The MinBox and MaxBox properties are unsupported for Popup windows.
- The Title of a Popup window cannot be dynamically specified.
- The setting of ControlMenu for Popup window will be ignored.
- The initial state for the window is always Normal!.
- The user cannot trigger any event (including Activate event and Deactivate event) will be triggered by clicking the title bar of a Popup window.

6) Response

- Loading a menu for a Response window with the ChangeMenu function is supported.
- In PowerBuilder, if a Response window gets focus, the focus cannot move to another location unless the Response window is closed. In Web applications, the focus cannot move to the parent window of the Response window but, it can move to another location.

- The MinBox and MaxBox properties are unsupported for Response windows.
- The Title of a Response window cannot be dynamically specified.
- The initial state for the window is always Normal!.
- The setting of ControlMenu for Response window will be ignored.
- When a Response window opens after the open of a Main window, the Response window will always get focus. In other cases, the Response window will have no focus when opened.
- The user cannot trigger any event (including Activate event and Deactivate event) that will be triggered by clicking the title bar of a Response window.

Controls in a window

Properties | Events | Functions

APPEON

Controls in a Window

- If the TextSize of a control is set as a negative value (for example, -3), the text will display correctly on the Web, but it will be unsupported to get the value of the TextSize property. If the TextSize is set to a positive value, the value of the property can be obtained.
- If setting the Height property of some object/control to 0, the object/control may not become invisible on the Web. However, a portion of it may still display.
- It is unsupported to write the following script in a control: MinBox = FALSE

Window variables

Properties | Events | Functions

APPEON

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 unsupported to dynamically create a window instance by using CREATE. For example, the following syntax is not supported:

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

- With Web applications, if the window instance is a global variable, the user can open several instances of the same window by calling the Open function for several times. This is different from PowerBuilder.

Opening and closing windows

Opening Windows

- Open functions

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

Open (windowvar, windowtype {, parent })
In the Web application, when a window is opened with an Open function, the *windowvar* argument cannot be a window object.

- It is unsupported to open a window by calling a user object function.

- Triggering of events

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 >> Activate >> Show.

- Front-to-back order

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.

Different from the PowerBuilder application, in Web applications, the descendant controls are not always in front of the other controls, and
the graphical controls (RoundRectangle, Oval, Rectangle and Line) are not always at the bottom.

- It is unsupported to open a response Window or MessageBox by triggering the Open event of a window, or by triggering the Constructor event of a control or user-defined event in the Open event of a window.

- 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.

Closing Windows

- Close functions

Refer to the Window Functions section of System Functions for the supported syntaxes for closing

the windows.

○ Triggering of events

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.

Events triggered for the window object follow the order Deactivate >> CloseQuery >> Close. In Web applications, only the sheets will have the behavior: “When the CloseQuery event returns 1, the window will not be closed.”

User operation in windows

APPEON

User Operation in Windows

Supported

The major user operations that are **supported**:

- Maximize (not for Main, Popup and Response windows)
- Restore
- Minimize (not for Main, Popup and Response windows)
- Single-click
- Double-click (except that double-click at the title of a Main/Popup/Response window cannot make the window maximized)
- Drag
- Close
- Switching between controls in the window with the Tab key
- Switch between windows by selecting a window from the window list in the menu (in fact, the window list does not display in Web applications)

Unsupported

The user operations that are **unsupported**:

- Scrolling a window using the PageUp/PageDown, upArrow/downArrow, or leftArrow/rightArrow keys is unsupported.
- Switch between windows by pressing the Ctrl + Tab key
- Switch between controls in the window by pressing the Enter/UpArrow/DownArrow/LeftArrow/RightArrow key
- When focus goes from control 1 to control 2 and then back to control 1 again, the previously selected text in control will be unselected although the focus is back. This is different behavior from

PowerBuilder.

- If the user double-clicks in a PowerBuilder window, the DoubleClicked event is triggered once and the Clicked event is triggered twice. If the user double-clicks in a deployed window, only the DoubleClicked event is triggered.

ArrayBounds object

This object, including its events, properties and functions, is not supported.

ClassDefinition object

This object, including its events, properties and functions, is not supported.

ConnectionInfo object

This object, including its events, properties and functions, is not supported.

ContextInformation object

This object, including its events, properties and functions, is not supported.

ContextKeyword object

This object, including its events, properties and functions, is not supported.

CPlusPlus object

This object, including its events, properties and functions, is not supported.

DynamicDescriptionArea object

This object, including its events, properties and functions, is not supported.

EnumerationDefinition object

This object, including its events, properties and functions, is not supported.

EnumerationItemDefinition object

APPEON

This object, including its events, properties and functions, is not supported.

Environment object

APPEON

This object, including its events, properties and functions, is not supported.

Error object

APPEON

This object, including its events, properties and functions, is not supported.

grAxis object

APPEON

This object, including its events, properties and functions, is not supported.

grDispAttr object

APPEON

This object, including its events, properties and functions, is not supported.

Inet object

APPEON

This object, including its events, properties and functions, is not supported.

InternetResult object

APPEON

This object, including its events, properties and functions, is not supported.

mailFileDescription object

APPEON

This object, including its events, properties and functions, is not supported.

mailMessage object

This object, including its events, properties and functions, is not supported.

mailSession object

This object, including its events, properties and functions, is not supported.

MenuCascade object

This object, including its events, properties and functions, is not supported.

OLEStorage object

This object, including its events, properties and functions, is not supported.

OLEStream object

This object, including its events, properties and functions, is not supported.

Pipeline object

This object, including its events, properties and functions, is not supported.

ProfileCall object

This object, including its events, properties and functions, is not supported.

ProfileClass object

This object, including its events, properties and functions, is not supported.

ProfileLine object

This object, including its events, properties and functions, is not supported.

ProfileRoutine object

APPEON

This object, including its events, properties and functions, is not supported.

Profiling object

APPEON

This object, including its events, properties and functions, is not supported.

ScriptDefinition object

APPEON

This object, including its events, properties and functions, is not supported.

SimpleTypeDefinition object

APPEON

This object, including its events, properties and functions, is not supported.

Timing object

APPEON

This object, including its events, properties and functions, is not supported.

TraceActivityNode object

APPEON

This object, including its events, properties and functions, is not supported.

TraceBeginEnd object

APPEON

This object, including its events, properties and functions, is not supported.

TraceError object

APPEON

This object, including its events, properties and functions, is not supported.

TraceESQL object

APPEON

This object, including its events, properties and functions, is not supported.

TraceFile object

APPEON

This object, including its events, properties and functions, is not supported.

TraceGarbageCollect object

APPEON

This object, including its events, properties and functions, is not supported.

TraceLine object

APPEON

This object, including its events, properties and functions, is not supported.

TraceObject object

APPEON

This object, including its events, properties and functions, is not supported.

TraceRoutine object

APPEON

This object, including its events, properties and functions, is not supported.

TraceTree object

APPEON

This object, including its events, properties and functions, is not supported.

TraceTreeError object

APPEON

This object, including its events, properties and functions, is not supported.

TraceTreeESQL object

APPEON

This object, including its events, properties and functions, is not supported.

TraceTreeGarbageCollect object

APPEON

This object, including its events, properties and functions, is not supported.

TraceTreeLine object

APPEON

This object, including its events, properties and functions, is not supported.

TraceTreeNode object

APPEON

This object, including its events, properties and functions, is not supported.

TraceTreeRoutine object

APPEON

This object, including its events, properties and functions, is not supported.

TraceTreeUser object

APPEON

This object, including its events, properties and functions, is not supported.

TraceUser object

APPEON

This object, including its events, properties and functions, is not supported.

Transport object

APPEON

This object, including its events, properties and functions, is not supported.

TypeDefinition object

APPEON

This object, including its events, properties and functions, is not supported.

VariableCardinalityDefinition object

APPEON

This object, including its events, properties and functions, is not supported.

VariableDefinition object

This object, including its events, properties and functions, is not supported.

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.

Object-oriented programming

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

- Inheritance

  It is unsupported to create a new control in a child object by copy 'n paste an existing control in the PowerBuilder painter, if the existing control is inherited from the parent object.

- Encapsulation
- Polymorphism

Language basics

Comments

Identifiers

Labels

Special ASCII Characters

Null values

Reserved words

Pronouns

Statement continuation & separation

Comments

APPEON

Supported

Common Comments:

Double-slash method: Code // Comment

A= B + C// Comment

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

A= /* comment */ B + C

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

Identifiers

APPEON

Supported

Rules for identifiers:

- Must start with a letter or an _ (underscore)
- Can be reserved words in JavaScript, apart from the identifier word "Object".
- Can have up to 40 characters but no spaces
- Can include any combination of letters, numbers, and the special character "_" (underscore).

Unsupported

- Identifiers cannot include any of the following special characters (please note that there is always a dash sign in a PowerBuilder default menu name. Before deploying the application with Appeon, you must clear all the dash signs from the menu names):

  - Dash

  # Number sign

  % Percent sign

  & Ampersand sign

  $ Dollar sign

  Double-byte characters, such as Chinese, Korean, Japanese, etc

- Application names cannot be the names of PowerBuilder control/object types. For example, an application cannot be named as CommandButton or CheckBox. In addition, Application object names can only include a combination of letters, numbers, and a dash.

- 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.

- Identifiers cannot be reserved words in Appeon: appeondatawindow, appeondatastore, appeonservice, AppeonGetAppeonUserName, AppeonGetBrowserVersion, AppeonGetClientID, AppeonGetClientIP, AppeonGetClientType, AppeonGetOSType.

- In case of duplicate objects, Appeon cannot guarantee that the instantiated object is the same as in PowerBuilder.
  In Appeon:
  1) If the duplicate objects are of the same type, the instantiated object will be in the bottom part of the PBL list of the Application Profile.

  2) If the duplicate objects are of different types, both of which are supported by Appeon, there is no fixed rule to determine which object will be instantiated. Because there are various scenarios, such as whether the objects are in the same PBL or not, the application is either incrementally deployed or fully deployed.

  3) If the duplicate objects are of different types, and one of the types is unsupported, the instantiated object will be the object that is supported.

Labels

Labels and GOTO statements are unsupported.

Special ASCII characters

APPEON

Supported

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

- Although new line and carriage return characters are supported, for deployed DataWindows (Grid, Freeform and Tabular), the use of carriage return and new line (~r~n) to make a new line in a column header are ignored in the converted Web application. The column headers in the DataWindows automatically wrap text if the text is wider than the column header itself.

Unsupported

1) Common ASCII character: Vertical tab (~v)
2) Any ASCII character: Decimal (~###)
3) In a Web application, a character that has an ASCII value greater than 127 cannot be saved into the database.
4) 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.
5) Appeon ignores the escape character ("~") specified in the PowerBuilder painter. In SQL statements, "~" is handled as escape character.

Null values

APPEON

Supported

NULL means undefined or unknown. It is not the same as an empty string or 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, in PowerBuilder, an arithmetic or relational operation involving a null value always returns null. In JavaScript, the return value is not always null, as shown in the following table:

Operation		Return Value	
Assuming SetNull(A), SetNull(B)		PowerBuilder	JavaScript
Arithmetic	A+1	Null	1
	A+B	Null	Null
	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"

Reserved words

Supported

Functionality associated with the following reserved words is supported:

and	call	case	catch	choose	close	commit	connect
constant	continue	create	cursor	declare	delete	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	return	rollback	select	step	subroutine	super	then
this	to	trigger	true	try	type	until	update
using	while	with	within				

- The reserved word PROCEDURE can only be used in DECLARE Procedure SQL statement
- The reserved word DESCRIBE is partially supported
- The reserved word HALT is supported.
  Note: there cannot be any code following the Halt statement.
  For example, the following PowerScript code is not supported:
  Halt
  Close(parent)

Unsupported

Functionality associated with the following reserved words is not supported:

enumerated	external	finally	goto	indirect	intrinsic	library	ref
rpcfunc	selectblob	system	systemread	systemwrite	throw	throws	updateblob

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.

Statement continuation & separation

APPEON

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

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

Unsupported

None.

Data types

APPEON

Standard data types

Any data type

Date type format

System object data types

Enumerated data types

[Forced conversion between data types]

Standard data types

APPEON

Supported

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

Decimal {4} and Dec {4} values are supported.
Values are assigned to Date, DateTime and Time variables by copy, not by reference in Appeon.

Unsupported

The Standard Data Type Blob is unsupported.

The UnsignedLong and UnsignedInteger data types are handled as Long and Integer. Therefore, the supported range for UnsignedLong is from -2,147,483,648 to +2,147,483,647. In addition, the supported range for UnsignedInteger is from -32768 to +32767.

In PowerScript, double and real values with 17 or more digits are automatically converted to scientific notation. However the rule in JavaScript is as follows, double and real values will not be automatically converted to scientific notation unless they are more than 22 digits.

If the decimal digits of a decimal data are all 0, the all-0 decimal digits will not be displayed on the Web. For example, 123.000 in PowerBuilder will display as 123 on the Web.

All numeric data types can have 19 digits in PowerBuilder. In the Web application, numeric data types can have 17 digits.

The Resize event can be triggered in Child or Sheet window scripts but not in the Main, MDI, MDIHelp, Popup and Response window scripts.

When you set the data type of a column as char with a specified length, the value of the column data type retrieved by Appeon is different from that in PowerBuilder. The value you get in Appeon is char. However, the value you get in PowerBuilder is char(n).

Any data type

APPEON

Any data type is fully supported, with the exception as follows:
It's unsupported to assign the return value of a String function to an Any variable that is not assigned before.

Date type format

The date type format can be any of the following:

MM-DD-YYYY	YYYY-MM-DD	MMM-DD-YYYY
MM/DD/YYYY	YYYY/MM/DD	MMM/DD/YYYY
MM-DD-YY	YYYY-MMM-DD	MMM-DD-YY
MM/DD/YY	YYYY/MMM/DD	MMM/DD/YY

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.

Enumerated data types

Supported

All the 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. It is unsupported to use the default values of enumerated data types.

Forced conversion between data types

APPEON

Supported

1. Forced conversion between string and char data type:

   - 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, and so on) can be assigned to char variables using string conversion, such as:
     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 about numeric data types is supported. The following is the order of highest to lowest precedence (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, with the following syntax:
     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 UnsignedInteger, both are promoted to UnsignedLong.

Unsupported

It is unsupported to convert a char array and use it as a string. The following forced conversions are unsupported:

1. Assigning strings to char arrays
2. Assigning char arrays to strings
3. Using char arrays in PowerScript functions, although in PowerBuilder, all PowerScript functions that take strings are also char arrays.
4. The forced conversion from double/real data to int/long data may arrive at different values in PowerScript and JavaScript (due to different rounding method used).
   For example:
   integer li_int, li_int_dec, li_int_dou, li_int_real
   double ldou_1
   real lreal_1
   li_int = 3/2 // li_int will be 1 in PowerScript and 2 in JavaScript
   ldou_1 = 3/2
   lreal_1 = 3/2
   li_int_dou = ldou_1 // li_int_dou will be 1 in PowerScript and 2 in JavaScript
   li_int_real = lreal_1 // li_int_real will be 1 in PowerScript and 2 in JavaScript
5. The forced conversion from string data to int/long data may arrive at different values in PowerScript and JavaScript (due to different rounding method used).
   For example: integer ("1.5") //The result will be 1 in PowerScript and 2 in JavaScript
6. If an int/long data is divided by another int/long data and the result is assigned to an integer/long, the result is rounding down in PowerScript and is rounding off in JavaScript.
   For example:
   integer int_1, int_2, int_3
   int_1 = 3
   int_2 = 2
   int_3 = 3/2 //int_3 will be 1 in PowerScript and 2 in JavaScript
7. When a real data is converted to a string, PowerScript will round the decimal digits while JavaScript will not. For example:
   real lr_num = 12345.567
   ls_num = string(lr_num) //ls_num will be “12346” in PowerScript and “12345.567” in JavaScript
8. In PowerBuilder, when Time data type is automatically converted into DataTime, the date is always 1900-01-01. Thus, DataWindow always gets a value of DataTime type: Retrieve (1900-01-01 xx:xx:xx). However, if connected to JDBC SQL Server, the default date is the current day. That is to say, DataWindow gets Retrieve(today xx:xx:xx) when Time data type is automatically converted into DateTime.

Declarations

Arrays

External functions

Variables and constants

APPEON

Supported

1) The variable scope can be global, local or instance.

2) Constants can be of any standard data type.

3) The following constant declaration syntax is 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()

4) The variable declaration syntax as follows is supported:

datatype { { size } } { { precision } } variablename { = value } {, variablename2 { = value2 } }

- The datatype can be any standard type (except Blob) and system object type.
- It is supported to declare multiple variables of the same type at one time.
  For example: integer li_a=5, li_b=10.

5) Initial values are supported:

- When a variable is declared, a default initial value is automatically assigned or an initial value can be specified in the declaration.
- Except for the initial values of enumerated data types and Any data type, the default initial values in PowerScript are supported.
- The initial values of enumerated data types are converted to null in JavaScript.

6) The order of searching for variables:

When there is an unqualified reference to a variable, PowerBuilder searches for variables in the following order: Local variable >> Shared variable >> Global variable >> Instance variable. As soon as PowerBuilder finds a variable with the specified name, it uses the variable's value.

However, JavaScript searches for variables in the order Local variable >> Instance variable >> Global variable.

For this reason, it is supported that a local variable has an identical name to an instance variable or global variable. However, it is unsupported when an instance variable has identical name to a global variable.

Unsupported

1) Specifying an expression as an initial value: different rules in PowerBuilder than in JavaScript

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, it is recommended to make the declaration of a variable and assigning it with the initial value in separate PowerScript statements. For example:
date d_date
d_date = Today( )

2) The Access keyword in the declaration syntax is not supported. For instance, variables and constants are all PUBLIC after conversion. This may cause problems if the instance variables are defined with PowerBuilder objects that are visual.

- access-right: PUBLIC, PROTECTED, PRIVATE
  The setting of access right in the syntax is always treated as PUBLIC.
- readaccess: PROTECTEDREAD, PRIVATEREAD
  The setting of readaccess in the syntax is always ignored.
- writeaccess: PROTECTEDWRITE, PRIVATEWRITE
  The setting of writeaccess in the syntax is always ignored.

3) Assigning a double or real value to an integer or long has different behavior in PowerBuilder than in JavaScript. If the decimal fraction in the value is 0.5, the value is rounded down in PowerBuilder but rounded up in JavaScript. For example,

```
Long ll_num
Double ld_num
ld_num = 2.5
ll_num = ld_num          //ll_num will be 2 in PowerBuilder, and will be 3 in JavaScript
```

4) The datatype in the variable declaration syntax cannot be Blob data type.

5) The variable scope shared is unsupported.

6) An instance variable having the same name as a global variable is unsupported.

7) It is unsupported to refer to a global variable by using the global scope operator (::) before the variable name;
(i.e. The syntax with "::globalname" is unsupported).

8) 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

9) 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.

Arrays

Declaration syntax

Initialization and assignment

Passing array as an argument

Complex arrays

Unsupported

Declaration syntax

APPEON

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. It is supported if the lower bound is a non-integer or a negative value. If the lower bound is not an integer, it will be rounded and the rounding method is rounding 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]

Initialization and assignment

APPEON

Initialization and Assignment

- 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 element is ""; the default value for numeric data type element is 0.

- Assigning values to part of an array and using the default values for the other part is supported.
  integer li_array [10]={1,2,3,4,5}
  //The values of li_array[6], …, li_array[10] are initialized to the default 0.

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

- It is supported to reinitialize arrays with the following steps:
  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. Example:
  for A = 1 to upperbound(ls_array)
  ls_array[A,1] = ……
  end for
  ls_array[integer(ltvi_item.data)] =123

Passing array as an argument

APPEON

Passing array as an argument

It is supported to pass a variable-size array or a fixed-size array as an argument. 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 array, multi-dimensional array or variable-size array:

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

Complex arrays

APPEON

Complex arrays

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

Unsupported

APPEON

Unsupported

In array declaration, it is unsupported to assign an expression to an element of the array. This is because in PowerBuilder, the expression's value is set to the element when the script is compiled. But in JavaScript, the expression's value is set to the element during execution.
For example, the following syntax is unsupported:

Date ld_date[] = {today(),relativedate(today(),1), 2002-12-31}

It is unsupported to convert a char array and use it as a string. The following forced conversions are unsupported:

1. Assigning strings to char arrays
2. Assigning char arrays to strings
3. Using char arrays in PowerScript functions. However, in PowerBuilder, all PowerScript functions that take strings also take char arrays.
4. The data type of an array variable cannot be Enumerated.

External Functions

APPEON

Declare external functions

Supported:

- Declare and call local external functions
- Declare and call global external functions

Data types of external function arguments can be the following:

- Unsigned char, char, unsigned short, short, unsigned long, long, float, string, and BOOL

Data types of external function return values can be the following:

- Unsigned short, short, unsigned long, and long

Operators & expressions

APPEON

Supported

- 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
- Expressions with “/”

  Expressions using the arithmetic operator “/” behave differently in PowerBuilder than on the Web:

1) If a and b both take a long/integer value, the expression “a/b” evaluates to a value rounded down in PowerBuilder, whereas on the Web it
evaluates to a value rounded up. For example:
Long ll_test
ll_test = 1/2
In PowerBuilder, ll_test evaluates to 0, but on the Web it evaluates to 1.

2) If a and b both take a long/integer value, if you form an arithmetic expression that contains “a/b“ (a and b both take the long data type),
the entire expression evaluates to a value rounded down in PowerBuilder, whereas on the Web, it evaluates to a value rounded up. For
example:
Long ll_test 1
ll_test 1 = 9+1/2
In PowerBuilder, ll_test1 evaluates to 9, but on the Web it evaluates to 10.

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

Unsupported

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

- Additions that include an unassigned char variable will arrive at an incorrect result on the Web. For example, the following script is unsupported:
Char A
String B = “abc"
B = A + B //Unsupported, because the char A is unassigned

Structures

APPEON

Definition and declaration of structures

Referring elements in a structure

Initialization and assignment

Passing structures as arguments

Complex structures

Unsupported

Definition and declaration of structures

APPEON

Definition and Declaration of Structures

It is supported to define a structure in the Structure painter or define a structure in an object painter:

- 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 variables.

- If the structure is defined in an object painter, the structure is 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.

It is supported to declare global instance or local instances for global structures. It is supported to declare instance and local instances for object structures. Several instances for a structure can be declared at one time.
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 has the identical name as a global structure, the object structure has higher priority than the global structure (same as in PowerBuilder).

Referring elements in a structure

APPEON

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

Initialization and assignment

APPEON

Initialization and assignment

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.
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.

Passing structures as arguments

Passing structures as arguments

It is supported to pass structures as arguments by value, by reference, or as read-only.

Complex structures

Complex structures

The following types of complex structures are supported:

- The structure has one or more of its structure variables being referred to another structure.
- One or more structure variables are autoinstantiated or non-autoinstantiated objects.
  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 are arrays that have no unsupported features.

Unsupported

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 the unsupported variables are not used. The structure is unsupported if the unsupported variables are used.
- If there are member variables of enumerated data type in the structure, the default value of enumerated variable is unsupported.
- The syntax structure.classname is unsupported. To work around it, please use the syntax classname (structure).

- The syntax structure.typeof is unsupported. To work around it, please use the syntax typeof (structure).

- In PowerBuilder, if a member of the structure is an autoinstantiated object, the Constructor event will be triggered the first time when the member is used. However, in the Web application, because the member is created during the declaration of the structure, the Constructor event will not be triggered.

User objects

APPEON

User objects

Autoinstantiated NVO

Nonautoinstantiated NVO

User objects

APPEON

Important Requirements

- For standard class user objects:
  - Standard class user objects can only inherit from the following non-visual system objects:
    - DataStore
    - Transaction Object
    - DynamicStagingArea
  - Non-visual standard class user objects must be defined in a PowerBuilder painter. They can be dynamically created (for example, using the CREATE statement).
  - 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:
  - Non-visual custom class user objects must be defined in PowerBuilder painter. They can be dynamically created (for example, using the CREATE statement).
  - The specification of EAServer Project property is unsupported.
  - 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 visual user objects and custom visual user objects:
  - Must be defined in PowerBuilder painter.
  - Cannot be dynamically created (e.g. using the CREATE statement).
  - Must contain only supported controls.
  - The SetFocus function is not supported for custom visual user objects, but is supported for standard visual user objects.
  - 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 object
- Custom visual user object
- Standard class user object
- Standard visual user object

Unsupported

- External visual user object
  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.

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

Autoinstantiated NVO

- Declaring an autoinstantiated NVO
  - 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.

  - If an instance variable contains an autoinstantiated NVO(b), when 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

  - 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

  - It is **unsupported** to assign a NonVisualObject object to an autoinstantiated NVO or assign an autoinstantiated NVO to a NonVisualObject object (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:
  - Instance variable. Instance variable can be an autoinstantiated NVO, an object, or have the same name as a window instance variable
  - System function or user defined function or object function
  - Constructor event, or object event or user-defined event
  - It is unsupported to use Destructor event in the definition of an autoinstantiated NVO.

- Autoinstantiated NVO array

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

    ```
    n_cst_string lnv_string[10]
    ```

  - If the autoinstantiated NVO array is variable-size array, when the array is declared, the NVO instances are not created. 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. 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

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

Nonautoinstantiated NVO

- Declaring a non-autoinstantiated NVO

  - Declaring a non-autoinstantiated user object does not create the object instance until there is the 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
    ```

  - When the object instance is created, the instance variables of the NVO are instantiated, and the Constructor event is triggered for each instance variable.
  - It is supported to instantiate an ancestor variable with an instance of one of its descendants. 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
  - 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
    ```

  - Different from an autoinstantiated NVO, it is **supported** to assign a NonVisualObject object to a non-autoinstantiated NVO or assign a non-autoinstantiated NVO to a NonVisualObject object. 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 NVO
  - The following can be included in the definition of a non-autoinstantiated NVO:
    - Instance variable. Instance variable can be an autoinstantiated NVO, an object, or have the same name as a window instance variable
    - System function or user defined function or object function
    - Constructor event, Destructor event or object event or user-defined event. 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 array. When the array is declared, the instance of the object is not created. A non-autoinstantiated NVO will only be created when there is the CREATE statement for the NVO.

- Non-autoinstantiated NVO as a structure member
  - It is supported to define a non-autoinstantiated NVO in a structure. Example:

    ```
    global type str_model from structure

    string s_emplid

    Date ld_inputday

    n_cst_base lnv_base
    ```

end type

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

Calling functions and events

APPEON

Syntax for calling functions and events

Triggering & Posting

Static & dynamic calls

Overloading, overriding, and extending functions and events

Arguments and return values

Syntax for calling functions and events

APPEON

Supported

The syntax used to call all PowerBuilder functions and events:

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

- The specification of calltype (DYNAMIC/STATIC) in the syntax will be ignored.
- If objectname is dynamic, it has to be determined during runtime (control[] for example, there are problems).

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.

The following features apply to the partly supported POST value of the *when* argument:

Supported
-Post Close(window)
-Post open(window)
Note that the *window* argument is an existing window object, not a variable.

Unsupported
- Post Open(windowvariable)
Note that the *windowvariable* argument is a window variable. The following example is not supported:

```
w_main gw_test
……
post open(gw_test)
```

- Post user_function()

Note that the user_function function is a user-defined global function.

- POST and TRIGGER cannot be used at the same time. For example, the following syntax is unsupported: parent post triggerevent (“ue_refresh?

For more information on calling DLL, refer to Program access techniques.

Unsupported

- A window/object function having the same name as a global function is unsupported.
- It is unsupported to refer to a global function by using the global scope operator (::) before the function name.
- It is unsupported to pass a control name as an argument in a PowerBuilder system function. For example:

```
cb_1.classname()        //Supported
classname(cb_1)         //Unsupported
dw_1.insertrow(0)        //Supported
InsertRow(dw_1,0)        //Unsupported
```

- PowerBuilder searches for events in the object's ancestor hierarchy until it gets to the top ancestor or finds an event that overrides its ancestor. However, Appeon searches for events from the object to the window.

Triggering & Posting

APPEON

Supported

- Triggering for functions
- Triggering for events
- Posting for functions
- Posting for events

Unsupported

- For application and message objects, triggering and posting for functions and events are unsupported.
- Triggering and posting global functions are unsupported.
- Triggering and posting user-defined global functions are unsupported.
- Triggering and posting system functions are unsupported.

Static & dynamic calls

APPEON

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.
- It is unsupported to dynamically call a function that has an argument passed by reference.
- If a function is dynamically called, its return value cannot be passed as an argument of another function.
- A menu object making dynamic calls to a function of its parent window is not supported.
  Syntax: parentwindow.dynamic *functionname*
  Example:
  string ls_info = 'sign value in menu';
  parentwindow.dynamic function wf_argupassgeneral (ls_info, 3)
- It is not recommended to assign the same name to a function and event in the same object. If a function has the same name as an event, neither the function nor the event can be dynamically called.

Overloading, overriding, and extending functions and events

APPEON

Supported

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

Unsupported

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

  wf_getname(dw_1.object.s_id[1])
- It is unsupported to use the local variable AncestorReturnValue in an event of a descendent object, unless the event of the descendent 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.

  It is **unsupported** to call function g() in object B. In other words, it is unsupported for a child object to call an inherited function from the parent object, where in the parent object, the function calls another function, which has an overloading function in the child object.

Arguments and return values

APPEON

Supported

The three ways used to pass arguments to functions and events:

- By value
- By reference
- Read-only

It is supported to pass an array constant as the argument of event/function. The array constant will be handled correctly (although the array lower bound begins with 0 in JavaScript and with 1 in PowerBuilder).

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 that have return values
- Return values for system events
- User-defined events that have return values
- The function has parameters of standard datatype passed by reference and the return value is used as a condition in RETURN, IF ... THEN, CHOOSE …CASE, or DO ... LOOP statements.

  For example, the following code is supported:

  ```
  :
  Return of_func( abc)       // where abc is a ref. argument
  :
  If of_func( abc ) = True Then       // where abc is a ref argument
  //
  :
  ```

- 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 …CASE, or DO ... LOOP statements.
- The return value of one function is used as the parameter of another function, for example, func1

(func2()).

- Each event or function can have a maximum of 20 arguments. If the number of arguments exceeds 20, the arguments after the 20th argument will be invalid.

Unsupported

Using the return values of functions and events in the following situation is not supported:

- Using cascaded calling and return values, where the return value of the function or event becomes the object for the following call, for example: func1returnsobject( ).func2returnsobject ( ).func3returnsanything( )

Using arguments in the following situations is not supported:

- Both function A and function B have an argument passed by reference. For example: f(int a, ref int b); g (ref int a). It is unsupported to call 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.
  With the above example, the following syntax is unsupported: 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: f(a,a); //f(ref int a, ref int b);

- It is unsupported to pass a property dot notation as the function argument, if the property refers to an object. For example:
  lvn_security.Of_setmenuright(this.MENUID)

  To work around this unsupported feature, use the following syntax:
  menu m_1 = this.MENUID
  lvn_security.Of_setmenuright(m_1)

It is unsupported to pass an argument that is an object property by reference. For 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)    //Unsupported
```

Document Interface

Important Requirements

The following are limitations for using MDI and SDI:

- When a sheet window is open 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. In PowerBuilder, you can choose the sheet toolbar to be added after the MDI toolbar, on the left/right of the screen, to be floating, and so on.)
- SDI can have only one menu and one toolbar.
- MDIClient object cannot be dynamically created (e.g. using the CREATE statement); it must be defined in the PowerBuilder painter.

Supported

- Both MDI and SDI 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 open within MDI, the menu of the sheet window will replace the menu of the MDI window. (This is called menu switch, exactly like PowerBuilder).
- Appending the names of open sheets (Window list) to a menu item is supported.

Unsupported

- Multiple MDI windows are unsupported. An application cannot have more than one MDI window.
- Multiple MDI frames are unsupported.
- MDIClient object cannot be dynamically created (e.g. using the CREATE statement); it must be defined in the PowerBuilder painter.

PowerScript statements

Supported

The following table shows the supported PowerScript statements with examples:

Statement	Examples
Assignment	a = b + 2 Notes: 1) It is unsupported to assign an expression to an autoinstantiated object variable. 2) It is unsupported to assign an expression to a Date, Time, and DateTime variable. 3) 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 descendent 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 Note: 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	Supported: (1) Dynamically creating the following non-visual system objects: DataStore, DynamicStagingArea, and Transaction Object. Example: DataStore lds_main lds_main = create DataStore lds_main.dataobject="d_salary" (2) Dynamically creating non-visual user objects. Example: n_cst_string lnv_string lnv_string = create n_cst_string (3) Dynamically choosing the object type (however, the parameter objecttypestring cannot be a variable). Example: pfc_n_cst_dwsrv lnv_dwsrv lnv_dwsrv = create using “n_cst_dwsrv" Unsupported:

(1) Dynamically creating visual controls, such as CommandButton.
Example:
commandbutton lcb_1
lcb_1 = create using "cb_2"
(2) Dynamically creating visual user objects.
Example:
UserObject luo_1
luo_1 = create using "u_dw" //u_dw is a visual user object
(3) Create application
Example: Application lapp_1
lapp_1 = create using "visual_controls_demo"
//visual_controls_demo is an application.

DESTROY

DESTROY DBTrans
Supported:
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
Unsupported:
The Destroy statement in visual controls and visual user objects is unsupported.
Example:
commandbutton lcb_1
lcb_1 = create using "cb_2"
Destroy lcb_1

Note: DESTROY statement is unnecessary for a Web application since an object variable will be automatically destroyed when the Web application exits.

DO…LOOP

Four formats of Do…Loop:
Do…Until
Do…While
Loop…Until
Loop…While

```
DO UNTIL a > 15
a = (a + 1) * b
LOOP

Integer a = 1, b = 1
DO WHILE a <= 15
    a = (a + 1) * b
LOOP

Integer a = 1, b = 1
DO
    a = (a + 1) * b
LOOP UNTIL a > 15

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
```

Note: there cannot be any code following the Halt statement.
The reserved word HALT is supported.

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

- Modification to both Create and Destroy statements is unsupported.

Database server and data types

APPEON

Supported

The database servers and data types supported are shown in the table below:

Database Server	Supported Data Types
MS SQL Server 7 or 2000	bit char datetime decimal float

	int nchar numeric nvarchar real text varchar
Sybase ASE/ASA Server	char datetime decimal double float int nchar nvarchar numeric real smalldatetime smallint smallmoney tinyint varchar text timestamp
Oracle Server	char date float long nchar number nvarchar2 varchar2 data
IBM DB2 Server	int real double decimal char varchar date time timestamp

- Bit data type 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).
- Global/local transaction object and global/local transaction user object are supported.
- Stored procedure grouping is supported.
- Outer join SQL syntax is supported for Oracle Server.

Unsupported

- The database servers and data types other than those listed in the above table are unsupported.
- If a DataWindow column is TimeStamp data type, no data will be retrieved into the column.
- You can only use the default value for user_quoted_identifier property (Using a value that is not default is unsupported) for MS SQL Server 7 or 2000.
- The char or varchar type string cannot be zero length for Sybase ASE/ASA Server.
- If a Date value is specified in an embedded SQL statement, the Date value cannot contain full or abbreviated month name (for example, “April" or “APR" or yy (for example, 97").
- For Oracle Server, if the length of a char string retrieved from a database is shorter than the full length of a data field, the string will be padded with trailing blanks. However, in a Web application, the string will be trimmed.

Operators

APPEON

Supported

The following are the SQL operators that can be used:

=	<	>	<=	>=	<>	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

The operator ‘^' is unsupported.

Transaction management statements

APPEON

Supported

Statement	Examples
COMMIT	Commit
CONNECT	Connect
DISCONNECT	Disconnect
RollBack	Rollback
ErrorHanding	SQLCODE

SQLNRows
SQLERRTEXT

Unsupported

None.

Non-cursor statements

Supported

The following table lists the supported information of the several non-cursor statements:

Type	Supported	Details
SELECT	Retrieval list	• Column names or column ID or column aliases Example 1: Select s_emplid, s_emplname //Column names Into :ls_emplid, :ls_emplname From Employee; Example 2: select e. emp_lname //Table aliases into :ls_emplname from employee e using sqlca; Example 3: Select s_emplid + s_emplname as emplinfo //Column (expression) aliases Into :ls_emplinfo From Employee; select "employee"."emp_fname"+emp_lname into :ls_emplid from employee using sqlca; • Column names as characters (for example, "empid") or in dot notation (for example, emp.empid) • Expressions: functions, sub queries, arithmetic operators or any combination of columns, constants and expressions are supported. Example 1 (Concatenation of strings): Select s_emplid + s_emplname as emplinfo Into :ls_emplinfo From Employee; Example 2 (Function expression): Select substring(s_emplid + s_emplname,1,3) as emplinfo,getdate() Into :ls_emplinfo, :ldt_sysdate From Employee; Example 3 (Operation expression): Select f_salary / 12 as f_persalary Into :lf_salary From salary;

	• Asterisk that represents all columns in one table Example: select * into :ls_emplid,:ls_date,:ld_num from bonus using sqlca;
	• The For Update syntax is not supported.
FROM clause	• A single table (view) or multiple tables (views) Example: Select Employee.s_emplid , Employee.s_emplname, viewbonus.f_bonus Into :ls_emplid, :ls_emplname, :lf_bonus From Employee, viewbonus Where Employee.s_emplid = viewbonus.s_emplid;
	• Table names or table aliases Example: Select a.s_emplid , a.s_emplname, b.f_bonus Into :ls_emplid, :ls_emplname, :lf_bonus From Employee a, viewbonus b Where a.s_emplid = b.s_emplid;
	• No lock or lock of tables
WHERE clause	• retrieval parameters • Standard comparison operators (=, >, <, <>, >=, <=) • Standard logical operators (NOT, AND, OR) • Special operators (UNION, BETWEEN, IN, LIKE, IS NULL) • Join conditions • Special characters (‘&’,’~’,’[]’, ‘!~~!’, ‘```’, ‘!~@~~~~!’, etc.)
HAVING clause	*Supported*
GROUP BY clause	*Supported*
COMPUTED clause	*Supported*
Variables list	• Variables select "employee"."emp_fname"+emp_lname into: ls_emplid from employee using sqlca;
	• Control property select "employee"."emp_fname"+emp_lname into: sle_1.Text from employee using sqlca;
Example	SELECT f1, f2, ..., fn into :v1, :v2, ..., :vn

		FROM table WHERE w1 = :p1 and w2 = :p2.prop and/or ... and/or wn = :pn
INSERT	INSERT INTO clause	• Table (view) name • Column list
	VALUES clause	• All supported data types in allowed scope • Space, special characters (‘<>‘, ‘!’, ‘@’, ‘#’, ‘$’, ‘%’, ‘^’, ‘&’, ‘*’) • Initial value, a single records, multiple records (up to 500)
	Validation	• Validates the data type • Checks whether there is overflow
	Example	INSERT INTO table VALUES(“v1”, 12.3, :p1, :p2, …, :pn)
UPDATE	SET clause	• Updates a single record or multiple records (up to 200) • Special characters (‘<>‘, ‘!’, ‘@’, ‘#’, ‘$’, ‘%’, ‘^’, ‘&’, ‘*’)
	UpdateWhere	• 0 - Key columns only • 1 - Key columns and all updateable columns • 2 - Key and modified columns
	UpdateKeyInPlace	• Yes - Use the Update statement when the key is changed • No - Use the Delete and Insert statement when the key is changed
	Example	UPDATE table SET f1 = :p1, f2 = :p2 WHERE w1 = :p3 and/or … wn = :pn
DELETE	DELETE FROM clause	• Table (view) name • Deletes a single record or multiple records (up to 500)
	WHERE clause	• 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

- Select statements cannot contain outer join syntax.
- In PowerBuilder, when a NULL value is added to the value of a column, the result of the expression is the value of the column. However, in the Web application, the expression result is NULL.
- 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).

Cursor statements

APPEON

Supported

The following statements for retrieving and updating cursors are supported:

CLOSE	DECLARE	DELETE	FETCH
FETCH FIRST	FETCH LAST	FETCH NEXT	FETCH PRIOR
OPEN	UPDATE		

Local cursors are supported.

Unsupported

- Global and instance cursors are unsupported.
- It is unsupported if a cursor statement contains some constant as its output parameter.
- 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, it is unsupported to modify (insert, delete, update) the table during the cursor open – close period. 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;
```

- It is unsupported to place the cursor declaration syntax in a statement block that may not be executed at runtime. 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
OPEN cur_empl;
FETCH cur_empl INTO :ls_emplid, :ls_emplname;
……
```

With the above syntax, if the li_length is not 10, in the Web application, the cursor declaration syntax cannot be read, and errors occur.

Database stored procedures

APPEON

Supported

The following statements for retrieving and updating stored procedures are supported:

Statement	Examples
CLOSE	CLOSE lproc_1;
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, ...;

- Input & output parameters are supported.
- Return value for stored procedure is supported.

Oracle stored procedures

Supported:

- In & Out & InOut parameters are supported.
- Constants used as stored procedure are supported.
- A stored procedure can have multiple result sets.
- It is supported if the stored procedure placed inside a package body. For example, the following syntax is supported:

```
CREATE OR REPLACE PACKAGE BODY KANIANJ.PKG_APPEON_TEST IS
PROCEDURE SP_SEARCH_BY_LOAN_NUM
?//Syntax in the procedure
END PKG_APPEON_TEST;
```

- Stored procedures (dynamic and static) only support arguments that contain only one variable or one variable with constants.

Unsupported:

- It is unsupported to use an expression as parameter for calling to an Oracle stored procedure.
- It is unsupported for parameters in Oracle stored procedure to use default values.
- It is unsupported to overload Oracle stored procedures.

Stored procedures of other database types

Supported:

For ASE stored procedure used as DataWindow data source:

- The AutoCommit property must be set to TRUE and the chain is off.

If the database server is **IBM DB2 database server**:

- RPC calling to an SP is unsupported. This is because Appeon does not support RPC.
- IBM DB2 database stored procedures can contain one or more parameters.

Stored procedures (dynamic and static) only support arguments that contain only one variable or one variable with constants.

Unsupported:

- It is unsupported to place the stored procedure declaration syntax in a statement block that may not be executed at runtime. 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_lenth = 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;
……
```

With the above syntax, if the li_length is not 10, in the Web application, the cursor declaration syntax cannot be read, and errors occur.

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*/
  ```

  Note: 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 100 in Appeon. If the number of fetched row(s) is more than 1, the SQLCODE in the transaction object is 0 in PowerBuilder while -1 in Appeon.

- 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 mp_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.

Unsupported

- Dynamic SQL Format 4: executing a SQL statement that produces a result set in which the number of input parameters, or the number of result set columns, or both, are unknown at compile time.

Note: the Transaction object SQLDA is not supported, since it is only used with Dynamic SQL Format 4.

System functions

APPEON

Supported

The following are the categories with some or all of their functions supported:

Array Functions	Data Type Checking and Conversion Functions
Date, Day, and Time Functions	International Functions
Miscellaneous Functions	Numeric Functions
Registry Functions	String Functions
System and Environment Functions	Timing Functions
Window Functions	

In addition, Appeon supports the function HyperLinkToURL as a system function (the PowerBuilder object Inet is unsupported). Therefore, the following syntax is partly supported:
servicereference.HyperlinkToURL (url)

The object name servicereference will be ignored, while the function HyperLineToURL will be converted in the same way as any other supported system functions.

Syntax Support Information:
If the syntax is informal and has not been introduced in the PowerBuilder Help, it is always unsupported. This applies to all functions, not just the ParentWindow function used in the following example:

```
Supported syntax:
Win1 w_parent
w_parent = child_1.ParentWindow()

Unsupported syntax:
w_parent = ParentWindow(child_1)
```

Unsupported

- The following syntax structure is unsupported:

  1. To get/set the property of an object that is the return value of a function.

     Syntax: function.*property*

     For example: ParentWindow( ).Enabled = TRUE

  2. 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( ))

- The following are the unsupported system function categories that do not contain any supported individual functions:

Blob functions	Class Definition functions	DDE functions
DDE Server Functions	File functions	Garbage Collection functions
Help functions	Library functions	Print and Printer functions
Shared Object functions (SharedObject functions)	Tracing functions	

Array functions

APPEON

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[] li_upper = UpperBound(b,2) li_upper = UpperBound(b,1) li_upper = UpperBound(a) li_upper = UpperBound(a,1) li_upper = UpperBound(c)

Data Type Checking and Conversion functions

APPEON

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("~n~~djfksdl") When char(0) is used in an expression, it returns “” (empty string) in PowerBuilder. In the Web application, char(0) will be regarded as the string ending sign, and the characters following the char(0) will be ignored. For example, ls_value = "abc" + char(0) + "xyz". In PowerBuilder, ls_value returns “abcxyz” and “abc” in the Web application.

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 (“”) in the Web application.

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 Bolvalue 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 returns 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
		Note: if the timevalue argument contains “am” or “pm”, the execution result of the IsTime is different on the Web from in PowerBuilder. For example, if the string is "9:23:21 am", the function returns True in PowerBuilder. However, it returns FALSE in the Web application.

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) The string argument cannot include "am" or "pm".
		2) String(data, format) cannot be used in DataWindow property expression.
		3) In a Web application, if the format argument includes more than one mask, the String function returns different value from PowerBuilder. For example, if the syntax is String(Date("2002-03-26"),"yy-mm-dd,yyyy-mm-dd"), in PowerBuilder, the String function returns "02-03-06,02-03-06" and "02-03-06,YYYY-0-DD" in the Web application.
		4) 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 format such as “!!!”, “aaa”.

Time	Supported	lt_test = time("23:00") Notes: 1) It is unsupported to convert a string containing "am" or "pm" to a time data type. 2) After conversion, the microsecond portion of the time will be omitted.

Date, Day and Time functions

APPEON

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()) Note: the time argument cannot include "am" or "pm".
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) 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())

International functions

APPEON

Function	Support Level	Coding Examples
Reverse	Supported	ls_return = Reverse(ls_test)
IsAllArabic	Unsupported	
IsAllHebrew	Unsupported	
IsAnyArabic	Unsupported	
IsAnyHebrew	Unsupported	
IsArabic	Unsupported	
IsArabicAndNumbers	Unsupported	
IsHebrew	Unsupported	

IsHebrewAndNumbers	Unsupported
FromAnsi	Unsupported
FromUnicode	Unsupported
ToAnsi	Unsupported
ToUnicode	Unsupported

Miscellaneous functions

APPEON

Function	Support Level	Coding Examples
ClassName	Supported	ls_classname = ClassName(li_array) If the argument is a numeric data type, the function returns “numeric? If the argument is an array data type, the function returns “array? If the argument is an enumerated data type, the function returns “string?
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) Note: it is unsupported to pass a structure variable as argument in the IsValid function.
KeyDown	Supported	KeyDown (keycode)
MessageBox	Supported	li_return = MessageBox('Title1','Text1',Information!,OK!,1) Note: be sure to use NewLine character in the text if the text is long. Otherwise, the text will be cut off and display the latter part only.
RGB	Supported	ll_color = RGB(255, 255, 255) The RGB value scope supported: 0~16777215 The custom color scope supported: 16777216~33554431 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. Note: a color value in a non-PowerBuilder 8.0 may map to a different color from the same value in PowerBuilder 8.0. If you prefer the original choice of color, when upgrading the application to a PowerBuilder 8.0 application, be sure to verify that correct colors are selected.
SetNull	Supported	SetNull(ls_test) Note: a color value in a non-PowerBuilder 8.0 may map to a

		different color from the same value in PowerBuilder 8.0.
SetPointer	Supported	SetPointer(Cross!) Note: in PowerBuilder, the pointer automatically changes back to an arrow after the function is executed. However, on the Web, the pointer will not change back until the function SetPointer(Arrow!) is executed.
Beep	Unsupported	
ChooseColor	Unsupported	
DebugBreak	Unsupported	
DraggedObject	Unsupported	
GetFolder	Unsupported	
PixelsToUnits	Unsupported	
PopulateError	Unsupported	
SignalError	Unsupported	
Sleep	Unsupported	
UnitsToPixels	Unsupported	

Numeric functions

APPEON

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) 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) Note: LogTen(0) is unsupported.
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)
Round	Supported	ldec_ return = Round(6655.16973,3) ldec_ return = Round(af_num,b_num)
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) ldeci_ return = Truncate(af_num)
Randomize	Unsupported	

Registry functions

Function	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!, RegMultiString!, but cannot be RegBinary!, RegLink!.
RegistrySet	Supported	RegistrySet("HKEY_USERS\MyApp.Settings\Fonts", "NameOfEntryNum", RegULong!, ul_num) Notes: 1) The ValueType argument can be RegString!, RegExpandString!, ReguLong!, ReguLongBigEndian!, RegMultiString!, but cannot be RegBinary!, RegLink!. 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)
RegistryKeys	Unsupported	

When setting the value for a key and value name in the system registry, 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) 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).

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.

String functions

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())
AscA	Supported	AscA(string) li_test = AscA("A") In PowerBuilder, AscA returns the ASCII value of the first character in a string. However, on the Web, it returns the Unicode code point value in a string in the same way as the Asc function.
Char	Supported	ls_test1 = Char(65) ls_test1 = Char("~n~~djfksdl")
CharA	Supported	CharA(n) ls_S = CharA(42) In PowerBuilder, CharA returns the first ANSI character of a string. However, on the Web, it returns the first Unicode character of a string in the same way as the Char function.
Fill	Supported	ls_fill = Fill('-+',10) ls_fill = Fill('hello', 6)
FillA	Supported	FillA(chars, n) ls_fill = FillA("*", 35) In PowerBuilder, FillA returns a string *n* bytes long filled with the characters in the argument *chars*. However, on the Web, it returns a string *n* characters long filled with the characters in the argument *chars* in the same way as the Fill function.
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	LeftA(string, n) ls_return = LeftA("BABE RUTH", 4) In PowerBuilder, LeftA returns the leftmost *n* characters in the source *string*. However, on the Web, it returns the characters for the leftmost *n* bytes in the source *string* in the same way as the Left function.
LeftTrim	Supported	ls_return = string(LeftTrim(' fdfsdf')) ls_return = string(LeftTrim(' 34ttrtertgre'))
Len	Supported	ls_return = string(Len('gfgdfgsdrgdfg'))
LenA	Supported	LenA(stringorblob) ls_return = LenA("gfgfgfgf") In PowerBuilder, LenA returns the length of a string in number of characters. However, on the Web, it returns the length of a string in number of bytes. In addition, if the argument is a blob, no conversion takes place in PowerBuilder or on the Web.

Lower	Supported	ls_return = Lower(You ARE Welcome!)
Match	Supported	ls_return = string(Match('ABBBC,ABB*C'))
Mid	Supported	ls_return = string(Mid('Appeon Appeon',5,2))
MidA	Supported	MidA(string, start{,length}) ls_return = MidA("BABE RUTH", 5, 5) In PowerBuilder, MidA returns characters specified by the number of bytes. However, on the Web, it returns characters specified by the number of characters searched in a source *string*; beginning with the character specified in the *star* argument. On the Web, MidA works same way as the Mid function.
Pos	Supported	ls_return = string(Pos('Appeon Appeon','on')) ls_return = string(Pos('Appeon Appeon','peon',2))
PosA	Supported	PosA(string1, string2{,start}) ls_return = PosA("BABE RUTH", "RU") In PowerBuilder, PosA returns a long following the position, in bytes, specified by *start*. However, on the Web, it returns a long whose value is the starting position of the first occurrence of *string2* in *string1* following the position, in characters, specified by *start*. On the Web, PosA works the same as the Pos function.
Replace	Supported	ls_return = string(Replace('BABE RUTH', 1, 4, 'BABY'))
ReplaceA	Supported	ReplaceA(string1,start, n,string2) ls_return = ReplaceA("ABCDEF", 3, 2, "ZZZZ") In PowerBuilder, ReplaceA replaces a string by number of bytes, whereas on the Web, it replaces a string by number of characters. ReplaceA also specifies the starting position of the string to be replaced by number of bytes, whereas on the Web, it specifies the starting position by number of characters. On the Web, ReplaceA works the same way as the Replace function.
Reverse	Supported	ls_return = string(Reverse('vDl%Qv'))
Right	Supported	ls_return = string(Right('Davis', 4))
RightA	Supported	RightA(string, n) ls_return = Right("BABE RUTH", 4) In PowerBuilder, RightA returns the specified number of bytes from the end of a string. However, on the Web, it returns a specified number of characters from the end of a string. On the Web, RightA works the same way as the Right function.
RightTrim	Supported	ls_return = string(RightTrim('fsdjd')) ls_return = string(RightTrim(' fsdfjdlfsd'))
Space	Supported	ls_return = string(Space(14))
Trim	Supported	ls_return = string(Trim('uifusd'))
WordCap	Supported	ls_return = string(WordCap('how do you do?'))
Upper	Supported	ls_return = string(Upper('how do you do?'))

System and Environment functions

Function	Support Level	Coding Examples
CommandParm	Supported	String ls_command_line ls_command_line = CommandParm()
GetFocus	Supported	Lobj_control = GetFocus()
GetApplication	Supported	application app app = GetApplication()
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") It is supported to run executable programs. If an application runs an executable program, you must specify that the application needs to run an executable program by checking the "Call to OLE objects, DLLs, or run executable programs on the client" option in the Appeon Developer Configuration.
SetProfileString	Supported	SetProfileString ("C:\PROFILE.INI", "Position", "Title", "MGR")
Clipboard	Unsupported	
GetEnvironment	Unsupported	
Handle	Unsupported	
Post	Unsupported	
Restart	Unsupported	
Send	Unsupported	
SignalError	Unsupported	
Yield	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).

- Note: files that have the same names cannot be used in Profile functions.

Timing functions

APPEON

Function	Support Level	Coding Examples
Timer	Supported	Timer(10, w_main)
CPU	Unsupported	
Idle	Unsupported	
Start	Unsupported	

Window functions

APPEON

Function	Support Level	Coding Examples
Close	Supported	close(w_commandbutton) close(parent)
CloseWithReturn	Supported	CloseWithReturn(parent,"return value") Note: 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.
Open	Supported	Open(w_main, parent) Note: the argument of the function cannot be a Child window object.
OpenSheet	Supported	OpenSheet(w_main,w_parent) OpenSheet(lw_main , "w_main", w_frame , 2, Cascaded!) Note that 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!) The following syntax is supported: OpenSheet (sheetrefvar {, windowtype }, mdiframe {, position {, arrangeopen } }) Appeon supports appending a window list to a menu item on the Web. The ArrangeOpen argument can be Cascaded!, Layerd!, Original!. In PowerBuilder, if position is 0 or greater than the number of items on the menu bar, PowerBuilder appends the name of

		the sheet to the next-to-last menu item in the menu bar. However, Appeon does not support the position argument greater than the number of items on the menu bar. In that case, the sheet name is not appended to any menu item.
OpenSheetWithParm	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) Note: the window type in the function cannot be MDI or MDIHelp.

Note that Open, OpenSheet, OpenWithParm and OpenSheetWithParm support *windowtype* string as parameter (see PowerBuilder syntax).

Blob functions

APPEON

This system function type is not supported.

Class Definition functions

APPEON

This system function type is not supported.

DDE functions

APPEON

This system function type is not supported.

DDE Server functions

APPEON

This system function type is not supported.

File functions

APPEON

This system function type is not supported.

Garbage Collection functions

APPEON

This system function type is not supported.

Help functions

APPEON

This system function type is not supported.

Library functions

APPEON

This system function type is not supported.

Print functions

APPEON

This system function type is not supported.

Printer functions

APPEON

This system function type is not supported.

Shared Object (SharedObject) functions

APPEON

This system function type is not supported.

Tracing functions

APPEON

This system function type is not supported.

User functions

APPEON

Supported

- Object Functions
- Global Functions

Unsupported

- Global and local External Functions
- Global and local Remote Procedure Calls (RPC)
- An object function matching the name and argument list of a global function is unsupported.
    - For example, an application contains a global function named f_effect and an object function located in the window object w_test that is also named f_effect. If the window object calls the function f_effect with the object name omitted, different results will occur: in PowerBuilder the global function will be called, but in JavaScript (e.g. in the Web application) the object function will be called.

Event types

APPEON

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.
- It is unsupported to use the local variable AncestorReturnValue in an event of a descendent object, unless the event of the descendent object is an extended event from the ancestor object, or the AncestorReturnValue is generated in Call Super statement.
- If a system event involves UI operation, the subsequent event will not be triggered until the UI operation is done.

System messages

Supported

The following table lists the supported system messages. The Notes in the table indicate that the event and event ID are partly supported:

Object/Control	Event ID	System Event using the ID
All	pbm_constructor	Constructor
	pbm_destructor	Destructor
	pbm_keydown	Key
Controls and windows, except RichTextEdit	pbm_rbuttondown	RButtonDown
Window	pbm_activate	Activate
	pbm_close	Close
	pbm_closequery	CloseQuery Note: the return value 1 only can prevent sheet window from closing.
	pbm_deactivate	Deactivate
	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_bnkillfocus	LoseFocus
	pbm_bnsetfocus	GetFocus
DropDownListBox/ DropDownPictureListBox	pbm_cbnkillfocus	LoseFocus
	pbm_cbnmodified	Modified

	pbm_cbnselchange	SelectionChanged
	pbm_cbnsetfocus	GetFocus
DataWindow, DataStore	pbm_dwnbuttonclicked	ButtonClicked
	pbm_dwnbuttonclicking	ButtonClicking
	pbm_dwndberror	DBError
	pbm_dwnhscroll	ScrollHorizontal
	pbm_dwnchanging	EditChanged Note: in PowerBuilder, when the user selects all the text of an edit column and types another value, the EditChanged event will be triggered twice. The first time is when the text of the column is being deleted, whereas the second time is when a new value is being input. However, in the Web application, the EditChanged event will be only triggered once for changing the text value.
	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_enkillfocus	LoseFocus
	pbm_enmodified	Modified
	pbm_ensetfocus	GetFocus
ListBox, PictureListBox	pbm_lbndblclk	DoubleClicked
	pbm_lbnkillfocus	LoseFocus

	pbm_lbnselchange	SelectionChanged
	pbm_lbnsetfocus	GetFocus
ListView	pbm_lvnclicked	Clicked
	pbm_lvncolumnclick	ColumnClick
	pbm_lvndeleteallitems	DeleteAllItems
	pbm_lvndeleteitem	DeleteItem
	pbm_lvndoubleclicked	DoubleClicked
	pbm_lvnkeydown	Key
	pbm_lvninsertitem	InsertItem
	pbm_lvnitemchanged	ItemChanged
	pbm_lvnitemchanging	ItemChanging
	pbm_lvnkillfocus	LoseFocus
	pbm_lvnrclicked	RightClicked
	pbm_lvnsetfocus	GetFocus
Tab	pbm_tcnclicked	Clicked
	pbm_tcnkeydown	Key
	pbm_tcnkillfocus	LoseFocus
	pbm_tcnselchanged	SelectionChanged Note: the event cannot be triggered when a window is opening.
	pbm_tcnselchanging	SelectionChanging Note: the event cannot be triggered when a window is opening.
	pbm_tcnsetfocus	GetFocus
TreeView	pbm_tvnclicked	Clicked
	pbm_tvndeleteitem	DeleteItem
	pbm_tvndoubleclicked	DoubleClicked
	pbm_tvnkeydown	Key
	pbm_tvnitemcollapsed	ItemCollapsed
	pbm_tvnitemcollapsing	ItemCollapsing
	pbm_tvnitemexpanded	ItemExpanded
	pbm_tvnitemexpanding	ItemExpanding
	pbm_tvnitempopulate	ItemPopulate
	pbm_tvnkillfocus	LoseFocus
	pbm_tvnselchanged	SelectionChanged
	pbm_tvnselchanging	SelectionChanging

	pbm_tvnsetfocus	GetFocus
Application	(None)	Open
	(None)	Close
Menu	(None)	Clicked
	(None)	Selected

Unsupported

The following table lists the unsupported system messages for the supported system objects/controls:

Object/Control	Event ID	System Event using the ID
Controls and windows	pbm_other	Other
Window	pbm_deactivate	Deactivate
	pbm_dragdrop	DragDrop
	pbm_dragenter	DragEnter
	pbm_dragleave	DragLeave
	pbm_dragwithin	DragWithin
	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
CheckBox, CommandButton, Picture, PictureButton, RadioButton, StaticText	pbm_bndragdrop	DragDrop
	pbm_bndragenter	DragEnter
	pbm_bndragleave	DragLeave
	pbm_bndragwithin	DragWithin
DataWindow, DataStore	pbm_dwndragdrop	DragDrop
	pbm_dwndragenter	DragEnter
	pbm_dwndragleave	DragLeave
	pbm_dwndragwithin	DragWithin
	(none)	Error
	pbm_dwnretrieverow	RetrieveRow
	pbm_dwnvscroll	ScrollVertical

SingleLineEdit, EditMask, MultiLineEdit, StaticText	pbm_endragdrop	DragDrop
	pbm_endragenter	DragEnter
	pbm_endragleave	DragLeave
	pbm_endragwithin	DragWithin
ListBox, PictureListBox	pbm_lbndragdrop	DragDrop
	pbm_lbndragenter	DragEnter
	pbm_lbndragleave	DragLeave
	pbm_lbndragwithin	DragWithin
ListView	pbm_lvnbegindrag	BeginDrag
	pbm_lvnbeginlabeledit	BeginLabelEdit
	pbm_lvnbeginrightdrag	BeginRightDrag
	pbm_lvndragdrop	DragDrop
	pbm_lvndragenter	DragEnter
	pbm_lvndragleave	DragLeave
	pbm_lvndragwithin	DragWithin
	pbm_lvnendlabeledit	EndLabelEdit
	pbm_lvnitemactivate	ItemActivate
	pbm_lvnrdoubleclicked	RightDoubleClicked
	pbm_lvnsort	Sort
Tab	pbm_tcndoubleclicked	DoubleClicked
	pbm_tcndragdrop	DragDrop
	pbm_tcndragenter	DragEnter
	pbm_tcndragleave	DragLeave
	pbm_tcndragwithin	DragWithin
	pbm_tcnrclicked	RightClicked
	pbm_tcnrdoubleclicked	RightDoubleClicked
TreeView	pbm_tvnbegindrag	BeginDrag
	pbm_tvnbeginlabeledit	BeginLabelEdit
	pbm_tvnbeginrightdrag	BeginRightDrag
	pbm_tvnrclicked	RightClicked Note: the user can use the ContextMenu event as a workaround.
	pbm_tvndragdrop	DragDrop
	pbm_tvndragenter	DragEnter
	pbm_tvndragleave	DragLeave
	pbm_tvndragwithin	DragWithin
	pbm_tvnendlabeledit	EndLabelEdit
Application	(none)	Idle

	(none)	SystemError
OLE	pbm_omndatachange	DataChange
	pbm_omndragdrop	DragDrop
	pbm_omndragenter	DragEnter
	pbm_omndragleave	DragLeave
	pbm_omndragwithin	DragWithin
	(none)	PropertyChanged
	(none)	PropertyRequestEdit
	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
UserObject	pbm_uondragdrop	DragDrop
	pbm_uondragenter	DragEnter
	pbm_uondragleave	DragLeave
	pbm_uondragwithin	DragWithin
PipeLine	pbm_pipeend	PipeEnd
	pbm_pipemeter	PipeMeter
	pbm_pipestart	PipeStart
RichTextEdit	pbm_rendragdrop	DragDrop
	pbm_rendragenter	DragEnter
	pbm_rendragleave	DragLeave
	pbm_rendragwithin	DragWithin
	pbm_renfileexists	FileExists
	pbm_reninputfieldselected	InputFieldSelected
	pbm_renkey	Key
	pbm_renmodified	Modified
	pbm_renlbuttondown	MouseDown
	pbm_renmousemove	MouseMove
	pbm_renlbuttonup	MouseUp

pbm_renpictureselected	PictureSelected
pbm_renprintfooter	PrintFooter
pbm_renprintheader	PrintHeader
pbm_renrbuttondown	RButtonDown

System message (non-standard EventID)

APPEON

Supported

The following table lists the supported system messages. The Notes in the table indicate that the event and event ID are partly supported:

Object/Control	Event ID (non-standard)	System Event using the ID
All	pbm_contextmenu	ContextMenu Note: Event bubbling is unsupported.
DataWindow, DataWindowChild	pbm_dwndropdown	Dropdown
	pbm_dwnkey	KeyDown
	pbm_dwnrbuttonup	Rbuttonup
	pbm_dwnprocessenter	Processenter
	pbm_rbuttondown	
	pbm_lbuttondown	
	pbm_rbuttonup	
	pbm_lbuttonup	

Unsupported

Except for the system messages listed in the table above, all the other system message with non-standard event IDs are unsupported by Appeon. For example, pbm_enchange event is unsupported. Refer to PowerBuilder Help for more details.

DataWindow data sources

APPEON

Supported

The five PowerBuilder data sources are supported:

- Quick Select data source
- SQL Select data source
- Query data source

- Stored Procedure data source
- External data source

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.

DataWindow presentation styles

APPEON

Important Requirements

All the controls and objects in the DataWindow cannot be dynamically created. They can only be created in the painter.

Appeon provides support for distributed DataWindows, which is an efficient way to work around the limitations that are specified in the below requirements. For example, to work around unsupported DataWindow functions, you can place the functions in the distributed DataObject, and then apply the content of the distributed DataObject to the client-side DataWindow.

- For Tabular, Freeform and Grid DataWindows:
    - DataWindows must be defined in the PowerBuilder painter and cannot be dynamically created (e.g. using the CREATE statement).
    - Controls cannot be dynamically created, added or removed from the DataWindow.
- For Label, N-Up, Group, Composite, Graph, CrossTab and Nested DataWindows:
    - DataWindows must be defined in the PowerBuilder painter and cannot be dynamically created (e.g. using the CREATE statement).
    - Controls cannot be dynamically created, added or removed from the DataWindow.
    - Only a few DataWindow functions including Print, SaveAs, and Retrieve are supported; most of the DataWindow functions, such as InsertRow, Update, GetItemValue, are unsupported.
    - If the DataStore presentation style is not Grid, Freeform or Tabular, only the functions Print and SaveAs are supported.
    - Any limitations mandated by PowerBuilder SaveAs method for saving DataWindow contents to WMF must be followed.
    - If a DataWindow contains group(s), it is recommended to disable the option "New Page on Group Break". This will ensure the DataWindow is displayed correctly.

- For Grid DataWindows:
    - If any of the following four controls - Column, Text, Button or Computed Field is placed over a grid line, the control will not display in a deployed DataWindow.
    - If the DataWindow presentation style is Grid, it is unsupported to sort the deployed DataWindow by clicking a column in the header band.
    - 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.
    - The X property of the Grid DataWindow cannot be dynamically changed.
    - In the Grid DataWindow, a non-Detail band object cannot be placed in the area between columns. If it is placed there, the object will not display correctly.
    - In the Grid DataWindow, 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.
    - In PowerBuilder, when the Visible property of a column in the detail band is set to FALSE, both the column and the column header are invisible. In the Web application, the column will be invisible, but the column header remains visible.

Refer to the *Application enhancements and differences* section on user interface and operations for DataWindow.

Supported

- Tabular DataWindow
- Freeform DataWindow
- Grid DataWindow
- Other DataWindows that can be converted to the Web as an image and printed to the PDF format:
    - Label DataWindow
    - N-Up DataWindow
    - Group DataWindow
    - Composite DataWindow
      It is unsupported to scroll a composite DataWindow using the PageUp/PageDown, UpArrow/DownArrow, LeftArrow/RightArrow keys.
    - Graph DataWindow
    - CrossTab DataWindow
    - Nested DataWindow

Unsupported features and workarounds

The following PowerBuilder presentation styles are unsupported:

- OLE DataWindow
- RichText DataWindow

The following DataWindow features can be well deployed and displayed in Image, PDF and Exported DataWindows, but they will not be available in HTML DataWindows:

- Defining one or more groups in an existing Grid, Freeform or Tabular DataWindow
- Group header band and trailer band
- Data grouping
- All the functions specific to Label, N-Up, Group, Composite, Graph, CrossTab and Nested DataWindows are not supported. For example: using the FindGroupChange function in the RetrieveEnd event for determining the number of group level breaks in a DataWindow is not supported.

For the Freeform DataWindow, it is supported to change the value of the X property. However, setting the property to a value greater than the current width of the DataWindow has no effect.

Displaying and validating data

Important Requirements

In PowerBuilder, default display formats, such as currency or time or date, do not need to be defined; they can be read from the Windows registry and use the default that the user has specified in Windows. However, Web applications in general are prevented from accessing the Windows registry. Therefore, you must either explicitly specify what the default display format is in the PowerBuilder property painter, or configure the desirable display formats in Appeon Enterprise Manager for currency, time and date data.

You must explicitly specify what the default display format is in the PowerBuilder property painter, otherwise the Web application will not display the correct format or any format at all. In other words, in PowerBuilder the default for each user can be different since it can be specified in Windows, but on the Web with Appeon the default needs to be specified in the PowerBuilder application so it would be the same default for everybody, regardless of how they have set their computers.

Supported

Data validation includes:

- 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).

The format property is supported for column edit style Edit and EditMask. The display formats allowed for different data types are listed in the table below:

Data Type	Coding Requirements

Number

#Number
##,###.00
##,###.##
#####
Currency(x) –international
$#,##0;[RED]($#,##0)
$#,##0.00;[RED]($#,##0.00)
Note: if the currency is not US dollar, to ensure that the currency symbol displays correctly in the deployed Web application, it is better to specify the format with the dollar symbol replaced with the actual currency symbol. For example, the format is #,##0.00. Please do not rely on the setting of the client machine to add the actual currency symbol.
0
0.00
0%
0.00%
#,##0.00
#,##0
$#,##0;($#,##0)
$#,##0.00;($#,##0.00)
0.00E + 00

String

! Upper Case
^ Lower Case
# Number
a Alphanumeric
x Any character
###-##-####
#####
@
(@@@)@@@@@
Note: you can only use comma as separator(s) in a string format.

Date

- Separator: "/", "-",
- Separator: Chinese character year, month and date.

mm +separator + dd + separator + yy (eg. mm/dd/yy or mm-dd-yy)
dd + separator + mm + separator + yy
yy + separator + mm + separator + dd
mm + separator + dd + separator + yyyy
dd + separator + mm + separator + yyyy
yyyy + separator + mm + separator + dd

Time

- Separator: "/", "-"
- Default format: hh:mm:ss:fff

[time]
hh:mm
hh:mm:ss
hh:mm:ss:fff
hh:mm:ss:ffffff
hh-mm-ss
hhmmss
Note: fraction in time is unsupported but can be identified.

DateTime

- Separator: "/", "-"
- Separator1: " " (space)
- Default format: mm/dd/yyyy hh:mm:ss:ffff

mm + separator + dd + separator + yy + hh:mm:ss (eg. mm/dd/yy or mm-dd-yy)
dd + separator + mm + separator + yy + hh:mm:ss
yy + separator + mm + separator + dd + hh:mm:ss
mm + separator + dd + separator + yyyy + hh:mm:ss
dd + separator + mm + separator + yyyy + hh:mm:ss
yyyy + separator + mm + separator + dd + hh:mm:ss
mm + separator + dd + separator + yy + Separator1 + hh:mm:ss:fff
dd + separator + mm + separator + yy + Separator1 + hh:mm:ss:fff
yy + separator + mm + separator + dd + Separator1 + hh:mm:ss:fff
mm + separator + dd + separator + yyyy + Separator1 + hh:mm:ss:fff
dd + separator + mm + separator + yyyy + Separator1 + hh:mm:ss:fff
yyyy + separator + mm + separator + dd + Separator1 + hh:mm:ss:fff

Unsupported

The display formats that are not listed as supported are unsupported. For example:

- Date or DateTime with full month name "mmmm"
- Date or DateTime with full day name "dddd" or day name abbreviation "ddd"
- Am/Pm in date or DateTime
- In PowerBuilder, if the format of a column is set to "#0%" with the SetItem function used, the column may allow a four-digit input (for example, "100%". In Web applications, with the "#0%" format, the column only allows three-digit input, and will cut the value "100%" to "10%".
- If a Mask is set as a four-digit string, when you use SetItem to set a value to the EditMask column, the column will only retrieve the first four digits of the entered value, which is different from in PowerBuilder.

DataWindow operators and expressions

APPEON

Supported

DataWindow expressions can be used in the following cases:

- Setting of properties such as Visible, X, Y, Width, Height, BackgroundColor, TextColor
- Computed columns
- Input validation
- Filter and sort criteria

The operators that can be used in DataWindow expressions are as follows:

- Arithmetic operators: +, -, *, /, ( )
- Relational operators: =, >, <, <>, >=, <=
- Logical for all data types: NOT, AND, OR
- Concatenation for string data types: +.

The order of precedence for operators is different than in PowerBuilder. If the "AND" and "OR" operators

coexist in a DataWindow expression, in PowerBuilder, the "AND" and "OR" have the same precedence, but for Appeon deployed DataWindow, the precedence of "AND" is higher than that of "OR". Use of parentheses may be required to have the correct precedence effect. For example, in the Web application, "(a OR b) AND c" is supported, but "a OR b AND c" will not produce the expected result.

In PowerBuilder, the operators >, <, <=, >=, =, <> are of the same precedence. With the deployed DataWindows, the operators >, <, <=, >= are of higher precedence than = and <>.

The functions that can be called in DataWindow expressions are as shown in the following list.

Abs	Acos	Asc	Asin	Atan	Avg*
Ceiling	Char	Cos	Count*	CurrentRow*	Date
DateTime	Day	DayName	DayNumber	DaysAfter	Exp
GetText*	GetRow	Hour	If	Int	Integer
IsDate	IsNull	IsNumber	IsRowModified	IsRowNew	IsSelected
IsTime	Left	LeftTrim	Len	Log	LogTen
Long	Lower	Match	Max*	Mid	Min*
Minute	Mod	Month	Now	Number	Page*
PageCount*	Pi	Pos	Rand	Real	RelativeDate
RelativeTime	Replace	RGB	Right	RightTrim	Round
RowCount	Second	SecondsAfter	Sign	Sin	Space
Sqrt	String*	Sum*	Tan	Time	Today
Trim	Truncate	Upper	WordCap	Year	

Notes:

- The functions CurrentRow and GetText cannot be used in computed column expressions.
- The functions Avg, Count, Max, Min and Sum cannot be used in sort and filter expressions.
- The functions Avg, Count, Max, Min, Sum, Page and PageCount cannot be used in property and validation expression.
- In PowerBuilder, the *Date* argument can be a datetime or a date. When the argument's value is a datetime, RelativeDate returns a datetime; the argument's value is a date, RelativeDate returns a date. However, on the Web application, the *Date* argument can only be a date and the RelativeDate returns a date.
- String
  Supported syntax: String (data)
  Unsupported syntax: String (data, format)

Unsupported

The following table lists the unsupported features for DataWindow expressions:

Expression Type	Unsupported Feature
All expressions	- Arithmetic operator: ^ (Exponentiation) - Relational operators: NOT =, LIKE, IN, BETWEEN, NOT LIKE, NOT IN, NOT BETWEEN - Comparison between Datetime, Date or Time data. For example, Date1>DateTime1 or DateTime2 = Date2 - User-defined functions - Matches any group of characters %(percent) - Matches any single character _(underscore) - Comment marker //
Computed column expressions	- Function CurrentRow and GetText - Dynamic configuration of the expressions - Using the keyword Distinct in the function Avg, Count, Max, Min and Sum - Using Computed column as parameter in the Avg, Count, Max, Min and Sum functions
Sort and Filter expressions	- Function Avg, Count, Max, Min and Sum - In a Web application, if a Find, Filter or Sort expression contains any of the 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	- Function Avg, Count, Max, Min, Sum, Page and PageCount - Overlapped quotes, for example, "sdf~"sdf", "dfg'sdf" - In the Web application, modifying a DataWindow property in a DataWindow expression may conflict with the settings in the source code. - Expressions with the following properties: • Protect or Border property • Format, Height, Width properties of Column control • Font properties (However, expressions with property Text.Color, BackGround.Color are supported) • Height, X, Y, Width properties with Grid DataWindow control
Validation expressions	- Function Avg, Count, Max, Min, Sum, Page and PageCount - Validation expressions cannot include column names.

In addition to the above, DataWindow expressions cannot use any of the following functions:

Bitmap	Case	CrosstabAvg	CrosstabCount
CrosstabMax	CrosstabMin	CrosstabSum	CumulativePercent
CumulativeSum	Describe	Fact	Fill
First	Large	Last	LastPos

LookUpDisplay	Median	Mode	PageAcross
PageCountAcross	Percent	ProfileInt	ProfileString
RowHeight	Small	StDev	StDevP
Var	VarP		

- Using user-defined functions in DataWindow expressions is unsupported.
- Subtraction (-). The setting of DashesInIdentifiers property is ignored. That is, "A-B" always means subtract B from A.
- 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.
- DataWindow expressions containing retrieval arguments are unsupported.
- DataWindow expressions cannot contain any argument.
- With the expression Min(c_id) + Max(c_id), where c_id is a char type variable, in PowerBuilder, '101' + '109' = 101109; on the Web, '101' + '109' = 210 (which is calculation on two numbers).

DataWindow object and the properties

APPEON

DataWindow object

Controls in a DataWindow and their properties

DataWindow object properties

DataWindow object

APPEON

Important Requirements

DataWindow object cannot be dynamically created (for example, using the CREATE statement); it must be defined in the PowerBuilder painter.

Supported

The following DataWindow controls are supported:

Button	Column	Computed Field	Line

Text

- Column control

- In the Web application, when the alignment for a column within its borders is set to Center, the CheckBox or RadioButton will not display as center-aligned. This is different from PowerBuilder.

- The name of a DataWindow column cannot be the same as the dbname of another column.

- Computed Field control

  - Computed Field control does not support attaching retrieval arguments to it.
  - The Line control in DropDownDataWindow will be ignored.
  - The column name of a computed field cannot have the same name as the column in the database.

The following column edit styles are supported:

CheckBox	DropDownDataWindow	DropDownListBox	Edit
EditMask	RadioButtons		

- DropDownDataWindow does not support DropDownDataWindow or DropDownListBox in it.
- DataWindow RadioButtons can only get focus by mouse clicking. They cannot get focus by stroking the Tab key.

Four DataWindow bands are supported with some limitations:

DataWindow Band	**Limited Support**
Detail band	For Grid DataWindow: - Each column can only have one control in it. - The control can be Button, Column, or Computed Field control. For Freeform DataWindow: - The column can have any of the supported controls: Button, Column, Computed Field or Text, and without restriction on the control number. - If the detail band is dragged upward to make some fields in the DataWindow invisible, be sure to uncheck the Visible property of these fields. Otherwise, the Image View of the deployed DataWindow will display these fields that shall be invisible over the fields that shall be visible, and the Image View is distorted. - If the height of the detail band is set to nnn in the painter, dynamically changing the height to a value smaller than nnn has no effect, while dynamically changing the height to a value larger than nnn takes effect. For Grid and Tabular DataWindow: - It is unsupported to place more than one object in the detail band.
Footer band	- Support all the controls: Button, Column, Computed Field or Text - In the Web application, no matter whether there is data in the Footer band or not, the column(s) will still be displayed.

Header band	- PowerBuilder DataWindow displays the content of the band according to the property Header.Height; However, Appeon-deployed DataWindow displays all the content in the band regardless of the Header.Height. The value of Header.Height cannot be 0. - In PowerBuilder, the header band is always unmovable. However, in the Web DataWindow, if there are too many rows to be displayed on one page, the header band will move as the scrollbar is scrolled up and down. For Grid DataWindow: - Each column can only have one control in it. - The control can be Button, Computed Field or Text control. - Button control must be placed within the border of the column. - The UI of Column control can be converted, but the data cannot display. For Freeform DataWindow: - The column can have any of the supported controls: Button, Column, Computed Field or Text, and without restriction on the control number For Grid, Freeform and Tabular DataWindows: - Although newline and carriage return characters are supported, for Appeon-deployed DataWindows (Grid, Freeform and Tabular), the use of carriage return and newline (~r~n) to make a new line in a column header is ignored in the converted Web application. - The column headers in deployed DataWindows automatically wrap text if the text is wider than the column header itself. - In the Web application, no matter whether there is data in the Head band or not, the column(s) will still be displayed.
Summary band	- Support all the controls: Button, Column, Computed Field or Text - If the content in the summary band cannot fully display at the last page of the deployed DataWindow, an additional page will show for displaying the rest of the content. This is different from PowerBuilder.

Unsupported

The following DataWindow controls are unsupported:

Graph	GroupBox	OLE	Oval
Picture	Rectangle	Report	RoundedRectangle
TableBlob			

The DataWindow band Trailer band is supported in Image and PDF DataWindows although not in HTML DataWindows.

In a Web application, it is unsupported to set the BringToTop or the FrontToBack property for DataWindow controls or objects.

Controls in a DataWindow and their properties

Button control

Column control

Computed Field control

Line control

Text control

Unsupported controls

Note: avoid placing controls above or behind other controls in a DataWindow, as overlapping controls are displayed differently on the Web to PowerBuilder. For example, if there is a Text control behind a DropDownDataWindow column in a DataWindow, the DropDownDataWindow field cannot be pulled down correctly.

Button control

APPEON

Supported

If a property is marked as "supported(r)" or "(r)", it means that the property can be read in script, but the property cannot be changed in script, and setting the property in the painter takes 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.

Property	What's Supported	What's Unsupported
Action	User Defined, Retrieve (Yield), Retrieve, PageNext, PagePrior, PageFirst, PageLast, Sort, Filter, DeleteRow, AppendRow, InsertRow, Update, SaveRowsAs, Print	Cancel, Preview, PreviewWithRulers, QueryMode, QuerySort, QueryClear The Action property cannot be set by using DataWindow expression. The property value being 3 (Cancel), 16 (Preview), 17 (PreviewWithRulers), 18 (QueryMode), 19 (QuerySort) or 20 (QueryClear) is not supported. If the button action is “sort”, different behavior will occur than seen in PowerBuilder. If the user clicks the “sort” button, the “specify sort column” window appears: (1) there will be no effect if the user double-clicks a column name in the Column listbox, the Modify Expression window will not appear as in PowerBuilder; (2) the

		user cannot drag the same column into the Columns listbox, there cannot be multiple identical columns in the Columns listbox.
Background.property	Color The Background.property property can be set by using DataWindow expression.	Mode
Band	Band	Background, Foreground
Color	The Color property can be set by using DataWindow expression.	
DefaultPicture		The DefaultPicture property cannot be set by using DataWindow expression.
Filename	Fully supported	
Font.*property*	(r/w): Face, Height, Italic, Strikethrough, Underline, Weight (r): Family, Charset, Pitch	Escapement, Width The Font.*property* property cannot be set by using DataWindow expression.
Height	The Height property can be set by using DataWindow expression.	
HTextAlign	Fully supported (r/w)	
Name	Fully supported (r/w)	
Pointer	Arrow!, Beam!, Cross!, HourGlass!, HyperLink!, Size!, SizeNESW!, SizeNWSE!, SizeWE!, UPArrow!	AppStarting!, Help!, Icon!, No!, SizeNS! All unsupported pointers are replaced with Arrow! in the Web application. The Pointer property cannot be set by using DataWindow expression. User-defined pointers are not supported.
SuppressEventProcessing	Fully supported (r/w)	
Tag	Fully supported (r/w)	
Text	Fully supported (r/w)	
Type	Fully supported (r/w)	
VTextAlign		MultiLine(3)

Visible	The Visible property can be set by using DataWindow expression.
Width	The Width property can be set by using DataWindow expression.
X	The X property can be set by using DataWindow expression.
Y	The Y property can be set by using DataWindow expression.

Unsupported

The following table lists the unsupported properties of Button control:

Attributes	HideSnaked	Moveable	Resizeable
SlideLeft	SlideUp		

Column control

APPEON

Supported

If a property is marked as "supported(r)" or "(r)", it means that the property can be read in script, but the property cannot be changed in script, and setting the property in the painter takes 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.

Property	What's Supported	What's Unsupported
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
Background.*property*	Color	Mode
Band	Band	Background, Foreground

Border	NoBorder!, Box!, Lowered!, Raised!	ShadowBox!, Underline!, ResizeBorder! The Border property cannot be set as an expression of value.
CheckBox.*property*	(r/w): Off, On, Text (r): 3D or ThreeD, LeftText (3D only)	Other, Scale
Color	Set the Color property using DataWindow expression	
ColType	dw_control.object.columnname.coltype dw_control.Describe ("columnname.coltype")	
dbName	Example: dw_control.object.columnname.dbName dw_control.Describe ("columnname.dbName")	The dbName property cannot be dynamically changed.
dddw.*property*	(r/w): AutoHScroll, AutoRetrieve, DataColumn, DisplayColumn, HscrollBar, NilIsNull, PercentWidth, VscrollBar (r): AllowEdit, Limit, Name, UseAsBorder, Required The size, font size of Web DropdownDataWindow are determined by the size of the deployed DataWindow; the width of Web DropdownDataWindow, if set to be smaller than the column width, will be displayed in the same width as the column. In PowerBuilder, on the left-click of an item in the ListBox, the user can move the mouse up and down to change the current focus. The ListBox will not close nor will the display column get the item value until the user looses the left-click. However, in the Web application, once on the left-click of an item, the list box will close and the display column will get the item value.	Case, HsplitScroll, Lines, ShowList
ddlb.*property*	(r/w): AutoHScroll, NilIsNull, Sorted, VScrollBar. (r): AllowEdit, Limit, ShowList, Required For the ddlb.property, when the value	Case, UseAsBorder In the Web application, when the user clicks the DropDownListBox and then presses the Tab key, the focus will not move from the current field into

	input in a column is saved, a whitespace will be added behind the string of the value. 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.	another field. In a Web application, if a DropDownListBox has items of the same value, the items will all be displayed in separate rows in the ListBox portion. This is different from PowerBuilder.
Edit.*property*	(r/w): DisplayOnly, Limit, NilIsNull, Password (r): AutoHScroll, AutoSelect, AutoVScroll, HscrollBar, Style, VscrollBar, Required Setting of Edit.HscrollBar and Edit.VScrollBar will be ignored when Edit.limit or Edit.password is set. In the Web application, the user can click the up and down arrows to cycle through the minimum and maximum values with a spin control. This is different from PowerBuilder. 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. In PowerBuilder, if the Edit.AutoHScroll property is set to FALSE and the Edit.Limit is set to 0, no more characters can be input to the column after the field is full. However, in a Web application, the user can still input more characters.	Case, CodeTable, FocusRectangle, Format , Name, ValidateCode If the Limit is set to a value outside the scope 0~255 in the PowerBuilder painter, the Edit.*property* property cannot be dynamically changed.
EditMask.*property*	(r/w): ReadOnly, SpinIncr, SpinRange (r): Mask, Spin, UseFormat, Required In the Web application, if the Mask of a column is set to General, the values in the column are all displayed as integers when the column gets focus.	Unsupported: AutoSkip, CodeTable, FocusRectangle Avoid changing the EditMask.SpinIncr dynamically. If the EditMask.SpinIncr has been specified in the painter, and then the user dynamically changes the property value; in the Web application, the incremental value at the click of the spin may be the old value first, then the new value. This is different behavior from PowerBuilder. In PowerBuilder, if the edit style of DataWindow column is EditMask, and the EditMask.Spin and EditMask.ReadOnly properties are

		both set to Yes, the user can still enter data in the column, or scroll through a list of possible values for the column with a spin control. However, with the same settings, the column of a deployed DataWindow does not allow user to enter data. When the mask is “0.00E+0.00”, the digits to the left of the decimal separator will be ignored.
Font.*property*	(r/w): Face, Height, Italic, Strikethrough, Underline, Weight (r): Family, Charset, Pitch	Escapement, Width The Font.*property* property cannot be set by using DataWindow expression.
Format	Edit	The Format property cannot be set by using DataWindow expression. When the user sets the Format property, the property value cannot include any space. If the format of a column is datetime, regardless of what the format is actually set to, the data in the column will always display time. In a Web application, if you specify a string for the display format of a column, you can use only one delimiter between different format portions.
Height	Set the Height property using DataWindow expression. All the rows in a Web Grid/Tabular DataWindow will be of the same height. However, with PowerBuilder DataWindow, it is possible to assign different heights for different rows by using DataWindow expression for the Height property of the column control.	In the Web application, when the Height of a column is set as 0, the Freeform DataWindow presents differently from that in PowerBuilder.
Initial	Fully supported (r/w)	
ID	Fully supported (r/w)	
Key	Fully supported (r/w)	
Name	Fully supported (r/w)	
Pointer		The Pointer property cannot be set by using DataWindow expression.
Protect	The Protect property can be set by using DataWindow expression.	In a deployed DataWindow, if the Protect property of all columns is set to 1, the user will not be able to select

		any rows. When a new row is inserted, the Protect property will not be validated.
RadioButtons.property	(r): 3D, Columns, LeftText (3D only)	Scale
Resizeable	Supported (r)	The Resizeable property cannot be dynamically changed.
TabSequence	Fully supported (r/w)	
Tag	Fully supported (r/w)	
Type	Fully supported (r/w)	
Update	Fully supported (r/w)	
Validation	Supported (r)	The Validation property cannot be dynamically changed.
Values	Supported (r) The Values property is supported for columns with the following edit styles: DropDownListBox, CheckBox, and RadioButtons.	The Values property cannot be dynamically changed.
Visible	The Visible property can be set by using DataWindow expression.	The same value of Visible property arrive at different UI result in PowerBuilder and on the Web.
Width	The Width property can be set by using DataWindow expression.	
X	Advoid using DataWindow expression to set a value to the X property.	
Y	Advoid using DataWindow expression to set a value to the Y property.	

Unsupported

The following table lists the unsupported properties of the Column control:

Accelerator	Attributes	BitmapName	Criteria.property
Height.AutoSize	HideSnaked	HTML.property	Identity
LineRemove	Moveable	MultiLine	SlideLeft
SlideUp	ValidationMsg	Width.Autosize	

Computed Field control

Supported

If a property is marked as "supported(r)" or "(r)", it means that the property can be read in script, but the property cannot be changed in script, and setting the property in the painter takes 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.

Property	What's Supported	What's Unsupported
Alignment	Left!, Center!, Right!, Justify!	None
Background.property	Color	Mode
Band	Band	Background, Foreground
Border	NoBorder!, Box!, Lowered!, Raised!	ShadowBox!, Underline!, ResizeBorder!
		The Border property cannot be set by using DataWindow expression.
Color	The Color property can be set by using DataWindow expression.	
Expression	Supported (r)	The Expression property cannot be set by using DataWindow expression.
Font.property	(r/w): Face, Height, Italic, Strikethrough, Underline, Weight	Escapement, Width
	(r): Family, Charset, Pitch	
Format	Supported (r)	The Format property cannot be set by using DataWindow expression.
Height	The Height property can be set by using DataWindow expression.	
Multiline	Supported (r)	The Multiline property cannot be set by using DataWindow expression.
Name	Fully supported	
Pointer	Supported (r)	The Pointer property cannot be set by using DataWindow expression.
Resizeable	Supported (r)	The Resizeable property cannot be set by using DataWindow expression.
Tag	Fully supported (r/w)	
Type	Fully supported (r/w)	
Visible	The Visible property can be set by using DataWindow expression.	

Width	The Width property can be set by using DataWindow expression.
X	The X property can be set by using DataWindow expression.
Y	The Y property can be set by using DataWindow expression.

Unsupported

The following table lists the unsupported properties of Computed Field control:

Attributes	ColType	Height.Autosize	HideSnaked
HTML.property	LineRemove	Moveable	SlideLeft
SlideUp	Width.Autosize		

Line control

APPEON

Supported

If a property is marked as "supported(r)" or "(r)", it means that the property can be read in script, but the property cannot be changed in script, and setting the property in the painter takes 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.

Property	What's Supported	What's Unsupported
Band	Supported (r)	The Band property cannot be dynamically changed.
Name	Supported (r)	The Name property cannot be dynamically changed.
Pen.*property*	Supported (r)	The Pen.property property cannot be dynamically changed.
Visible	Supported (r)	The Visible property cannot be dynamically changed.
X1, X2	Supported (r)	The X1, X2 property cannot be dynamically changed.
Y1, Y2	Supported (r)	The Y1, Y2 property cannot be dynamically changed.

Unsupported

The following table lists the unsupported properties of Line control:

Attributes	Background.*property*	HideSnaked	Moveable
Pointer	Resizeable	SlideLeft	SlideUp
Tag	Type		

Text control

APPEON

Supported

If a property is marked as "supported(r)" or "(r)", it means that the property can be read in script, but the property cannot be changed in script, and setting the property in the painter takes 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.

Property	What's Supported	What's Unsupported
Alignment	Left!, Center!, Right!, Justify!	None
Background.*property*	Color The Background.property can be set by using DataWindow expression.	Mode
Band	Band	Background, Foreground
Border	NoBorder!, Box!, Lowered!, Raised!	ShadowBox!, Underline!, ResizeBorder! The Border property cannot be set by using DataWindow expression.
Color	The Color property can be set by using DataWindow expression.	
Font.property	(r/w): Face, Height, Italic, Strikethrough, Underline, Weight (r): Family, Charset, Pitch	Escapement, Width The Font.property cannot be set by using DataWindow expression.
Height	The Height property can be set by using DataWindow expression.	
Name	Fully supported (r/w)	
Tag	Fully supported (r/w)	

Text		The Text property cannot be set by using DataWindow expression.
Type	Fully supported (r/w)	
Visible	The Visible property can be set by using DataWindow expression.	
Width	The Width property can be set by using DataWindow expression.	
X	The X property can be set by using DataWindow expression.	
Y	The Y property can be set by using DataWindow expression.	

Unsupported

The following table lists the unsupported properties of Text control:

Attributes	Height.AutoSize	HideSnaked	HTML.property
Moveable	Pointer	Resizeable	SlideLeft
SlideUp			

Unsupported controls

APPEON

Unsupported

The properties of all the following unsupported controls are unsupported:

Graph	GroupBox	OLE	Oval
Picture	Rectangle	Report	RoundedRectangle
TableBlob			

DataWindow object properties

APPEON

Supported

If a property is marked as "supported(r)" or "(r)", it means that the property can be read in script, but the property cannot be changed in script, and setting the property in the painter takes 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.

None of Graphical DataWindow properties are supported.

The table below lists the supported properties for DataWindow object:

Property	What's Supported	What's Unsupported
Units	PowerBuilder	Pixels, 1/1000Inch, 1/1000Centimeter
Bandname.property	(r): Detail, Height.autosize Detail.Color, Detail.Height, Detail.Height.Autosize, Detail.Pointer, Footer.Color, Footer.Height, Footer.Height.Autosize, Footer.Pointer, Header.Color, Header.Height, Header.Height.Autosize, Header.Pointer, Summary.Color, Summary.Height, Summary.Height.Autosize, Summary.Pointer When the detail area is not fully covered by the data columns, the setting of the background color will only apply to the data columns.	Trailer.Color, Trailer.Height, Trailer.Height.Autosize, Trailer.Pointer
Bands	Supported (r)	The Bands property cannot be dynamically changed.
Color	Set the Color property using DataWindow expression	
Column.count	Supported (r) If a Button control is placed in the last column, the column always shows the grid line, even if the Grid.Lines is set to 1 (not displayed).	The Column.count property cannot be dynamically changed.
FirstRowOnPage	Supported (r)	The FirstRowOnPage property cannot be dynamically changed.
Grid.Lines	Fully supported (r/w) The value of the Grid.Lines property can be set to 0, 1, 2, or 3. However, in the Web application, only 0 and 1 take effect.	
LastRowOnPage	Supported (r)	The LastRowOnPage property cannot be dynamically changed.

Name	Supported (r)	The Name property cannot be dynamically changed.
Objects	Fully supported (r/w) Note: in the Web application, the returned object names are listed in a different order from that in PowerBuilder.	
Processing	Fully supported (r/w)	
QueryMode	Supported (r) The QueryMode property is partially supported. It is supported to set or get the property, however, the query mode is always disabled, even if the property is set to TRUE.	
ReadOnly	Fully supported (r/w)	
Selected.Mouse	Supported (r)	The Selected.Mouse property cannot be dynamically changed.
Syntax.Data	Fully supported (r/w)	
Table.*property*	Arguments, Filter, Select, Sort, UpdateKeyInPlace, UpdateTable, UpdateWhere 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. When changing the value of Table.Select property, the user can only make changes to the WHERE clause. Changes to the SELECT clause are unsupported. In addition, changes to the WHERE clause cannot be verified. 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 have more	CrosstabData, Delete.Argument, Delete.Method, Delete.Type, GridColumns, Insert.Argument, Insert.Method, Insert.Type, Procedure, Select.Attribute, SQLSelect, Storage, Update.Argument, Update.Method, Update.Type Table.property is not supported for External data source.

	than one user accessing the same tables concurrently, to avoid data loss, set the DataWindow Table.UpdateWhere property to 1 (“Key and updatable columns”).
Zoom	The effect of a zoom factor presents differently in PowerBuilder from in the Web application.

Unsupported

All column edit styles are unsupported:

CheckBox	DropDownDataWindow	DropDownListBox	Edit
EditMask	RadioButton		

DataWindow data and property expressions

APPEON

Supported

1) The following DataWindow data expressions are supported:

- Syntax for one or all data item in a named column

  *dwcontrol.*Object*.dwcolumnname{.buffer} {.datasource}{[rownum]}*

- Syntax for selected data in a named column

  *dwcontrol.*Object*.dwcolumnname{.*Primary*}{.datasource}.*Selected

- Syntax for a range of data in a named column

  *dwcontrol.*Object*.columnname{.buffer}{.datasource}[startrow,endrow]*

- Syntax for a single data item in a DataWindow

  *dwcontrol.*Object.Data *{.buffer}{.datasource}[rownum, colnum]*

- Syntax for data in a block of rows and columns

  *dwcontrol.*Object.Data*{.buffer}{.datasource}[startrow,startcol, endrow, endcol]*

- Syntax for data in a single row or all rows

  *dwcontrol.*Object.Data*{.buffer}{.datasource}{[rownum]}*

- Syntax for all data from selected rows

  *dwcontrol.*Object.Data*{*.Primary*}{.datasource}.*Selected

2) The expression for accessing to the text displayed in the column header is supported:

dwcontrol.Object.*ColumnName*.*Text*

3) Basic syntax for DataWindow property expressions is supported:

*dwcontrol.*Object.*dwcontrolname{.property}.property{=value}*

4) Set or get the value of a dot notation is supported, except that there is structure data in the expression

Unsupported

- It is unsupported to get the data value from a filter buffer according to the row number. This is because data is stored in 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: Appeon Unsupported Features Analysis tool cannot detect the unsupported features in the DataWindow dot notations. The user 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.

DataWindow constants

APPEON

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:

Constant	Supported Values	Unsupported Values
Alignment	Left!	None

	Center! Right! Justify!	
Band	Detail! Header! Footer!	None
Border	NoBorder! Box! Lowered! Raised!	ShadowBox! Underline! ResizeBorder!
BorderStyle	StyleBox! StyleLowered! StyleRaised!	StyleShadowBox!
CharSet	CharSetAnsi! CharSetUnicode! CharSetAnsiHebrew! CharSetAnsiArabic!	CharSetDBCS-Japanese!
DWBuffer	Primary! Delete! Filter!	None
DWItemStatus	NotModified! DataModified! New! NewModified!	None Note that 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: if the row's status is changed from New! Into NewModified and updated in script.
SaveAsType	Excel! HTMLTable! Text! WMF!	CSV!, SYLK!, WKS!, WK1!, DIF!, dBASE2!, dBASE3!, SQLInsert!, Clipboard!, PSReport!, Excel5!
SQLPreviewFunction	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:

ConnectionSource	DriverType	DWConflictResolution	FillPattern
grColorType	grDataType	grObjectType	grSymbolType

LineStyle	RowFocusInd

DataWindow UI dynamic manipulation

APPEON

Unsupported

DataWindow UI dynamic manipulation is unsupported:

- Adding or deleting controls from the DataWindow object
- Dynamically creating the DataWindow object during execution

DataWindow control

Properties | Events | Functions

APPEON

Properties of DataWindow control

Properties | Events | Functions

APPEON

Supported

The supported properties of DataWindow controls are listed in the table below:

Border	BorderStyle*	DataObject	Enabled*
Height	HScrollBar	Object	TabOrder
Tag	Visible*	VScrollBar	Width
X	Y		

* Three types of BorderStyle are supported: Box, 3D Raised, 3D Lowered, but Shadow Box is not supported.

* Enabled
- When the Enabled property is changed from False to True, data in the DataWindow header area will have a drop shadow or display out of place.
- When the Enabled property is FLASE, all DataWindow controls will be grayed.

* In PowerBuilder, when the VScrollBar or HScrollBar property 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.

* In PowerBuilder, when the Visible property of a column in the detail band is FALSE, both the column and the column header are invisible. In a Web application, the column is invisible but the column header remains visible; the gridlines stay visible or invisible without being affected by the column Visible property.

Unsupported

The unsupported properties of DataWindow controls are listed in the table below:

BringToTop	ClassDefinition	ControlMenu	DragAuto
DragIcon	HSplitScroll	Icon	LiveScroll
MaxBox	MinBox	Resizeable	RightToLeft
Title	TitleBar		

Events of DataWindow control

Properties | Events | Functions

Supported

1) Only the SQLPreview event is supported for the following DataWindow presentation styles: Label, N-Up, Group, Composite, Graph, Crosstab, and Nested DataWindows.

2) The following events are supported for deployed DataWindows Tabular, Freeform and Grid:

ButtonClicked	ButtonClicking	Clicked	Constructor
DBError*	Destructor	DoubleClicked	EditChanged*
GetFocus	ItemChanged*	ItemError*	ItemFocusChanged*
LoseFocus	RButtonDown	Resize*	RetrieveEnd
RetrieveStart	RowFocusChanged*	RowFocusChanging*	ScrollHorizontal*
SQLPreview*	UpdateEnd	UpdateStart	

- DBError

  The *sqlsyntax* argument does not work in the DBError event.

  The GetSQLPreview function in the DBError event will be ignored.

  The *row* argument in the DBError event will be ignored. Its value will always be considered as 0.

- EditChanged

  In PowerBuilder, when the user selects all the text of an edit column and types another value, the EditChanged method will be triggered twice. One is when the text of the column is being deleted, whereas the other is when a new value is being input. However, in the Web application, the EditChanged method will only trigger once for changing the text value.

  If a column's edit style is EditMask, it is unsupported to call the AcceptText function in the EditChanged event in a Web application.

- ItemChanged

  In a Web application, if a changed value of a column does not pass validation, the focus will not go back to the column being edited.

- ItemError

  In a Web application, the ItemError event cannot return 1.

- Resize

  When the value of Width or Height is reset, the Resize event will be triggered, no matter whether the value is changed or remains the same.

- RowFocusChanged

  In PowerBuilder, the RowFocusChanged will be triggered twice at the execution of the RowsCopy or RowsMove function. However, in the same scenario on the Web, the RowsFocusChanged event will only be triggered once.

- ItemChanged, RowFocusChanged, and RowFocusChaning

  It is unsupported to call the SetItem, Sort, Filter, ModifiedCount or Retrieve function in the ItemChanged, RowFocusChanged, or RowFocusChaning event.

- ScrollHorizontal

  When the user scrolls a DataWindow horizontally, the ScrollHorizontal event is triggered many times on the Web. This is different behavior from in PowerBuilder, where in the same case, the event is triggered only once.

- SQLPreview

  The SQLPreview event can only be triggered by the Retrieve function, not by Update or ReselectRow. The *sqltype* argument only can use a SELECT statement, but not INSERT, DELETE, or UPDATE statement.

Unsupported

1) If a DataWindow event requires the server interaction, and the event is immediately followed by another event, the second event cannot be executed.

2) The following events are unsupported for deployed DataWindows Tabular, Freeform and Grid:

DragDrop	DragEnter	DragLeave	DragWithin
Error	Help	Other	PrintEnd
PrintPage	PrintStart	RetrieveRow	ScrollVertical

Functions of DataWindow control

Properties | Events | Functions

Supported

1) The following functions are supported for the DataWindow presentation styles Label, N-Up, Group, Composite, Graph, CrossTab, and Nested DataWindows:

Describe*	Filter*	GetSQLSelect	Modify*
Print	Retrieve	RowCount	SaveAs
SetFilter*	SetSQLSelect		

- Filter and SetFilter
  It is of slow performance to execute Filter or SetFilter function for an Image DataWindow.

- Modify
  For the deployed Graphical DataWindow, only the table.Select property can be used with Modify and Describe functions.

2) The following DataWindow functions are supported:

AcceptText	ClassName	DeletedCount	DeleteRow
Describe*	Filter	FilteredCount	Find*
FindRequired	GetChanges	GetChild	GetClickedColumn
GetClickedRow*	GetColumn	GetColumnName	GetFullState
GetItemDate	GetItemDateTime	GetItemDecimal*	GetItemNumber
GetItemStatus	GetItemString	GetItemTime	GetNextModified
GetParent	GetRow	GetSelectedRow	GetSQLPreview
GetSQLSelect	GetText	GetValue*	Hide
InsertRow	IsSelected	ModifiedCount	Modify*
Move	PostEvent	Print*	Reset*
ResetUpdate	Resize	Retrieve*	RowCount
RowsCopy*	RowsDiscard	RowsMove	SaveAs*

ScrollNextPage	ScrollNextRow	ScrollPriorPage	ScrollPriorRow
ScrollToRow	SelectRow	SetChanges	SetColumn
SetFilter*	SetFocus	SetFullState	SetItem
SetItemStatus	SetPosition*	SetRow	SetSort*
SetSQLPreview	SetSQLSelect*	SetTabOrder	SetText
SetTransObject	SetValue*	ShareData	ShareDataOff
Show	Sort*	TriggerEvent*	TypeOf
Update*			

The functions with superscript * are not fully supported (see below).

- The Describe and Modify functions do not support updating multiple properties at the same time.

- Describe
  Syntax: string dwcontrol.Describe (string propertylist)
  Describe reports the values of properties of a DataWindow object and controls within the DataWindow object. If there is no value for a property, Describe returns a question mark (?) in PowerBuilder. However, in the Web application, it returns the default value (for example, “Arrow!).
  The Evaluate function cannot be used within the Describe function.

- Find
  Supported example: long dwcontrol.Find (string expression, long start, long end)
  Note: if the *expression* argument is a string and is followed by space(s), the Find function cannot be executed successfully.

- GetChild
  In the earlier versions of PowerBuilder, the GetChild function can be used in a DataWindow for referring to a child DataWindow of the DataWindow. PowerBuilder 8.0 and above, as well as Appeon does not support this usage.

- GetItemDecimal
  In the Web application, if a data is a decimal, the method returns a string, omitting the zero(s) to the right of the decimal point. For example: 3000.000 will be returned as 3000 after conversion.

- GetValue
  string dwcontrol.GetValue (string *column*, integer index)
  string dwcontrol.GetValue (integer *column*, integer index)
  Column is a column number or a column name. The edit style of the column can be Dropdown ListBox or Radio Button only.

- Modify
  1) It is unsupported to dynamically modify a DataWindow property expression, or change the value of a property to a property expression.

  2) 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 will not validate the statement whereas JavaScript will.
  Supported syntax: string dwcontrol.Modify (string modstring)
  Note that, in a Web application, the following two types of statements cannot be used in modstring: a)

CREATE control (settings); b) DESTROY [COLUMN] control.

3) The style of quotes used within the string argument can be single quotes nested with double quotes or double quotes nested with single quotes, but it cannot be single quotes nested with single quotes or double quotes nested with double quotes.
Unsupported examples:
dw_report.Modify("datawindow.table.select=~""+sz_sql+"~", datawindow.color = ...")
dw_report.Modify('datawindow.table.select = ~'select * from employee where ~'employee~'.~'emp_id~' > 500 ~' datawindow.color=9999 ')

However, the only exception is datawindow.table.select, but you can modify only one property at a time.
Supported examples:
dw_report.Modify("datawindow.table.select=~""+sz_sql+"~"")
dw_report.Modify("datawindow.table.select = ~"select * from employee where ~"employee~".~"emp_id~" > 500 ~"")
dw_report.Modify('datawindow.table.select = ~'select * from employee where ~'employee~'.~'emp_id~' > 500 ~' ')

Unsupported examples:
dw_report.Modify("datawindow.table.select=~""+sz_sql+"~", datawindow.color = 9999")
dw_report.Modify('datawindow.table.select = ~'select * from employee where ~'employee~'.~'emp_id~' > 500 ~' datawindow.color=9999 ')

In addition, for datawindow.table.select, there are no single quotes nested with double quotes or double quotes nested with single quotes in the string argument, you can modify more than one property at a time.
Supported example:
dw_report.Modify("datawindow.table.select='select * from employee where emp_id=3' datawindow.color = 9999")

- Print
  Arguments specified in the Print function will be ignored in a Web application. For example, even if the *canceldialog* argument is set to TRUE or FALSE, no Cancel dialog will be displayed on the Web to cancel printing. Instead, a Print Setting dialog pops up.

- Reset
  The following syntax is not supported: Reset(datawindow). Use dwcontrol.Reset.

- Retrieve
  Because Appeon Web applications cannot check whether the retrieve result set matches the DataWindow definition syntax, the DBError event will not be triggered.
  If the data type of an EditMask control is numeric, the value of the control cannot be set to “0”.

- SetPosition
  Supported: Specify the front-to-back position of a DataWindow control in a window. The position can be ToTop!, ToBottom!, and cannot be Behind!
  Syntax: *dwcontrol*.SetPosition (*position {, precedingobject* })
  Unsupported: Move an object in a DataWindow to another band or to specify its front-to-back position within a band
  Syntax: dwcontrol.SetPosition (objectname, band, bringtofront)

- SetSort
  Support multi-column sorting with space as the delimiter.

- SetSQLSelect
  If a DataWindow uses a stored procedure as its data source, the SetSQLSelect function cannot be executed.

- SetValue
  number dwcontrol.SetValue (string *column,* number index, string value)
  number dwcontrol.SetValue (number *column*, number index, string value)
  Column is a column number or a column name. The edit style of the column can be DropdownListBox only.

- SaveAs
  The SaveAs type in SaveAs function can be TEXT, HTMLTable, WMF or EXCEL only.
  The supported SaveAs syntax:
  ll_testvalue = dw_test.SaveAs(filename,saveastype, colheading)
  The unsupported SaveAs syntax:
  dw_test.SaveAs([filename,]graphcontrol[, saveastype, colheading])

  The *filename* argument does not take any effect. Instead, the Web DataWindow will prompt a SaveAs dialogue box for the user to specify the file path and file name.
  For a list of supported value of the *saveastype* argument, please see SaveAsType in DataWindowConstants.
  The *colheading* argument does not take any effect.

- ShareData and ShareDataOff
  Example:
  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**()
  Note:
  1. The primary or the secondary DataWindow cannot be a Graphical DataWindow, but can be the following styles: Freedom, Grid, and Tabular.
  2. If a Computed field in the secondary DataWindow control has data which is in the primary DataWindow control, the name of the Computed field control in the secondary DataWindow must be the same as that in the primary DataWindow.

- Sort
  The sort function is case-insensitive.
  Different from the Sort in PowerBuilder, the Web Sort function considers that the "~" symbol has precedence over characters. For this reason, the sort result may be a little different in the Web and PowerBuilder applications.
  To specify the sort criteria, the user drags and drops items in the built-in dialog box on the Rows menu of the DataWindow painter. However, in the Web application, the user double clicks the column items instead.
  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.
  In a Web application, the Sort function will not sort the rows that are a Computed Field.

- TriggerEvent
  Events cannot be triggered by moving data within the primary buffer of one DataWindow.
  The following TriggerEvent syntax is unsupported: *object1.*TriggerEvent (*object2, event).*

- SetFilter
  If the SetFilter criteria is incorrect, on the Web, there will be no relevant error message box.
  To specify their own filter expression for a DataWindow control, the users can select a function from the Functions field and paste it into the expression in the Specify Filter dialog box. The argument is selected so that it can be replaced with a column name. Also, the users can select a column and paste it into the expression at the cursor. However, in the Web application, the function or column selected

will be always added behind the expression.

- SetSort
  If the Format argument of the SetSort contains sorting criteria for more than one columns, the criteria of the columns must be separated with commas.

- Update
  Example: ll_testvalue = dw_test.Update(accept, resetflag)

- GetClickedRow
  In the Web application, if the user clicks or double-clicks any area within the DataWindow, the GetClickedRow function returns the same value as in PowerBuilder. If the GetClickedRow is called in some control or window, the return value is different on the Web from in PowerBuilder. For example, if the function is called in the Clicked event of a CommandButton, dw_1.GetClickedRow(), it returns –1 in PowerBuilder and 0 in the Web application.

Unsupported

1) The following DataWindow functions are unsupported:

CanUnDo	CategoryCount	CategoryName	Clear
ClearValues	Clipboard	Copy	CopyRTF
Create	CrosstabDialog	Cut	DataCount
DBCancel	DBErrorCode	DBErrorMessage	Drag
FindCategory	FindGroupChange	FindNext	FindSeries
GenerateHTMLForm	GenerateResultSet	GetBandAtPointer	GetBorderStyle
GetContextService	GetData	GetDataPieExplode	GetDataStyle
GetDateValue	GetFormat	GetMessageText	GetObjectAtPointer
GetRowFromRowId	GetRowIdFromRow	GetSeriesStyle	GetStateStatus
GetTrans	GetUpdateStatus	GetValidate	GroupCalc
ImportClipboard	ImportFile	ImportString	InsertDocument
LineCount	ObjectAtPointer	OLEActivate	Paste
PasteRTF	PointerX	PointerY	Position
PrintCancel	ReplaceText	ReselectRow	ResetDataColors
ResetTransObject	SaveAsAscii	Scroll	SelectedLength

SelectedLine	SelectedStart	SelectedText	SelectText
SelectTextAll	SelectTextLine	SelectTextWord	SeriesCount
SeriesName	SetActionCode	SetBorderStyle	SetDataPieExplode
SetDataStyle	SetDetailHeight	SetFormat	SetHTMLAction
SetPosition	SetRedraw	SetRowFocusIndicator	SetSeriesStyle
SetTrans	SetValidate	ShowHeadFoot	TextLine
Undo			

DataStore object

Properties | Events | Functions

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.

Properties of DataStore object

Properties | Events | Functions

Supported

The supported properties of DataStore are listed in the table below:

DataObject	Object

Unsupported

The unsupported property of DataStore is ClassDefinition.

Events of DataStore object

Properties | Events | Functions

Supported

The following DataStore events are supported:

Constructor	DBError	Destructor*	ItemChanged
ItemError	RetrieveEnd	RetrieveStart	SQLPreview*
UpdateEnd	UpdateStart		

Notes:
1) 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.
2) The SQLPreview event can be triggered by Retrieve method, and cannot be triggered by Update or ReselectRow method.

Unsupported

The following DataStore events are unsupported:

Error	PrintEnd	PrintPage	PrintStart
RetrieveRow			

Functions of DataStore object

Properties | Events | Functions

Supported

The following DataStore functions are supported:

AcceptText	ClassName	DeletedCount	DeleteRow
Describe*	Filter	FilteredCount	Find
GetChild	GetColumn	GetChanges	GetColumnName
GetFullState	GetItemDate	GetItemDateTime	GetItemDecimal
GetItemNumber	GetItemStatus	GetItemString	GetItemTime
GetNextModified	GetParent	GetRow	GetSelectedRow
GetSQLSelect	GetText	GetValue*	InsertRow
IsSelected	Modify*	ModifiedCount	PostEvent
Print	Reset	ResetUpdate	Retrieve
RowCount	RowsCopy	RowsDiscard	RowsMove*
SaveAs*	SelectRow	SetChanges	SetColumn
SetFilter	SetFullState	SetItem	SetItemStatus
SetRow	SetSort	SetSQLPreview	SetSQLSelect

SetText	SetTransObject	SetValue*	ShareData
ShareDataOff	Sort*	TriggerEvent*	TypeOf
Update			

The functions with superscript * are not fully supported (see below).

- The Describe and Modify functions do not support updating multiple properties at the same time.
- If the value of a property in the propertylist argument is null, in PowerBuilder, the value will be represented by "?" in the return value, and the values of the subsequent properties can be returned. In Appeon, the null value will be represented by "!" and the values of the subsequent properties cannot be returned.
- GetValue

  String dwcontrol.GetValue (string *column*, integer index)
  string dwcontrol.GetValue (integer *column*, integer index)
  Column is a column number or a column name. The edit style of the column can be Dropdown ListBox or Radio Button only.

- 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.
- SetValue

  number dwcontrol.SetValue (string *column*, number index, string value)
  number dwcontrol.SetValue (number *column*, number index, string value)
  Column is a column number or a column name. The edit style of the column can be DropdownListBox only.

- SaveAs
  The SaveAs type in SaveAs function can be TEXT, WMF or EXCEL only.
  The supported SaveAs syntax:
  ll_testvalue = dw_test.SaveAs(filename,saveastype, colheading)
  The unsupported SaveAs syntax:
  dw_test.SaveAs([filename,]graphcontrol[, saveastype, colheading])

- ShareData and ShareDataOff
  Example:
  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**()
  Notes:
  1. The primary or the secondary DataWindow cannot be a Graphical DataWindow, but can be the following styles: Freeform, Grid, and Tabular.
  2. If a Computed field in the secondary DataWindow control has data which is in the primary

DataWindow control, the name of the Computed field control in the secondary DataWindow must be the same as that in the primary DataWindow.

- Sort

  The sort function is case-insensitive.

- RowsMove

  Events cannot be triggered by moving data within the primary buffer of one DataWindow.

- TriggerEvent
  The following TriggerEvent syntax is unsupported: *object1.*TriggerEvent (*object2, event).*

Unsupported

The following DataStore functions are unsupported:

CategoryCount	CategoryName	ClearValues	Clipboard
CopyRTF	Create	CreateFrom	DataCount
DBCancel	FindCategory	FindGroupChange	FindRequired
FindSeries	GenerateHTMLForm	GenerateResultSet	GetBorderStyle
GetClickedColumn	GetClickedRow	GetContextService	GetData
GetDataPieExplode	GetDataStyle	GetDataValue	GetFormat
GetRowFromRowId	GetRowIdFromRow	GetSeriesStyle	GetStateStatus
GetTrans	GetValidate	GroupCalc	ImportClipboard
ImportFile	ImportString	InsertDocument	PasteRTF
PrintCancel	ReselectRow	Reset(graphs)	ResetDataColors
ResetTransObject	SeriesCount	SetBorderStyle	SeriesName
SetDataPieExplode	SetDataStyle	SetDetailHeight	SetFormat
SetHTMLAction	SetPosition	SetSeriesStyle	SetTrans
SetValidate			

DataWindowChild object

Properties of DataWindowChild object

Properties | Functions

APPEON

Supported

None.

Unsupported

The unsupported property of DataWindowChild is ClassDefinition.

Functions of DataWindowChild object

Properties | Functions

APPEON

Important Requirements

DataWindowChild object cannot be dynamically created (e.g. using the CREATE statement); it must be defined in the PowerBuilder painter

Supported

The following DataWindowChild functions are supported:

AcceptText	ClassName	DeletedCount	DeleteRow
Describe*	Filter	FilteredCount	Find
FindRequired	GetChanges	GetChild	GetClickedColumn
GetClickedRow	GetColumn	GetColumnName	
GetItemDate	GetItemDateTime	GetItemDecimal	GetItemNumber
GetItemStatus	GetItemString	GetItemTime	GetNextModified
GetParent	GetRow	GetSelectedRow	GetSQLPreview
GetSQLSelect	GetText	GetValue*	InsertRow
IsSelected	ModifiedCount	Modify*	Print
Reset	ResetUpdate	Retrieve *	RowCount
RowsCopy	RowsDiscard	RowsMove *	SaveAs*

ScrollNextPage	ScrollNextRow	ScrollPriorPage	ScrollPriorRow
ScrollToRow	SelectRow	SetChanges	SetColumn
SetFilter	SetItem	SetItemStatus	SetPosition*
SetRow	SetSort	SetSQLPreview	SetSQLSelect
SetTabOrder	SetText	SetTransObject	SetValue*
Sort	TypeOf	Update	

The functions with superscript * are not fully supported (see below).

- Describe and Modify
  The Describe and Modify functions do not support updating multiple properties at the same time.
  If the value of a property in the propertylist argument is null, in PowerBuilder, the value will be represented by "?" in the return value, and the values of the subsequent properties can be returned. In Appeon, the null value will be represented by "!" and the values of the subsequent properties cannot be returned.

- GetValue

  string dwcontrol.GetValue (string *column*, integer index)
  string dwcontrol.GetValue (integer *column*, integer index)
  Column is a column number or a column name. The edit style of the column can be Dropdown ListBox or Radio Button only.

- 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.

- SetPosition
  Supported: Specify the front-to-back position of a DataWindow control in a window. The position can be ToTop!, ToBottom!, and cannot be Behind!.
  Syntax: *dwcontrol*.SetPosition (*position {, precedingobject* })
  Unsupported: Move an object in a DataWindow to another band or to specify its front-to-back position within a band
  Syntax: dwcontrol.SetPosition (objectname, band, bringtofront)

- SetValue

  number dwcontrol.SetValue (string *column*, number index, string value)
  number dwcontrol.SetValue (number *column*, number index, string value)
  Column is a column number or a column name. The edit style of the column can be DropdownListBox only.

- SaveAs
  The SaveAs type in SaveAs function can be TEXT, WMF or EXCEL only.
  The supported SaveAs syntax:

ll_testvalue = dw_test.SaveAs(filename,saveastype, colheading)
The unsupported SaveAs syntax:
dw_test.SaveAs([filename,]graphcontrol[, saveastype, colheading])

- Sort

  The sort function is case-insensitive.

- RowsMove

  Events cannot be triggered by moving data within the primary buffer of one DataWindow.

Unsupported

The following DataWindowChild functions are unsupported:

CanUnDo	CategoryCount	CategoryName	Clear
ClearValues	Clipboard	Copy	CopyRTF
Create	CrosstabDialog	Cut	DataCount
DBCancel	DBErrorCode	DBErrorMessage	Drag
FindCategory	FindGroupChange	FindNext	FindSeries
GenerateHTMLForm	GenerateResultSet	GetBandAtPointer	GetBorderStyle
GetContextService	GetData	GetDataPieExplode	GetDataStyle
GetDateValue	GetFormat	GetMessageText	GetObjectAtPointer
GetRowFromRowId	GetRowIdFromRow	GetSeriesStyle	GetStateStatus
GetTrans	GetUpdateStatus	GetValidate	GroupCalc
ImportClipboard	ImportFile	ImportString	InsertDocument
LineCount	ObjectAtPointer	OLEActivate	Paste
PasteRTF	PointerX	PointerY	Position
PrintCancel	ReplaceText	ReselectRow	ResetDataColors
ResetTransObject	SaveAsAscii	Scroll	SelectedLength
SelectedLine	SelectedStart	SelectedText	SelectText
SelectTextAll	SelectTextLine	SelectTextWord	SeriesCount
SeriesName	SetActionCode	SetBorderStyle	SetDataPieExplode
SetDataStyle	SetDetailHeight	SetFormat	SetHTMLAction
SetPosition	SetRedraw	SetRowFocusIndicator	SetSeriesStyle

SetTrans	SetValidate	ShareData	ShareDataOff
ShowHeadFoot	TextLine	Undo	

DataWindow performance considerations

APPEON

Reduce the number of columns in DataWindow

It takes time to initialize columns and controls in a DataWindow. Moreover, the more columns there are in a DataWindow, the slower it is to retrieve and display data. Therefore, Appeon recommends the following:

1) Remove unnecessary columns and controls from DataWindow.

2) If the Visible property of a column is set to zero (the control is not visible), and the column is not used at all, it is best to remove it to reduce the DataWindow size.

Avoid placing DataWindow functions that refresh the UI in a loop

Two types of DataWindow functions cause the Web UI to refresh: those that are directly UI-related and those that operate on data but ultimately lead to redrawing the UI. If these types of DataWindow functions are put into a loop and repeated many times, you will see that the Web DataWindow keep flashing, which is very annoying. For instance, if you use a loop to copy 100 rows of data into a DataWindow one row by another, the Web UI will flash each time a row is added. So 100 flashes would take place before the operation completed.

The following DataWindow functions will result in redrawing or flashing of the UI: Retrieve(), SetItem(), InsertRow(), DeleteRow(), RowsCopy(), RowsMove(), RowsDiscard(), Reset(), operations on DataWindow data using Dot Notation.

Appeon suggests you avoid using these the above functions in a loop.

Avoid computed fields

Computed columns involve a lot of recalculation in many situations. For example, when a column is deleted, added, or renamed. This recalculation contains is process intensive which negatively impacts performance and it can be worked around. Therefore, Appeon recommends the following:

1) Avoid using computed columns. Instead, add expressions in the SQL statements for getting specific data.

2) If a computed column is "*Text: Sum or Expression*", it is recommended to divide the column into two columns: an edit style column with the "*Text*" and a computed column with "*Sum or Expression*".

Rework DropDownDataWindow and reduce EditMask edit styles

There are namely two DataWindow edit styles that are significantly slower than other DataWindow Edit styles. They are DropDownDataWindow and EditMask. For example, DropDownDataWindow is generally 40% slower than DropDownListBox to display data. Therefore, Appeon suggests the following:

1) Avoid using EditMask and DropDownDataWindow. Use other DataWindow edit styles such as Radiobuttons, Edit, and DropDownListBox.

2) If a DropDownDataWindow is only for displaying the data (without requiring the user to select or input a value), you can replace it with the Edit edit style and use SQL statements to achieve the same functionality (from the end-user perspective). Best of all you will not be degrading performance.

Reduce usage of DataWindow expressions

Generally speaking, when doing Retrieve operation, the execution time of the DataWindow without property expressions is at least twice as fast. Therefore, Appeon recommends you reduce usage of DataWindow expressions and especially in the following situations:

1) Avoid using DataWindow expressions for computing and setting column property.

2) Avoid setting sort and filter criteria directly for a DataWindow object. Instead, write the sort and filter criteria in the SQL statement.

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.

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.

DataStore preferred over DataWindow for non-visual data processing

Reasons: A DataStore is a non-visual DataWindow control. When you perform operations on DataStore, the Web application does not need to do any UI work, so it is faster than a DataWindow. DataStore is more than two times faster than DataWindow in performing data retrieval. Therefore, Appeon recommends you use DataStores instead of DataWindows for non-visual data processing whenever possible.

Use SetRow instead of ScrollToRow

The DataWindow SetRow and ScrollToRow functions perform similar functionalities: they both change the current row in DataWindow. However, ScrollToRow also scrolls a DataWindow to the specified row, which is UI-related and costs more time. Therefore, Appeon recommends you use SetRow instead of ScrollToRow to locate the specified row.

The following is an example:

```
dw_employee.ScrollToRow(10)
```

⇓

```
dw_employee.SetRow(10)
```

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.

Minimize code in RowsFocusChanging & RowsFocusChanged

The DataWindow RowsFocusChanging and RowsFocusChanged events can be triggered under many situations, especially when a DataWindow retrieves data. Since usually data is automatically retrieved into DataWindows when a window is opened, if a lot of code is written into the RowsFocusChanging and RowsFocusChanges events, it will significantly prolong the time to open the window. Therefore, Appeon recommends you do not write code into RowsFocusChanging and RowsFocusChanged events unless necessary.

Undetected Features

APPEON

Overview

Appeon provides certain measures to detect unsupported features in a PowerBuilder application before parsing the PBL source code, generating corresponding Web files, and deploying them to the Web. The Unsupported Features Analysis scans each object within a PowerBuilder application for unsupported features and generates a report that, when combined with the Migration User Guide and Appeon 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 will not list in the Analysis Report. Approximately 5% of PowerBuilder features that are not supported by Appeon may not be detected in the Unsupported Features Analysis.

APPEON

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.

Undetected Features Listing

The following are unsupported features that are not detected during Unsupported Features Analysis.

Naming Conventions
The Unsupported Features Analysis cannot detect the following issues which can cause problems in the converted Web application:

- 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. It is unsupported to have more than one Application object in an application. 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 of PowerBuilder control/object types are not supported. Unsupported Features Analysis cannot detect whether this unsupported issue is present in an application.

Null values

- Unsupported operation
- Expressions with Null value

Special ASCII characters
Special ASCII characters that are not supported cannot be detected in Unsupported Features Analysis:

- Common ASCII characters: Vertical tab (~v)

Enumerated data type

- Default values of enumerate type variable

Variables and constants

- Instance variables having identical names as global variables.
- Unsupported property or method of a supported variable type.

Arrays

- Assigning char arrays to strings.
- Having the return value of a variable or function as an element in an arraylist.

Forced conversion

- Forced conversion between types.

Window

- Multiple MDI windows in an application.

Non-visual UserObject

- The Constructor event of non-visual UserObject.
- Assign a NonVisualObject object to an autoinstantiated NVO or assign an autoinstantiated NVO to a NonVisualObject object.

Overloading, overriding, and extending functions and events

- Unsupported overloading cases (calling to the function of the ancestor object)
- Use of the local variable AncestorReturnValue in an event of a descendent object, except when the event of the descendent object is an extended event from the ancestor object.

Event triggering sequence

- The event triggering sequence on the Web may be different from the sequence in the PowerBuilder application.

Static and dynamic calls

- Unsupported overloading cases (dynamic call)

Using the return value of some supported functions

- The return value of the Open or OpenSheet functions
- Assigning the today() or time() to a date or time variable

User functions

- Remote procedure call (RPC)

Operators

- The operator '^' with embedded SQL statements.
- The order of precedence for operators.

Stored procedure

- DB2 stored procedures
- Stored procedures placed inside a package

- Oracle stored procedures with Appeon unsupported features

Cursor declare requirement

- 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.

- Unsupported Features Analysis cannot detect whether this requirement is fulfilled. See Cursordeclarerequirement in the Basic and Architectural Requirements section for more details.

Cursor statements

- Global and instance cursors
- The following two syntaxes:
    - UPDATE TableName SetStatement WHERE CURRENT OF CursorName;
    - DELETE FROM TableName WHERE CURRENT OF CursorName;

DataWindow presentation styles

- OLE DataWindow, RichText DataWindow

DataWindow operators and expressions

- DataWindow expressions if they are not supported

DataWindow data expressions

- Appeon UFA (unsupported features analysis) tool cannot detect the unsupported features in the DataWindow dot notations. The user must make sure the specifications about dot notations are satisfied in the application. Otherwise, it will cause errors.

Functions of DataWindow control

- Some partly supported features (such as DataWindow Modify and Describe functions).

Events of DataWindow and DataStore

- The RetrieveRow event cannot be detected by Unsupported Features Analysis, though RetrieveRow is not supported.

User interface interactions

- Unsupported features in the UI such as shortcut key.