

Workarounds & API Guide

Appeon® for PowerBuilder® 2013 R2
FOR WINDOWS & UNIX & LINUX

DOCUMENT ID: ADC20236-01-0680-01

LAST REVISED: February 11, 2014

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

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

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

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

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

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

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

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

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

Contents

1 About This Book	1
1.1 Audience	1
1.2 How to use this book	1
1.3 Related documents	1
1.4 If you need help	3
2 Appeon Workarounds PBL Reference	4
2.1 Introduction to Appeon Workarounds	4
2.2 Best practices for calling APIs	4
2.3 General API	4
2.3.1 AppeonExtFuncs object	5
2.3.1.1 of_getappeonusername function	7
2.3.1.2 of_getbrowserversion function	7
2.3.1.3 of_getcachedir function	8
2.3.1.4 of_getclientid function	8
2.3.1.5 of_getclientip function	9
2.3.1.6 of_getclienttype function	9
2.3.1.7 of_gethttpinfo function	9
2.3.1.8 of_getiehandle function	10
2.3.1.9 of_getieurl function	10
2.3.1.10 of_getostype function	10
2.3.1.11 of_getrunmode function	11
2.3.1.12 of_getservertype function	11
2.3.1.13 of_getsessioncount function	12
2.3.1.14 of_is64browser function	12
2.3.1.15 of_ldaplogon function	12
2.3.1.16 of_log function	13
2.3.1.17 of_clearlog function	14
2.3.1.18 of_popmenu function	14
2.3.1.19 of_popmenuon function	15
2.3.1.20 of_print2file function	15
2.3.1.21 of_Print2PDF function (Obsolete)	17
2.3.1.22 of_switchRealTimeCalc function	17
2.3.2 Appeon Client Functions	17
2.3.2.1 AppeonGetAppeonUserName function	17
2.3.2.2 AppeonGetBrowserVersion function	18
2.3.2.3 AppeonGetCacheDir function	18
2.3.2.4 AppeonGetClientID function	19
2.3.2.5 AppeonGetClientIP function	19
2.3.2.6 AppeonGetClientType function	19
2.3.2.7 AppeonGetHttpInfo function	19
2.3.2.8 AppeonGetIEHandle function	20
2.3.2.9 AppeonGetIEURL function	20
2.3.2.10 AppeonGetOSType function	20
2.3.2.11 AppeonGetRemainingdays function	21
2.3.2.12 AppeonGetServerType function	21
2.3.2.13 AppeonGetSessionCount function	22

2.3.2.14	appeonisin64browser function	22
2.3.2.15	AppeonLDAPLogon function	22
2.3.2.16	AppeonPopMenu function	22
2.3.2.17	AppeonPopMenuOn function	23
2.3.2.18	AppeonPrint2File function	24
2.3.2.19	AppeonPrint2PDF function (Obsolete)	25
2.3.2.20	AppeonSwitchRealTimeCalc function	25
2.3.3	Client Logs	26
2.3.3.1	eon_logex object	26
2.3.4	File Upload and Download	26
2.3.4.1	Configuring and deploying Appeon File Server	26
2.3.4.2	Uploading and downloading files	43
2.3.5	Distributed DataWindows (EAServer only)	48
2.3.5.1	AppeonDataWindow and AppeonDataStore	48
2.3.5.2	Workaround steps	50
2.3.5.3	Workaround limitations	50
2.3.6	Appeon Labels	51
2.3.6.1	Appeon Commit/Rollback Label	52
2.3.6.2	Appeon Commit Label	53
2.3.6.3	Appeon Rollback Label	53
2.3.6.4	Appeon Queue Labels	54
2.3.6.5	Appeon Immediate Call Label	56
2.3.6.6	Appeon Update Label	57
2.3.7	Calling EJB Component	59
2.3.7.1	Appeon Bridge	59
2.3.7.2	EJBObject object	60
2.3.7.3	Appeon requirements for EJB development	67
2.3.8	Calling .NET/COM server components (.NET only)	68
2.3.8.1	AppeonDotNetComponent object	69
2.3.9	Calling Web Service	73
2.3.9.1	appeonwebservicecomponent object	73
2.4	Mobile Device API	77
2.4.1	Appeon Workspace	77
2.4.1.1	eon_mobile_awsex object	77
2.4.2	Barcode	91
2.4.2.1	eon_mobile_barcodeex object	91
2.4.3	Camera	96
2.4.3.1	eon_mobile_cameraex object	96
2.4.4	Connection	100
2.4.4.1	eon_mobile_connectionex object	101
2.4.5	Device	101
2.4.5.1	eon_mobile_deviceex object	101
2.4.6	Geolocation	108
2.4.6.1	eon_mobile_geolocationex object	108
2.4.7	Map	115
2.4.7.1	eon_mobile_mapex object	116
2.4.8	Media	125
2.4.8.1	eon_mobile_mediaex object	125

2.4.9 Notification	126
2.4.9.1 eon_mobile_notificationex object	126
2.4.10 Offline	127
2.4.10.1 eon_offlineex object	128
2.4.11 Textchecker	131
2.4.11.1 eon_mobile_textcheckex object	131
2.4.12 NFC	136
2.4.12.1 eon_mobile_nfcex object	137
3 Apeon Resize PBL Reference	149
3.1 eon_cst_resize object	149
3.1.1 Functions	149
3.1.1.1 of_checkSizeType	149
3.1.1.2 of_getComponentName	149
3.1.1.3 of_init	149
3.1.1.4 of_init (userobject)	150
3.1.1.5 of_init (window)	150
3.1.1.6 of_register	150
3.1.1.7 of_setResizeType	151
3.1.1.8 of_typeOf	151
3.1.2 Properties	151
3.2 eon_cst_resizeattrib object	152
3.3 eon_cst_resizecompose object	152
3.3.1 Functions	152
3.3.1.1 of_init	152
3.3.1.2 of_register (a_obj, as_method)	152
3.3.1.3 of_register (a_component, a_obj, as_method)	152
3.3.2 Properties	153
3.3.3 Event	153
3.3.3.1 pfc_resize	153
4 Workarounds for Unsupported Features	154
4.1 Objects & Controls	154
4.1.1 External user object	154
4.1.2 UserObject object	154
4.1.2.1 AddItem function (EAServer only)	154
4.1.2.2 DeleteItem function	154
4.1.3 System Objects	154
4.1.3.1 PipeLine object	154
4.1.3.2 Application object	155
4.1.4 System Controls	155
4.1.4.1 ListView control	155
4.1.4.2 TreeView control	156
4.1.4.3 Tab control	156
4.2 System Functions	156
4.2.1 DDE Server functions	156
4.2.2 FileLength64 function	156
4.2.3 FileReadEx function	157
4.2.4 FileSeek64 function	157
4.2.5 FileWriteEx function	157

4.2.6	FromAnsi function	157
4.2.7	FromUnicode function	157
4.2.8	Garbage Collection functions	158
4.2.9	GetLibraryList function	158
4.2.10	Help functions	158
4.2.11	HyperLinkToURL function	159
4.2.12	LibraryDirectory function	159
4.2.13	PopulateError function	160
4.2.14	Registry function	160
4.2.15	Shared Object functions	160
4.2.16	SignalError function	160
4.3	PowerScript Reference	161
4.3.1	PowerScript Topics	161
4.3.1.1	Calling functions and events	161
4.3.1.2	Declarations	162
4.3.1.3	Language basics	163
4.3.2	PowerScript Statements	164
4.3.2.1	GOTO statement	164
4.3.3	SQL Statements	165
4.3.3.1	Stored procedure with Null output	165
4.3.4	PowerScript Events	165
4.3.4.1	Help event	165
4.3.4.2	Other event	166
4.4	DataWindow Reference	166
4.4.1	DataWindow operators	166
4.4.1.1	DataWindow operator precedence	166
4.4.2	DataWindow Object Properties	166
4.4.2.1	Retrieve.AsNeeded	166
4.4.3	DataWindow control	167
4.4.3.1	Functions	167
4.4.3.2	Events	170
4.4.3.3	Properties	172
4.4.4	Controls in a DataWindow	172
4.4.4.1	Large Binary/Text database OLE object	172
4.4.4.2	Column control	173
4.4.5	Functions of DataStore object	174
4.5	DBParm parameters in Database	174
4.5.1	ConnectionString parameter	174
4.6	Workaround Techniques (for EAServer only)	174
4.6.1	Appeon GetFullState/SetFullState/GetChanges/SetChanges (Windows only)	174
4.6.2	Packaging unsupported features in n-Tier NVOs or DLLs	177
4.6.2.1	Packaging unsupported objects into n-Tier NVOs	177
4.6.2.2	Placing unsupported functions into n-Tier NVOs	179
5	FAQ & Workarounds	182
5.1	How to remove the Internet Explorer menu	182
5.2	How to deploy NVO to EAServer 6.1	183
5.3	How to deploy NVOs to EAServer without PowerBuilder	194

5.4 How to use Client resources in Appeon Web applications	194
5.5 How to replace Appeon image that displays at the running of applications	195
5.6 How to deploy an Appeon application without Appeon Developer	196
5.7 How to log in the Web application with single sign-on	196
5.8 How to add headers & footers to a Web application	197
5.9 How to get the user name and password of the operating system	197
5.10 How to modify the storage location of Web application files in Appeon Developer	197
5.11 How to integrate Appeon Web applications with JSP/ASP	198
5.11.1 Applying Appeon CommandParm and Hyperlink features	198
5.11.2 Using Internet Explorer Frame	198
5.11.3 Integration through intermediate n-Tier Server-level solutions.....	199
5.12 No Workarounds	199
Index	200

1 About This Book

1.1 Audience

This book is for developers who want to use APIs to implement various functionalities in the Web or mobile applications and get solutions for issues encountered when using Appeon Web or Appeon Mobile.

1.2 How to use this book

There are five chapters in this book.

Chapter 1: About This Book

A general description of this book.

Chapter 2: Appeon Workarounds PBL Reference

Lists the syntax and code examples of the programming interfaces provided by the Appeon Workarounds PBL, to implement corresponding functionalities.

Chapter 3: Appeon Resize PBL Reference

Lists the syntax and code examples of the programming interfaces provided by the Appeon Resize PBL, to resize and move the controls within a window.

Chapter 4: Workarounds for Unsupported Features

Provides suggestions to work around the unsupported features that have functional impact on the running of the application.

Chapter 5: FAQ and Workarounds

Lists some frequently asked questions and workaround tips regarding the Appeon application architecture or product features.

1.3 Related documents

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

- Introduction to Appeon:
Gives general introduction to Appeon for PowerBuilder and its editions.
- Getting Started (for Appeon Mobile):
Guides you through installing PowerBuilder and Appeon for PowerBuilder, and developing and deploying a mobile application.
- New Features Guide:
Introduces new features and changes in Appeon for PowerBuilder.
- Appeon Mobile Tutorials:
Gives instructions on deploying, running, and debugging the mobile application, distributing native mobile apps, and configuring the Appeon Server cluster.

- **Appeon Mobile (Offline) Tutorials:**
Gives instructions on setting up the Appeon Mobile (Offline) environment, and configuring, deploying, running, updating, and debugging the offline application.
- **Appeon Installation Guide:**
Provides instructions on how to install Appeon for PowerBuilder successfully.
- **Installation Guide on Cloud Platform:**
Provides instructions on how to install Appeon for PowerBuilder on the cloud-based platform such as Windows Azure and AWS EC2 and S3.
- **Mobile UI Design & Development Guide:**
Introduces general guidelines on designing and developing the mobile app and UI.
- **Migration Guidelines for Appeon Web:**
A process-oriented guide that illustrates the complete diagram of the Appeon Web migration procedure and various topics related to steps in the procedure, and includes a tutorial that walks you through the entire process of deploying a small PowerBuilder application to the Web.
- **Supported PB Features:**
Provides a detailed list of supported PowerBuilder features which can be converted to the Web/Mobile with Appeon as well as which features are unsupported.
- **Appeon Developer User Guide:**
Provides instructions on how to use the Appeon Developer toolbar in Appeon for PowerBuilder.
- **Workarounds & API Guide:**
Provides resolutions for unsupported features and various APIs to facilitate you to implement the features (including Web and mobile) that are not easy or impossible to implement in the PowerBuilder IDE.
- **Appeon Workspace User Guide:**
Gives a general introduction on Appeon Workspace and provides detailed instructions on how to use it.
- **Appeon Server Configuration Guide:**
Provides instructions on how to configure Appeon Server Monitor, establish connections between Appeon Servers and database servers, and configure AEM for maintaining Appeon Server and the deployed applications.
- **Web Server Configuration Guide:**
Describes configuration instructions for different types of Web servers to work with the Appeon Server.

- **Troubleshooting Guide:**
Provides information on troubleshooting issues; covering topics, such as product installation, application deployment, AEM, and Appeon application runtime issues.
- **Appeon Performance Tuning Guide:**
Provides instructions on how to modify a PowerBuilder application to achieve better performance from its corresponding Web/mobile application.
- **Testing Appeon Web Applications with QTP:**
Provides instructions on how to test Appeon Web applications with QTP.

1.4 If you need help

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

2 Appeon Workarounds PBL Reference

This chapter lists the syntax and code examples of the programming interfaces provided by the Appeon Workarounds PBL, to implement the various functionalities in Appeon Web application and Appeon mobile application.

2.1 Introduction to Appeon Workarounds

Appeon Workarounds provides a bunch of APIs and packaged solutions to facilitate you implement the features that are not easy or impossible to implement in the PowerBuilder IDE. Appeon Workarounds includes one PBL, three DLLs and one XML file to help to work around some PowerBuilder features and extend the usability of the deployed application.

- One PBL: `appeon_workarounds.pbl`
- Three DLLs: `EonAXNVO.dll`, `EonEmfPic.dll` and `Eonejbclient.dll`
- XML file: `ejb_err_info.xml`

They are located in the `\appeon_workaroundsxxx` (**xxx** indicates the PowerBuilder version) folder under the Appeon Developer installation directory. For example, `C:\Program Files\Appeon\Developer2013R2\appeon_workarounds125e`.

To use the APIs or solutions provided in the Appeon Workarounds PBL, you will need to add the `appeon_workarounds.pbl` library to the Library Search Path of the application.

2.2 Best practices for calling APIs

Not all of the APIs can be used in both the Mobile and Web environment. In [Section 2.3, “General API”](#), the following functions: [of_popmenu](#), [of_popmenuon](#), [of_print2file](#), [of_Print2PDF](#), [of_getbrowserversion](#), [of_getiehandle](#), [of_getieurl](#), & [of_is64browser](#) from the [AppeonExtFuncs object](#) are effective in the Web environment only. In [Section 2.4, “Mobile Device API”](#), all listed functions are effective in the Mobile environment. If a function that is effective in the Web environment only is executed in a Mobile environment, or if a function that is effective in the Mobile environment only is executed in a Web environment, the function may return unexpected values, and cause the application to perform abnormally.

Therefore, to avoid the aforementioned problem, you should detect the running environment first before calling the corresponding functions.

For example, to call the `of_setapprotationlock` API which take effects in the mobile environment only, you are recommended to write the scripts in this way:

```
if appeongetclienttype()="MOBILE" then
    eon_mobile_awsex lgnv_aws
    lgnv_aws = CREATE eon_mobile_awsex
    lgnv_aws.of_setAppOrientation(2)
    lgnv_aws.of_setapprotationlock(1)
    destroy lgnv_aws
end if
```

2.3 General API

The APIs listed in this section take effects in both the Web and Mobile environment, except for those specified otherwise.

2.3.1 AppeonExtFuncs object

AppeonExtfuncs object provides functions to help users manipulate the deployed application. The function of the AppeonExtfuncs object usually returns different values in the PowerBuilder application and in the deployed application. The values returned in the PowerBuilder application are defined in the AppeonExtfuncs object; while the values returned in the deployed application are defined in the Appeon client library.

The function of AppeonExtfuncs object performs almost identical to the global function of the [Appeon client functions](#). Typically, the function of AppeonExtfuncs object is recommended over the [Appeon client functions](#) for the following three reasons:

- The function of the AppeonExtfuncs object delivers a better extensibility and is much easier to manage.
- Calling too much global functions of [Appeon client functions](#) may affect the performance.

Calling functions of AppeonExtFuncs object

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 information (such as browser version, user name etc.) can be utilized to code more application features such as security authentication, auditing, logging, file operation, etc. This means you can write more scripts in NVO components for implementing more application features.

Below lists the function of AppeonExtfuncs object and their corresponding [Appeon client functions](#).

Table 2.1: Functions of AppeonExtFuncs object

Function of AppeonExtFuncs object	Appeon client functions	Description
of_getappeonusername function	AppeonGetAppeonUserName function	Gets the user name that you type into the Appeon Login dialog box.
of_getbrowserversion function	AppeonGetBrowserVersion function	Gets the Internet Explorer version of the client.
of_getcachedir function	AppeonGetCacheDir function	Gets the Cache directory that is used by the current Web application. Generally the cache directory is \Documents and Settings\username \Application Data\appeon \application_name.
of_getclientid function	AppeonGetClientID function	Gets the unique session identifier for the Internet Explorer client.
of_getclientip function	AppeonGetClientIP function	Gets the IP address of the Internet Explorer client.
of_getclienttype function	AppeonGetClientType function	Gets the type of an application.

Function of AppeonExtFuncs object	Appeon client functions	Description
of_gethttpinfo function	AppeonGetHttpInfo function	Gets the HTTP header information from a particular request.
of_getiehandle function	AppeonGetIEHandle function	Gets the Internet Explorer handle for the current application.
of_getieurl function	AppeonGetIEURL function	Gets the URL of the Web application.
of_getostype function	AppeonGetOSType function	Gets the type of OS that runs your application (the Appeon Web application, the Appeon mobile application, or the PowerBuilder client application).
NA	AppeonGetRemainingdays function	Gets the remaining day(s) of license or technical support.
of_getservertype function	AppeonGetServerType function	Gets the Appeon Server type where the application runs.
of_getsessioncount function (Obsolete)	AppeonGetSessionCount function (Obsolete)	of_getsessioncount function is obsolete since Appeon for PowerBuilder 2013.
of_is64browser function	appeonisin64browser function	Detects if the IE browser where the application runs is 64-bit.
of_ldaplogon function	AppeonLDAPLogon Function	Logs in to the LDAP server with the specified user name and password.
of_popmenu function	AppeonPopMenu function	Pops up Appeon DataWindow menu at a specified position in a specified DataWindow control.
of_popmenuon function	AppeonPopMenuOn function	Pops up Appeon DataWindow menu in a specified window when you right click the mouse button.
of_print2file function	AppeonPrint2File function	Saves the specified DataWindow as image files of BMP, JPG or GIF format.
of_Print2PDF function (Obsolete)	AppeonPrint2PDF function (Obsolete)	of_Print2PDF is an obsolete function and will be discontinued in a future release. Please replace it with the SaveAs function of

Function of AppeonExtFuncs object	Appeon client functions	Description
		the DataWindow or Child DataWindow and set the saveas file type to PDF format.
of_switchRealTimeCalc function	AppeonSwitchRealTimeCalc function	Performs the DataWindow real-time expression calculation in time or performs the calculation for only one time in the whole life-cycle.

2.3.1.1 of_getappeonusername function

Description

Gets the user name that you type into the Appeon Login dialog box.

Syntax

`appeonextfuncs.of_getappeonusername ()`

Argument	Description
<code>appeonextfuncs</code>	A reference to an AppeonExtFuncs object.

Return value

String.

Usage

This function takes effect in the deployed Appeon application, not in the PowerBuilder application.

2.3.1.2 of_getbrowserversion function

Description

Gets the Internet Explorer version of the Web client.

Supported on the Web client only.

Syntax

`appeonextfuncs.of_getbrowserversion ()`

Argument	Description
<code>appeonextfuncs</code>	A reference to an AppeonExtFuncs object.

Return value

String.

Usage

This function takes effect in the deployed Appeon application, not in the PowerBuilder application.

2.3.1.3 of_getcachedir function

Description

Gets the Cache directory that is used by the current application.

Generally, for the Web application, the cache directory is C:\Users\Administrator\AppData\Local\Microsoft\Windows\Temporary Internet Files\systemfolder1\subsystemfolder1 (The name for the subsystemfolder1 is random), for example, C:\Users\Administrator\AppData\Local\Microsoft\Windows\Temporary Internet Files\content.IE5\. The directory may vary depending on different operating systems.

For the iOS mobile application, the cache directory is /\$Appeon_Mobile_folder\$/Documents/\$your_application_folder\$, for example, /var/mobile/Applications/144F5F33-A33F-480D-A3D9-01BBA5410EB2/Documents/4c001b05.

For the Android mobile application, the cache directory is /data/data/\$Appeon_Mobile_folder\$/files/\$your_application_folder\$, for example, /data/data/com.appeon.mobile/files/21746870.

Syntax

appeonextfuncs.of_getcachedir()

Argument	Description
<i>appeonextfuncs</i>	A reference to an AppeonExtFuncs object.

Return value

String.

Usage

This function takes effect in the deployed Appeon application, not in the PowerBuilder application.

2.3.1.4 of_getclientid function

Description

Gets the unique session identifier for the Web or mobile client.

Syntax

appeonextfuncs.of_getclientid()

Argument	Description
<i>appeonextfuncs</i>	A reference to an AppeonExtFuncs object.

Return value

String.

2.3.1.5 of_getclientip function

Description

Gets the IP address of the Web or mobile client.

Syntax

```
appeonextfuncs.of_getclientip()
```

Argument	Description
<i>appeonextfuncs</i>	A reference to an AppeonExtFuncs object.

Return value

String.

2.3.1.6 of_getclienttype function

Description

Gets the type of an application.

Syntax

```
appeonextfuncs.of_getclienttype()
```

Argument	Description
<i>appeonextfuncs</i>	A reference to an AppeonExtFuncs object.

Return value

String.	Returns "WEB" if the application runs on the Web. Returns "PB" if the application runs in PowerBuilder. Returns "MOBILE" if the application runs on a mobile device.
---------	--

2.3.1.7 of_gethttpinfo function

Description

Gets the HTTP header information from a particular request.

Syntax

```
appeonextfuncs.of_gethttpinfo(string attribute)
```

Argument	Description
<i>appeonextfuncs</i>	A reference to an AppeonExtFuncs object.
<i>attribute</i>	The required HTTP information. For example, "Host", "Cookie", etc.

Return value

String.

Usage

This function takes effect in the deployed Appeon application, not in the original PowerBuilder application.

2.3.1.8 of_getiehandle function

Description

Gets the Internet Explorer handle for the Web application.

Supported on the Web client only.

Syntax

appeonextfuncs.of_getiehandle()

Argument	Description
<i>appeonextfuncs</i>	A reference to an AppeonExtFuncs object.

Return value

Long.

Usage

This function takes effect in the deployed Appeon application, not in the PowerBuilder application.

2.3.1.9 of_getieurl function

Description

Gets the URL of the Web application.

For the mobile application, use [of_getappinfo](#) to get the URL of the mobile application.

Syntax

appeonextfuncs.of_getieurl()

Argument	Description
<i>appeonextfuncs</i>	A reference to an AppeonExtFuncs object.

Return value

String.

Usage

This function takes effect in the deployed Appeon application, not in the PowerBuilder application.

2.3.1.10 of_getostype function

Description

Gets the type of OS that runs your application (the Appeon Web application, the Appeon mobile application, or the PowerBuilder client application).

Syntax

appeonextfuncs.of_getostype ()

Argument	Description
<i>appeonextfuncs</i>	A reference to an AppeonExtFuncs object.

Return value

String.	Returns the type of OS that runs the Appeon Web application, the Appeon mobile application, or the PowerBuilder client application.
---------	---

2.3.1.11 of_getrunmode function

Description

Gets the running mode of an application.

Syntax

appeonextfunc.of_getrunmode ()

Argument	Description
<i>appeonextfuncs</i>	A reference to an AppeonExtFuncs object.

Return value

String	Returns "" an empty string if the application runs in PowerBuilder. Returns "RELEASE" or "DEBUG" (Debug JS, or debug JS/PS) (according to the generation mode setting in Appeon Developer, see Section 4.2.1.1.2, “Selecting an application file generation mode” in <i>Appeon Developer User Guide</i>) if the application runs on the Web or a mobile device.
--------	---

2.3.1.12 of_getservertype function

Description

Gets the Appeon Server type where the application runs.

Syntax

appeonextfuncs.of_getservertype ()

Argument	Description
<i>appeonextfuncs</i>	A reference to an AppeonExtFuncs object.

Return value

Integer	<p>Returns 1 if the Appeon applications runs on an Appeon Server that is installed to a Java Server (such as EAServer).</p> <p>Returns 2 if the Appeon applications runs on an Appeon Server that is installed to a .NET IIS server.</p>
---------	--

Usage

This function takes effect in the deployed Appeon application, not in the PowerBuilder application.

2.3.1.13 of _getsessioncount function

This function is obsolete since Appeon for PowerBuilder 2013.

2.3.1.14 of _is64browser function

Description

Detects if the IE browser where the Web application runs is 64-bit.

Supported on the Web client only.

Syntax

```
appeonextfuncs.of_is64browser ( )
```

Argument	Description
<i>appeonextfuncs</i>	A reference to an AppeonExtFuncs object.

Return value

Boolean	<p>Returns "true" if the Appeon application runs on a 64-bit IE browser.</p> <p>Returns "false" if the Appeon application runs on a 32-bit IE browser.</p>
---------	--

Usage

This function takes effect in the deployed Appeon application, not in the PowerBuilder application.

2.3.1.15 of _ldaplogon function

Description

Logs in to the LDAP server with the specified user name and password.

Syntax

```
appeonextfuncs.of_ldaplogon ( string as_username, string as_password )
```

Argument	Description
<i>appeonextfuncs</i>	A reference to an AppeonExtFuncs object.

<i>as_username</i>	User name for logging into the LDAP server.
<i>as_password</i>	Password for logging into the LDAP server.

Return value

String.	Returns "" an empty string if the login succeeds. Returns the error information.
---------	---

Usage

This function takes effect in the deployed Appeon application, not in the PowerBuilder application. To make this function work in the Appeon application, you should set the System Security to ON in AEM, and make sure the LDAP server is correctly set up. For detailed instructions, refer to Section 5.3.5.5, “System Security” in *Appeon Server Configuration Guide for .NET* or *Appeon Server Configuration Guide for J2EE*.

2.3.1.16 of_log function

Description

Generates the Web/mobile client log information, and the default log level is INFO.

This function is supported on both the Web and mobile clients.

Syntax

appeonextfuncs.of_log (integer *ai_level*, string *as_info*)

Argument	Description
<i>appeonextfuncs</i>	A reference to an AppeonExtFuncs object.
<i>ai_level</i>	1 - FATAL level. 2 - ERROR level. 3 - WARNING level. 4 - INFO level. (Default value) 5 - DEBUG level.
<i>as_info</i>	The log information.

Return value

integer

1 - Success.

-1 - It is called in PowerBuilder, or there is an error.

Usage

For the log path of Web applications, mobile applications (that run in Appeon Workspace), and native mobile applications, please refer to Section 5.4.6.8, “Client Logs” in *Appeon Server Configuration Guide for .NET* or *Appeon Server Configuration Guide for J2EE*.

2.3.1.17 of_clearlog function

Description

Clears the Web/mobile client log information.

This function is supported on both the Web and mobile clients.

Syntax

```
appeonextfuncs.of_clearlog()
```

Argument	Description
<i>appeonextfuncs</i>	A reference to an AppeonExtFuncs object.

Return value

integer

1 - Success.

-1 - It is called in PowerBuilder, or there is an error.

2.3.1.18 of_popmenu function

Description

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

Appeon DataWindow Menu is available in Appeon Web only. For more information about the menu items, refer to Section 8.3, “Appeon DataWindow menu” in *Appeon Developer User Guide*.

Supported on the Web client only.

Syntax

```
appeonextfuncs.of_popmenu ( datawindow adw_dw , integer nx , Integer ny )
```

Argument	Description
<i>appeonextfuncs</i>	A reference to an AppeonExtFuncs object.
<i>adw_dw</i>	The DataWindow control on which you want to pop up the Appeon DataWindow menu.
<i>nx</i>	The distance from the left edge of the popup menu.
<i>ny</i>	The distance from the up edge of the popup menu.

Return value

None.

Usage

1. The user customized RMB menu has a higher priority than the Appeon customized menu.
2. The AppeonPopupMenu function has a higher priority than the AppeonPopupMenuOn function.

3. Defining your RMB menu in RButtonDown event is not recommended because this will cause confusion in the system. To work around this, define your RMB menu in the RButtonUp event.

2.3.1.19 of _popmenuon function

Description

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

Appeon DataWindow Menu is available in Appeon Web only. For more information about the menu items, refer to Section 8.3, “Appeon DataWindow menu” in *Appeon Developer User Guide*.

Supported on the Web client only.

Syntax

```
appeonextfuncs.of_popmenuon ( datawindow adw_dw, Boolean ab_show )
```

Argument	Description
<i>appeonextfuncs</i>	A reference to an AppeonExtFuncs object.
<i>adw_dw</i>	The DataWindow control on which you want to show the Appeon DataWindow menu.
<i>ab_show</i>	Gives an option whether to display the Appeon DataWindow menu. True - Display the Appeon DataWindow menu. False - Not to display the Appeon DataWindow menu.

Return value

None.

Usage

1. The user customized RMB menu has a higher priority than the Appeon customized menu.
2. The AppeonPopupMenu function has a higher priority than the AppeonPopupMenuOn function.
3. 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.

2.3.1.20 of _print2file function

Description

Saves the specified DataWindow as image files of BMP, JPG or GIF format.

Supported on the Web client only.

Syntax

appeonextfuncs.of_print2file (datawindow *adw*, string *asoutpath*, string *asoutname*, long *alouttype*)

Argument	Description
<i>appeonextfuncs</i>	A reference to an AppeonExtFuncs object.
<i>adw</i>	The DataWindow object to be saved as image files.
<i>asoutpath</i>	The path of image files.
<i>asoutname</i>	The specified name of image files.
<i>alouttype</i>	The format type of files: 1-BMP; 2-JPG; 3-GIF

Return value

Integer	<p>Returns 1 if it succeeds in saving the specified DataWindow as image files.</p> <p>Returns -1 if an unknown error occurs.</p> <p>Returns -2 if alouttype is an unsupported format.</p> <p>Returns -3 if adw is an invalid DataWindow object, DataStore object or DataWindowChild object.</p> <p>Returns -4 if it fails in creating a file, e.g., the specified path does not exist or without access rights.</p> <p>Returns -5 if it fails in creating device context, e.g., user sets a large size when customizing page property.</p> <p>Returns -6~-12 if an internal error occurs.</p>
---------	---

Usage

1. This function is used to execute saving DataWindow as image files.
2. If the page size is large enough, a DataWindow is saved as one file; if the size of a DataWindow surpasses the page size, the DataWindow is saved as several files.
3. The asoutname is the file name specified by user, for example, the function is `appeonprint2file(adw, "c:\", "appeon", 1)`, if a DataWindow is saved as one file, the file is like `C:\appeon.bmp`; if a DataWindow is saved as several files, the files are like `C:\appeon1.bmp, C:\appeon2.bmp, ..., C:\appeonN.bmp`. Another example, the function is `appeonprint2file(adw, "c:\", "test.bmp", 1)`, if a DataWindow is saved as one file, the file is like `C:\ test.bmp.bmp`; if a DataWindow is saved as several files, the files are like `C:\test.bmp1.bmp, C:\test.bmp2.bmp, ..., C:\test.bmpN.bmp`.
4. The upper limit of page size is restrained by the type of operating system. For instance, in Windows Vista, the upper limit of customized page size is 5500*5500 around. However, the page size can also be 1024*10000 by reducing page width and increasing page height.

2.3.1.21 of `_Print2PDF` function (Obsolete)

Obsolete function

`of_Print2PDF` is an obsolete function and will be discontinued in a future release. Please replace it with the `SaveAs` function of the `DataWindow` or `Child DataWindow` and set the `saveas` file type to PDF format. For details, refer to Section 11.10.3, “Functions of `DataWindow` control” in *Supported PB Features for Appeon Web* or *Supported PB Features for Appeon Mobile*.

Supported on the Web client only.

2.3.1.22 of `_switchRealTimeCalc` function

Description

Performs the `DataWindow` real-time expression calculation in time or performs the calculation for only one time in the whole life-cycle.

Syntax

`appeonextfuncs.of_switchRealTimeCalc (powerobject adw, integer para)`

Argument	Description
<code>appeonextfuncs</code>	A reference to an <code>AppeonExtFuncs</code> object.
<code>adw</code>	The <code>DataWindow/DataStore/DataWindowChild</code> object.
<code>para</code>	0 - Default value, performs the real-time calculation; 1 - No need to perform the real-time calculation, and performs the calculation for only one time in the whole life-cycle.

Return value

Integer.	0 - Success. -1 - Failure.
----------	-------------------------------

2.3.2 Appeon Client Functions

Appeon client functions are PowerBuilder global functions which perform identical functionalities to the function of [AppeonExtFuncs object](#). However, global functions are not recommended for use due to their poor extensibility and performance compared to object functions, therefore, Appeon client function should be replaced with the equivalent function of [AppeonExtFuncs object](#).

2.3.2.1 `AppeonGetAppeonUserName` function

Description

Gets the user name that you type into the Appeon Login dialog box.

Syntax

AppeonGetAppeonUserName ()

Return value

String.

Usage

This function takes effect in the deployed Appeon application, not in the PowerBuilder application.

2.3.2.2 AppeonGetBrowserVersion function

Description

Gets the Internet Explorer version of the Web client.

Supported on the Web client only.

Syntax

AppeonGetBrowserVersion ()

Return value

String.

Usage

This function takes effect in the deployed Appeon application, not in the PowerBuilder application.

2.3.2.3 AppeonGetCacheDir function

Description

Gets the Cache directory that is used by the current application.

Generally, for the Web application, the cache directory is C:\Users\Administrator\AppData\Local\Microsoft\Windows\Temporary Internet Files\systemfolder1\subsystemfolder1 (The name for the subsystemfolder1 is random), for example, C:\Users\Administrator\AppData\Local\Microsoft\Windows\Temporary Internet Files\content.IE5\. The directory may vary depending on different operating systems.

For the iOS mobile application, the cache directory is /\$Appeon_Mobile_folder\$/Documents/\$your_application_folder\$, for example, /var/mobile/Applications/144F5F33-A33F-480D-A3D9-01BBA5410EB2/Documents/4c001b05.

For the Android mobile application, the cache directory is /data/data/\$Appeon_Mobile_folder\$/files/\$your_application_folder\$, for example, /data/data/com.appeon.mobile/files/21746870.

Syntax

AppeonGetCacheDir ()

Return value

String.

Usage

This function takes effect in the deployed Appeon application, not in the PowerBuilder application.

2.3.2.4 AppeonGetClientID function

Description

Gets the unique session identifier for the Web or mobile client.

Syntax

```
AppeonGetClientID ()
```

Return value

String.

2.3.2.5 AppeonGetClientIP function

Description

Gets the IP address of the Web or mobile client.

Syntax

```
AppeonGetClientIP ()
```

Return value

String.

Usage

This function takes effect in the deployed Appeon application, not in the PowerBuilder application.

2.3.2.6 AppeonGetClientType function

Description

Gets the type of an application.

Syntax

```
AppeonGetClientType ()
```

Return value

String	Returns "WEB" if the application runs on the Web. Returns "PB" if the application runs in PowerBuilder. Returns "MOBILE" if the application runs on a mobile device.
--------	--

2.3.2.7 AppeonGetHttpInfo function

Description

Gets the HTTP header information from a particular request.

Syntax

AppeonGetHttpInfo (string *attribute*)

Argument	Description
<i>attribute</i>	The required HTTP information. For example, "Host", "Cookie", etc.

Return value

String

2.3.2.8 AppeonGetIEHandle function

Description

Gets the Internet Explorer handle for the Web application.

Supported on the Web client only.

Syntax

AppeonGetIEHandle ()

Return value

Long.

Usage

This function takes effect in the deployed Appeon application, not in the PowerBuilder application.

2.3.2.9 AppeonGetIEURL function

Description

Gets the URL of the Web application.

For the mobile application, use [of_getappinfo](#) to get the URL of the mobile application.

Syntax

AppeonGetIEURL ()

Return value

String.

Usage

This function takes effect in the deployed Appeon application, not in the PowerBuilder application.

2.3.2.10 AppeonGetOSType function

Description

Gets the type of OS that runs your application (the Appeon Web application, the Appeon mobile application, or the PowerBuilder client application).

Syntax

AppeonGetOSType ()

Return value

String.	Returns the type of OS that runs the Appeon Web application, the Appeon mobile application, or the PowerBuilder client application.
---------	---

2.3.2.11 AppeonGetRemainingdays function

Description

Gets the remaining day(s) of license or technical support.

Syntax

AppeonGetRemainingdays (String *as_type*, ref string *as_error*)

Argument	Description
<i>as_type</i>	License or technical support that you want to get the remaining day(s). "license" indicates to get remaining day(s) of license. "support" indicates to get remaining day(s) of technical support.
<i>as_error</i>	An empty string or error messages.

Return value

Long.	<p>Returns a number >0 if the license or technical support has remaining day(s).</p> <p>Returns 0 if there is no expiration date.</p> <p>Returns -1 if license or technical support has expired, Appeon Server has an exception, or parameter is invalid.</p>
-------	--

2.3.2.12 AppeonGetServerType function

Description

Gets the Appeon Server type where the application runs.

Syntax

AppeonGetServerType ()

Return value

Integer.	<p>Returns 1 if the application runs on an Appeon Server that is installed to a Java Server (such as EAServer).</p> <p>Returns 2 if the application runs on an Appeon Server that is installed to a .NET IIS server.</p>
----------	--

Usage

This function takes effect in the deployed Appeon application, not in the PowerBuilder application.

2.3.2.13 AppeonGetSessionCount function

This function is obsolete since Appeon for PowerBuilder 2013.

2.3.2.14 appeonisin64browser function

Description

Detects if the IE browser where the application runs is 64-bit.

Syntax

```
appeonisin64browser ( )
```

Return value

Boolean.	Returns "true" if the application runs on a 64-bit IE browser. Returns "false" if the application runs on a 32-bit IE browser.
----------	---

Usage

This function takes effect in the deployed Appeon application, not in the PowerBuilder application.

2.3.2.15 AppeonLDAPLogon function

Description

Logs in to the LDAP server with the specified user name and password.

Syntax

```
AppeonLDAPLogon ( string as_username, string as_password )
```

Argument	Description
<i>as_username</i>	User name for logging into the LDAP server.
<i>as_password</i>	Password for logging into the LDAP server.

Return value

String.

Usage

This function takes effect in the deployed Appeon application, not in the PowerBuilder application. To make this function work in the deployed application, you should set the System Security to **ON** in AEM, and make sure the **LDAP server** is correctly set up. For detailed instructions, refer to Section 5.3.5.5, "System Security" in *Appeon Server Configuration Guide for .NET* or *Appeon Server Configuration Guide for J2EE*.

2.3.2.16 AppeonPopupMenu function

Description

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

Appeon DataWindow Menu is available in Appeon Web only. For more information about the menu items, refer to Section 8.3, “Appeon DataWindow menu” in *Appeon Developer User Guide*.

Supported on the Web client only.

Syntax

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

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

Return value

None.

Usage

1. The user customized RMB menu has a higher priority than the Appeon customized menu.
2. The AppeonPopupMenu function has a higher priority than the AppeonPopupMenuOn function.
3. Defining your RMB menu in RButtonDown event is not recommended because this will cause confusion in the system. To work around this, define your RMB menu in the RButtonUp event.

2.3.2.17 AppeonPopupMenuOn function

Description

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

Appeon DataWindow Menu is available in Appeon Web only. For more information about the menu items, refer to Section 8.3, “Appeon DataWindow menu” in *Appeon Developer User Guide*.

Supported on the Web client only.

Syntax

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

Argument	Description
<i>adw_dw</i>	The DataWindow control on which you want to show the Appeon DataWindow menu.
<i>ab_show</i>	Gives an option whether to display or not display the Appeon DataWindow menu.

Argument	Description
	True - Enables the display of the Appeon DataWindow menu. False - Disables the display of the Appeon DataWindow menu.

Return value

None.

Usage

1. The user customized RMB menu has a higher priority than the Appeon customized menu.
2. The AppeonPopupMenu function has a higher priority than the AppeonPopupMenuOn function.
3. 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.

2.3.2.18 AppeonPrint2File function

Description

Saves the specified DataWindow as image files of BMP, JPG or GIF format.

Supported on the Web client only.

Syntax

`AppeonPrint2File (datawindow adw, string asoutpath, string asoutname, long alouttype)`

Argument	Description
<i>adw</i>	The DataWindow object to be saved as image files
<i>asoutpath</i>	The path of image files
<i>asoutname</i>	The specified name of image files
<i>alouttype</i>	The format type of files: 1-BMP; 2-JPG; 3-GIF

Return value

Integer.	Returns 1 if it succeeds in saving the specified DataWindow as image files. Returns -1 if an unknown error occurs. Returns -2 if <i>alouttype</i> is an unsupported image format. Returns -3 if <i>adw</i> is an invalid DataWindow object, DataStore object or DataWindowChild object.
----------	--

	Returns -4 if it fails in creating a file, e.g., the specified path does not exist or without access rights.
	Returns -5 if it fails in creating device context, e.g., user sets a large size when customizing page property.
	Returns -6~-12 if an internal error occurs.

Usage

1. This function is used to execute saving DataWindow as image files.
2. If the page size is large enough, a DataWindow is saved as one file; if the size of a DataWindow surpasses the page size, the DataWindow is saved as several files.
3. The asoutname is the file name specified by user, for example, the function is `appeonprint2file(adw, "c:", "appeon", 1)`, if a DataWindow is saved as one file, the file is like `C:\appeon.bmp`; if a DataWindow is saved as several files, the files are like `C:\appeon1.bmp, C:\appeon2.bmp, ..., C:\appeonN.bmp`. Another example, the function is `appeonprint2file(adw, "c:", "test.bmp", 1)`, if a DataWindow is saved as one file, the file is like `C:\ test.bmp.bmp`; if a DataWindow is saved as several files, the files are like `C:\test.bmp1.bmp, C:\test.bmp2.bmp, ..., C:\test.bmpN.bmp`.
4. The upper limit of page size is restrained by the type of operating system. For instance, in Windows Vista, the upper limit of customized page size is 5500*5500 around. However, the page size can also be 1024*10000 by reducing page width and increasing page height.

2.3.2.19 AppeonPrint2PDF function (Obsolete)

Obsolete function

AppeonPrint2PDF is an obsolete function and will be discontinued in a future release. Please replace it with the SaveAs function of the DataWindow or Child DataWindow and set the saveas file type to PDF format. For details, refer to Section 11.10.3, “Functions of DataWindow control” in *Supported PB Features for Appeon Web* or *Supported PB Features for Appeon Mobile*.

Supported on the Web client only.

2.3.2.20 AppeonSwitchRealTimeCalc function

Description

Performs the DataWindow real-time expression calculation in time or performs the calculation for only 1 time in the whole life-cycle.

Syntax

`AppeonSwitchRealTimeCalc (powerobject adw, integer para)`

Argument	Description
<i>adw</i>	The DataWindow/DataStore/DataWindowChild object.

Argument	Description
<i>para</i>	0 - Default value, performs the real-time calculation; 1 - No need to perform the real-time calculation, and performs the calculation for only one time in the whole life-cycle.

Return value

Integer.	0 - Success. -1 - Failure.
----------	-------------------------------

2.3.3 Client Logs

2.3.3.1 eon_logex object

Description

Facilitates users to debug the deployed applications.

The of_log function and of_clearlog function of eon_logex object call the corresponding function of AppeonExtFuncs object respectively, please refer to [of_log function](#) and [of_clearlog function](#) of AppeonExtFuncs object for details.

These two functions are supported on both the Web and mobile clients.

2.3.4 File Upload and Download

Appeon provides a non-visual object, AppeonFileService object, with five functions for uploading files to and downloading files from the file server. This is a web-based solution, and it does not work in your client/server application.

You should follow the steps below to implement File Upload and Download:

Step 1: [Configuring and deploying the file server](#)

Step 2: [Uploading and Downloading files](#)

2.3.4.1 Configuring and deploying Appeon File Server

Appeon File Server is a standard non-visual Web application running on the back-end for uploading and downloading files. For Appeon Server installed to the .NET IIS, the setup wizard of Appeon File Server (setup.exe) can be found under *%IIS_Web_Root%\appeon\plugin\fileservice*. For Appeon Server installed to the Java server, the WAR package of Appeon File Server (fileservice.war) can be found under *%Appeon_Server%\plugin\fileservice*.

You must configure and deploy the file server first in order to use the AppeonFileService object in the Appeon Workarounds PBL. In most cases Appeon File Server is deployed to the machine where the application server is installed. It can also be deployed separately. For example, when the file size is too big, uploading or downloading it will negatively impact the performance of the application server, you can deploy the file server separately to another machine.

In the following section, you will be guided to configure and deploy Appeon File Server to IIS (.NET Framework), WebLogic, WebSphere, JBoss, JEUS, EAServer and NetWeaver. For more about the deploy instructions, refer to related documents of IIS, WebLogic, WebSphere, JBoss, JEUS, EAServer, and NetWeaver.

2.3.4.1.1 Deploying Appeon File Server to IIS

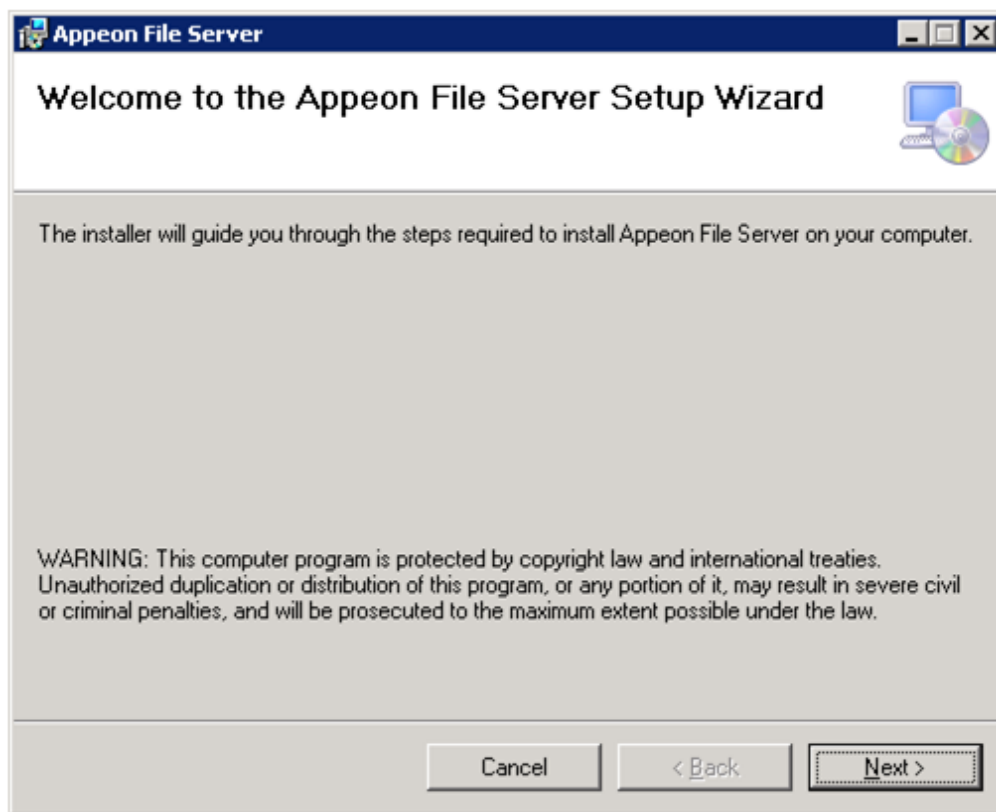
Installing the Appeon File Server

You will need to install first and then configure the Appeon File Server. For installing the file server to IIS, Appeon provides a **setup.exe** file at `%IIS_Web_Root%\appeon\plugin\fileservice`.

Step 1: Double click the **setup.exe** file to start the setup.

If the "Installation Incomplete" error displays, please try the solution in the Section 1.4.1, "Appeon File Server Installation Incomplete" in *Troubleshooting*.

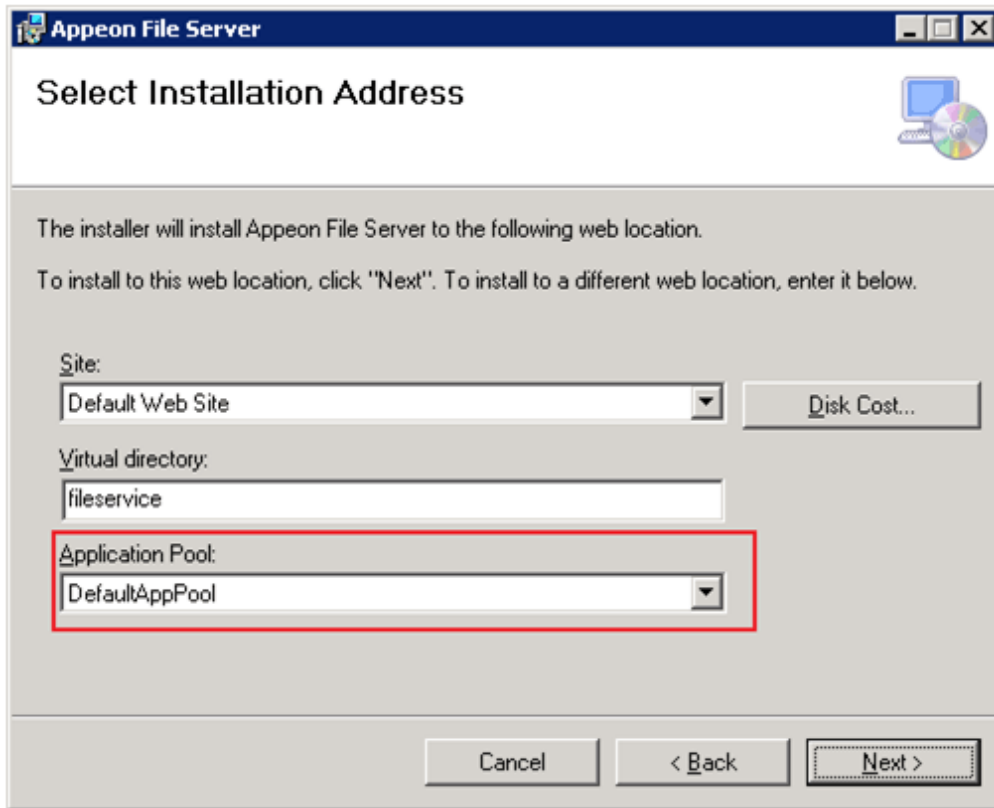
Figure 2.1: Appeon File Service Setup Wizard



Step 2: Click the **Next** button and then the following window pops up. Select a Web site where the Appeon File Server will be installed.

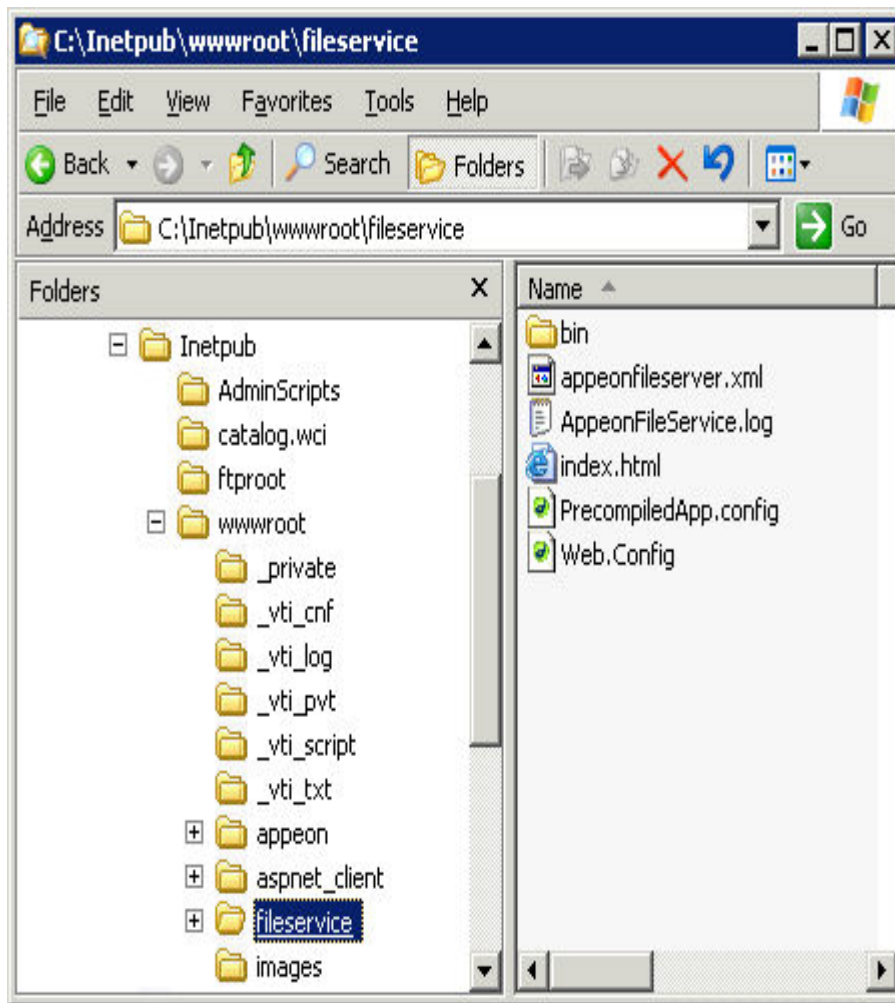
Note that DO NOT change the name of Virtual directory otherwise Appeon File Server will fail to start.

Figure 2.2: Select Installation Address



Step 3: Click **Next** until the installation is complete.

Step 4: Go to directory where the Appeon File Server is installed, for example (*C:\inetpub\wwwroot\fileservice*).

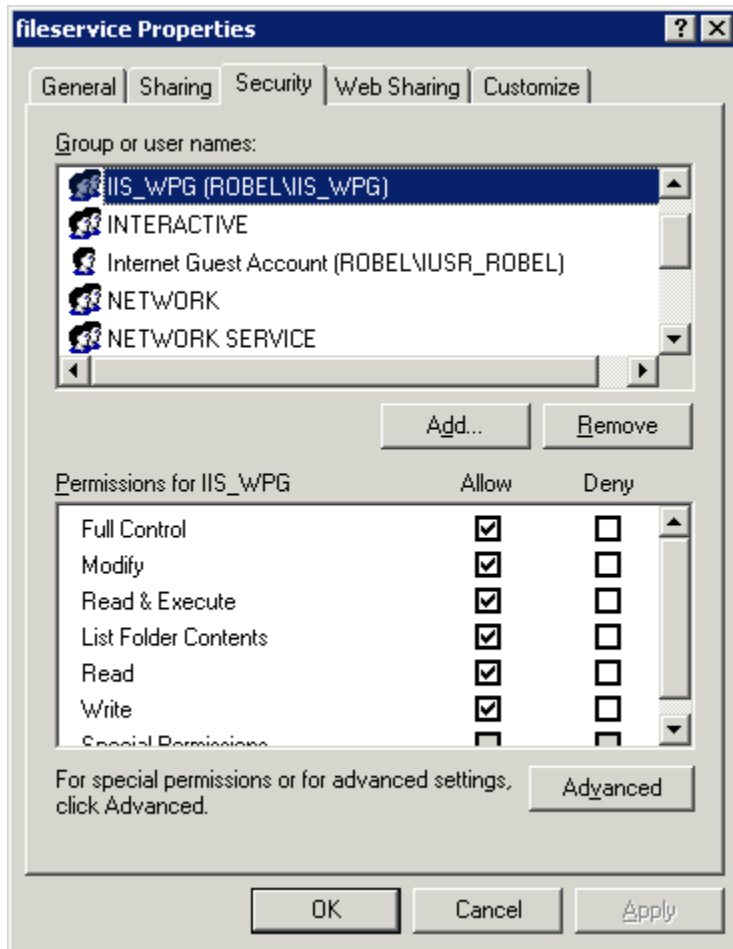
Figure 2.3: Apeon File Server directory

Step 5: Select **Properties** from the right-click menu of the **fileservice** folder.

Step 6: In the **Security** tab page of the **fileservice Properties** window,

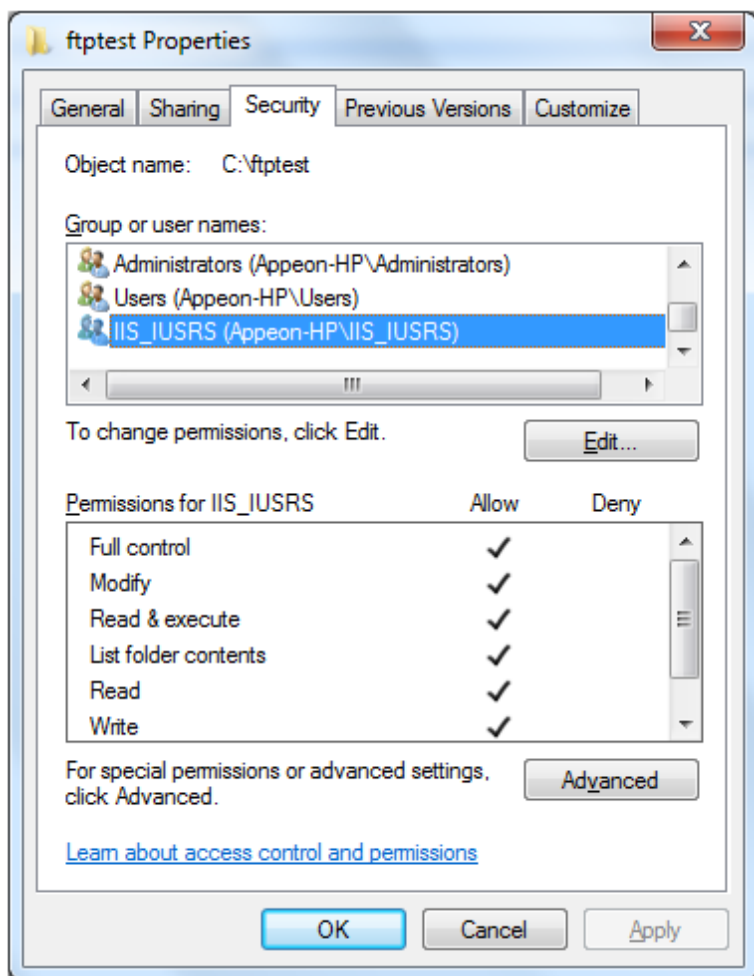
For **IIS 6.0**, grant the **IIS_WPG** user with **Full Control** permission to this folder.

Figure 2.4: Grant Full Control (IIS6)



For **IIS 7.0 or above**, grant the **IIS_IUSRS** user with **Full Control** permission to this folder.

Figure 2.5: Grant Full Control (IIS7)



Uninstall Appeon File Server

Step 1: Go to **Start | Control Panel | Add and Remove Programs**.

Step 2: Select **Appeon File Server** and click **Remove** button.

Configuring the Appeon File Server

After deploying the Appeon File Server, follow steps below to configure it.

Open the `appeonfileservice.xml` in `C:\Inetpub\wwwroot\fileservice`.

```
<?xml version="1.0" encoding="UTF-8"?>
<webserver>
<!-- The value of attribute "value" must begin with either a single or double quote character. -->
<file-path value="D:\appeon\upload" />
<log-level value="3" /> <!--0 Error,1 Info,2 Func, 3 Debug-->
<session-timeout value="3600" /> <!-- the unit is second -->
<allowed-file-types value="txt;doc;jpg;mpeg" ignorecase="true" />
<max-file-size value="20" /> <!-- the unit is M -->
<users>
<user name="test" password="password" />
<user name="userA" password="userA" />
<user name="userB" password="userB" />
</users>
</webserver>
```

Modify the file according to your own demands.

Table 2.2: Settings specification value

Settings	The value of the setting specifies...
<file-path>	Specifies the directory to which all uploaded files are saved. Files are automatically saved under the application folder.
<log-level>	Specify the log level. Value: 0 - Error 1 - Information 2 - Function 3 - Debug
<session-timeout>	Specifies the time, in seconds, that passes after the last request is processed before the session times out.
<allow-file-type>	Specifies the file types that can be uploaded. Use a ";" between two file types. 1. "*" - Any file types are allowed. Note: Do not use "*" with other file types because "*" will not take any effect. For example, "*;text" means that only text file is allowed. 2. Using a "-" before the file types - Any file type is allowed, excluding the listed ones. For example, "-text;doc;jpg;mpeg" means that any file type is allowed, excluding text, doc, jpg and mpeg files. 3. "Ignorecase" - The file type is not case sensitive.
<max-file-size>	Specifies the maximum size (MB) of the file that can be uploaded.
<user>	Specifies the User Name and Password that can log on the file server. Multiple users are allowed as shown in the code example above.

2.3.4.1.2 Deploying Appeon File Server to WebLogic

Configuring the Appeon File Server

You will need to configure first and then deploy the Appeon File Server. Follow steps below to configure the Appeon File Server:

Step 1: On the machine where the Appeon File Server will be deployed, create an XML file wherever you like and name it whatever you wish. In this example, create an XML file named **"appeonfileserver.xml"** under *D:\appeon\config directory*.

Step 2: Copy the following code to the XML file.


```
<?xml version="1.0" encoding="UTF-8"?>
<fileserver>
  <file-path value="D:\appeon" />
  <session-timeout value="3600" />
  <allowed-file-types value="txt;doc;jpg;mpeg" ignorecase="true" />
  <max-file-size value="20" />
  <users>
    <user name="userA" password="userA" />
    <user name="userB" password="userB" />
  </users>
</fileserver>
```

Step 3: Modify the settings in the XML file if necessary.

Table 2.3: The settings specification value

Settings	The value of the setting specifies...
<file-path>	Specifies the directory to which all uploaded files are saved. Files are automatically saved under the application folder.
<session-timeout>	Specifies the time, in seconds, that passes after the last request is processed before the session times out.
<allow-file-type>	Specifies the file types that can be uploaded. Use a ";" between two file types. <ol style="list-style-type: none"> 1. "*" - Any file types are allowed. Note: Do not use "*" with other file types because "*" will not take any effect. For example, "*;text" means that only text file is allowed. 2. "-" - Exclude the file types listed after "-". For example, "-text;doc;jpg;mpeg" means that any file type is allowed, except for text, doc, jpg and mpeg files. 3. "Ignorecase" - The file type is not case sensitive.
<max-file-size>	Specifies the maximum size (MB) of the file that can be uploaded.
<user>	Specifies the User Name and Password that can log on the file server. Multiple users are allowed as shown in the code example above.

Step 4: On the machine where the Appeon Server is installed, unzip the **fileservice.war** (%AppeonServer%\plugin\fileservice) and find the **web.xml** file under the **WEB-INF** folder. Then open **web.xml** with a text editor. Replace the **bold** text with the name and directory of the XML file created in the previous steps.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
'http://java.sun.com/dtd/web-app_2_3.dtd'>
<web-app>

<servlet>
  <servlet-name>filesevice</servlet-name>
  <servlet-class>com.appeon.fileserver.WebServiceDispatcher</servlet-class>
  <init-param>
    <param-name>config</param-name>
    <param-value>D:\appeon\config\appeonfileserver.xml</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet>
  <servlet-name>uploadfile</servlet-name>
  <servlet-class>com.appeon.fileserver.UploadFile</servlet-class>
  <init-param>
    <param-name>config</param-name>
    <param-value>D:\appeon\config\appeonfileserver.xml</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>filesevice</servlet-name>
  <url-pattern>/filesevice</url-pattern>
</servlet-mapping>

</web-app>
```

Step 5: Save the file and re-compress the **filesevice** folder to **filesevice.war** using WinZip, WinRAR or JDK. Do not use the other compression tools such as 7-zip.

Then follow the next section [Deploying the Appeon File Server](#) to deploy filesevice.war.

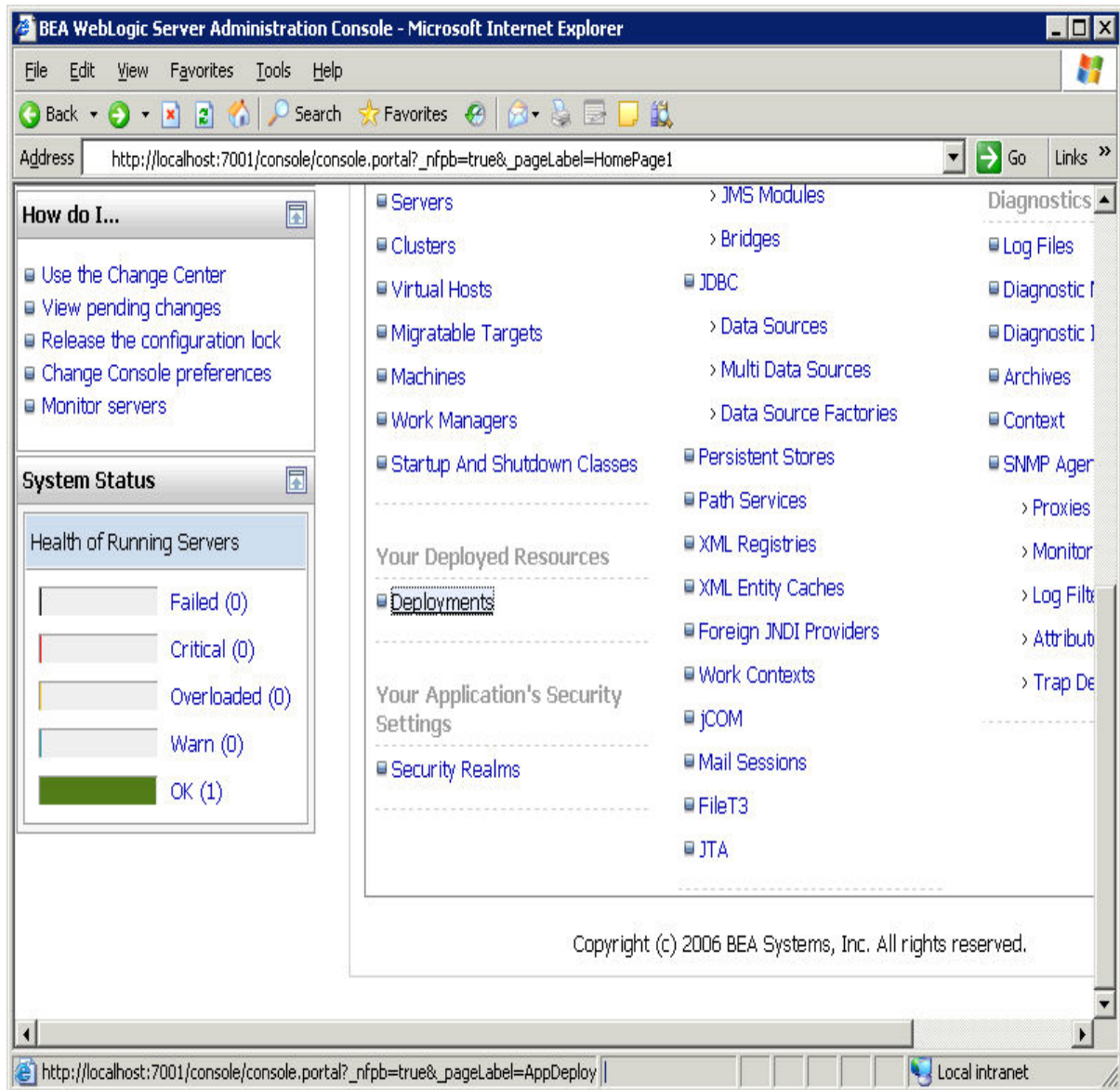
Deploying the Appeon File Server

After configuring the Appeon File Server, follow steps below to deploy it:

Step 1: On the machine where Appeon Server is installed, access the **WebLogic Server Administration Console** in a Web browser.

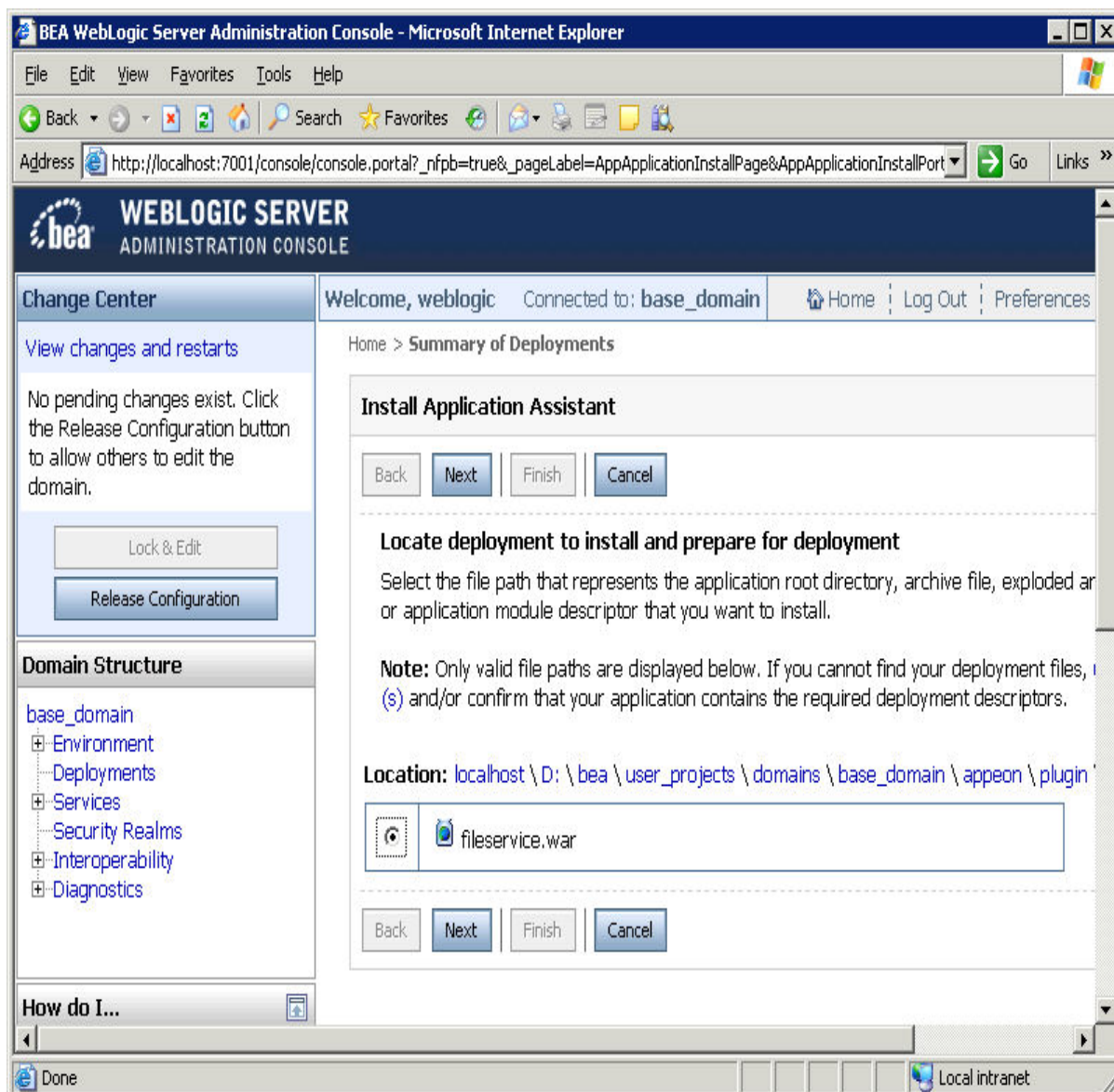
Step 2: Click the **Deployment** link.

Figure 2.6:



Step 3: Click the **Install** button and select the **fileservice.war** file at `%ApeonServer%\plugin\fileservice`.

Figure 2.7:



Step 4: Click the **Next** button to use the default settings and then click the **Finish** button to finish the deployment of Appeon File Server.

2.3.4.1.3 Deploying Appeon File Server to WebSphere

Configuring the Appeon File Server

Configuring the Appeon File Server in the Java server such as WebLogic, WebSphere, JBoss, JEUS, NetWeaver etc. is the same. Please refer to [Configuring the Appeon File Server](#) for WebLogic.

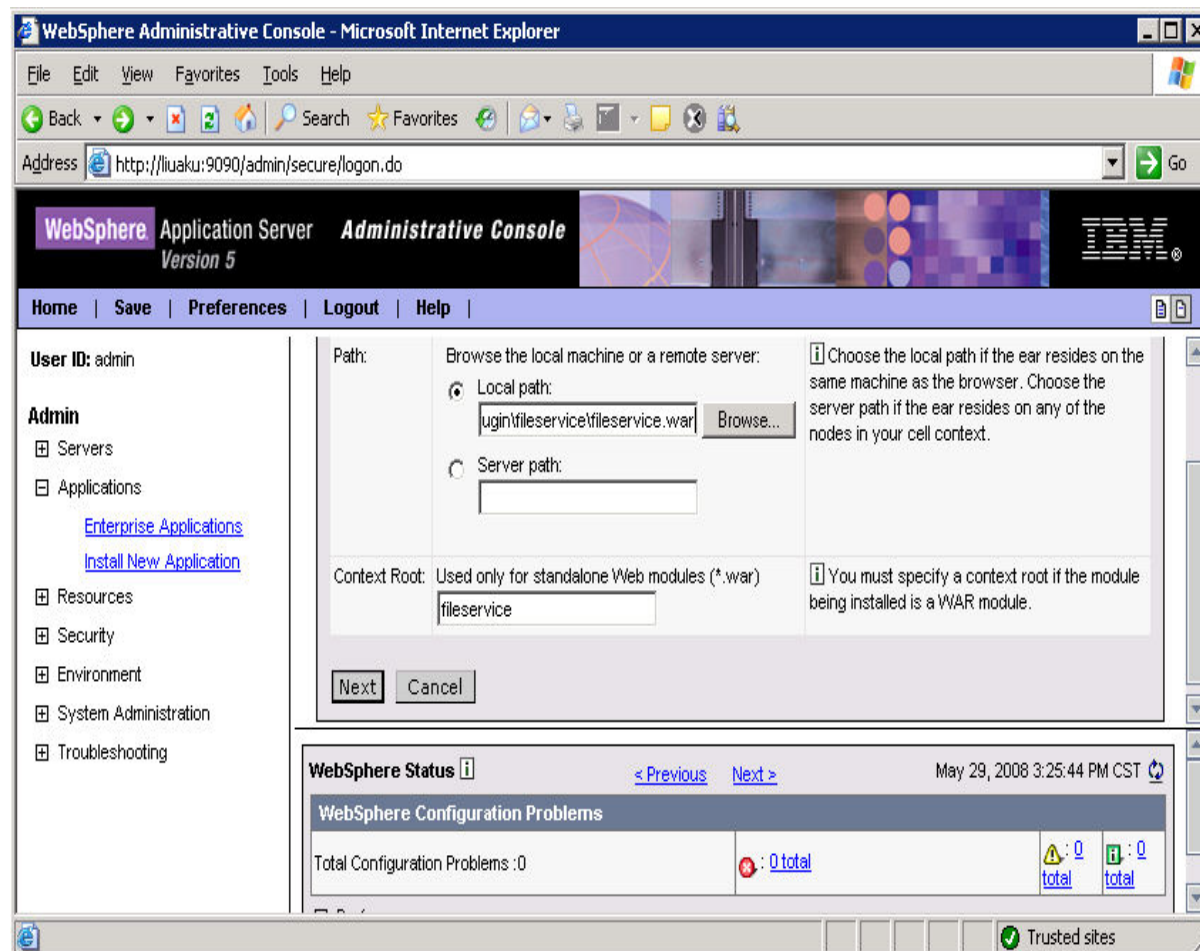
Deploying the Appeon File Server

After configuring the Appeon File Server, follow steps below to deploy it:

Step 1: On the machine where Appeon Server is installed, access the **WebSphere Administrative Console** in a Web browser.

Step 2: Select **Applications** | **Install New Applications** from the left tree view. And then in the right page specify the path where the **fileservice.war** is (by default at *%AppeonServer%\plugin\fileservice*) and **Context Root** for the WAR file and click the **Next** button.

Figure 2.8:



Step 3: Click **Next** with default settings until the file server is successfully installed on the WebSphere server.

2.3.4.1.4 Deploying Appeon File Server to JBoss

Configuring the Appeon File Server

Configuring the Appeon File Server in the Java server such as WebLogic, WebSphere, JBoss, JEUS, NetWeaver etc. is the same. Please refer to [Configuring the Appeon File Server](#) for WebLogic.

Deploying the Appeon File Server

After configuring the Appeon File Server, follow steps below to deploy it. There are two methods to deploy the Appeon File Server in JBoss:

- Automatic deploy

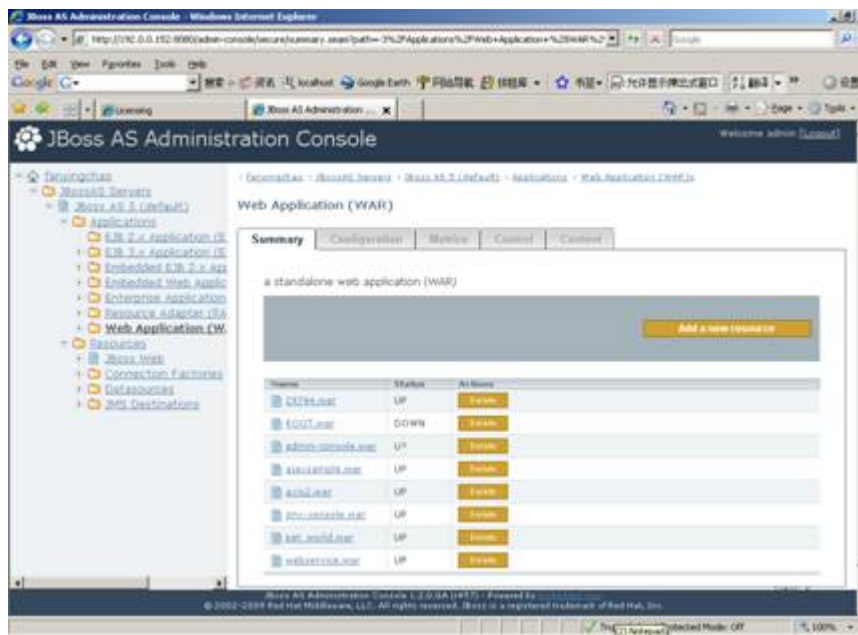
Copy the file **fileservice.war** to the directory *%JBoss installation root directory%\server\<instance>\deploy* and then start JBoss.

The fileservice.war file will be deployed automatically.

- Manual deploy

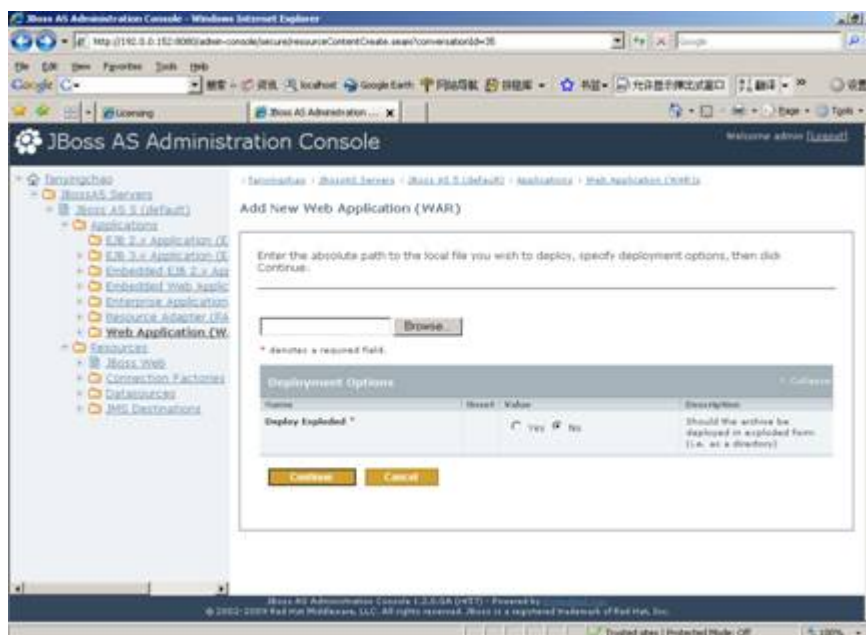
1. Start JBoss and log in to JBoss console.
2. Select **Web Application** in the left tree view and then click **Add a new resource** in the right page.

Figure 2.9: PDF



3. Click **Browse** and locate the file **fileservice.war**.

Figure 2.10:



4. Click **Continue** with default settings until the file server is successfully deployed in the JBoss server.

2.3.4.1.5 Deploying Appeon File Server to JEUS

Configuring the Appeon File Server

Configuring the Appeon File Server in the Java server such as WebLogic, WebSphere, JBoss, JEUS, NetWeaver etc. is the same. Please refer to [Configuring the Appeon File Server](#) for WebLogic.

Deploying the Appeon File Server

After configuring the Appeon File Server, follow steps below to deploy it:

Copy the **fileservice.war** file to the directory *%JEUS installation root directory%\webhome\autodeploy* and then start JEUS server. The fileservice.war file will be deployed automatically.

2.3.4.1.6 Deploying Appeon File Server to EAServer

Configuring the Appeon File Server

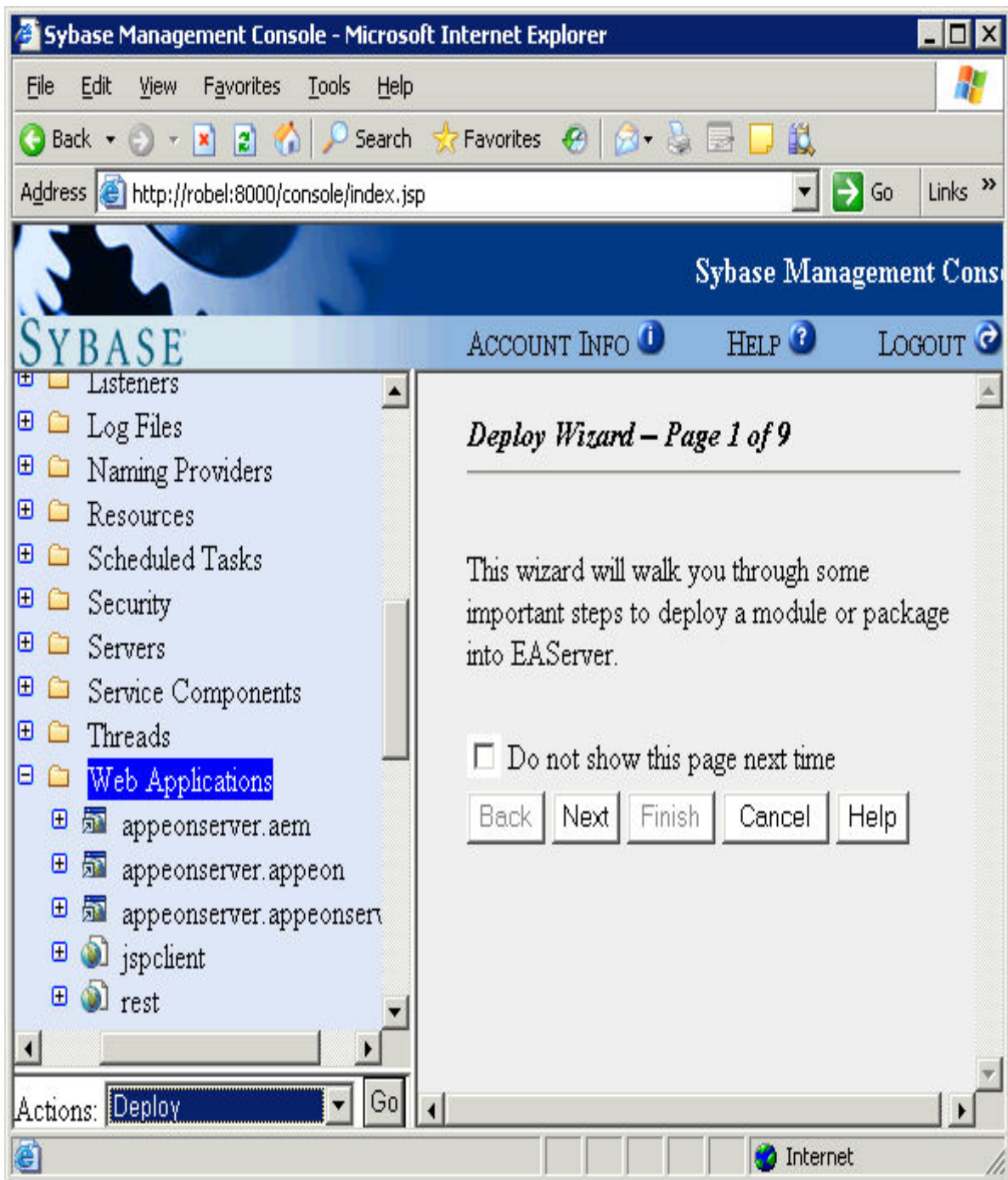
Configuring the Appeon File Server in the Java server such as WebLogic, WebSphere, JBoss, JEUS, NetWeaver etc. is the same. Please refer to [Configuring the Appeon File Server](#) for WebLogic.

Deploying the Appeon File Server

After configuring the Appeon File Server, follow steps below to deploy it:

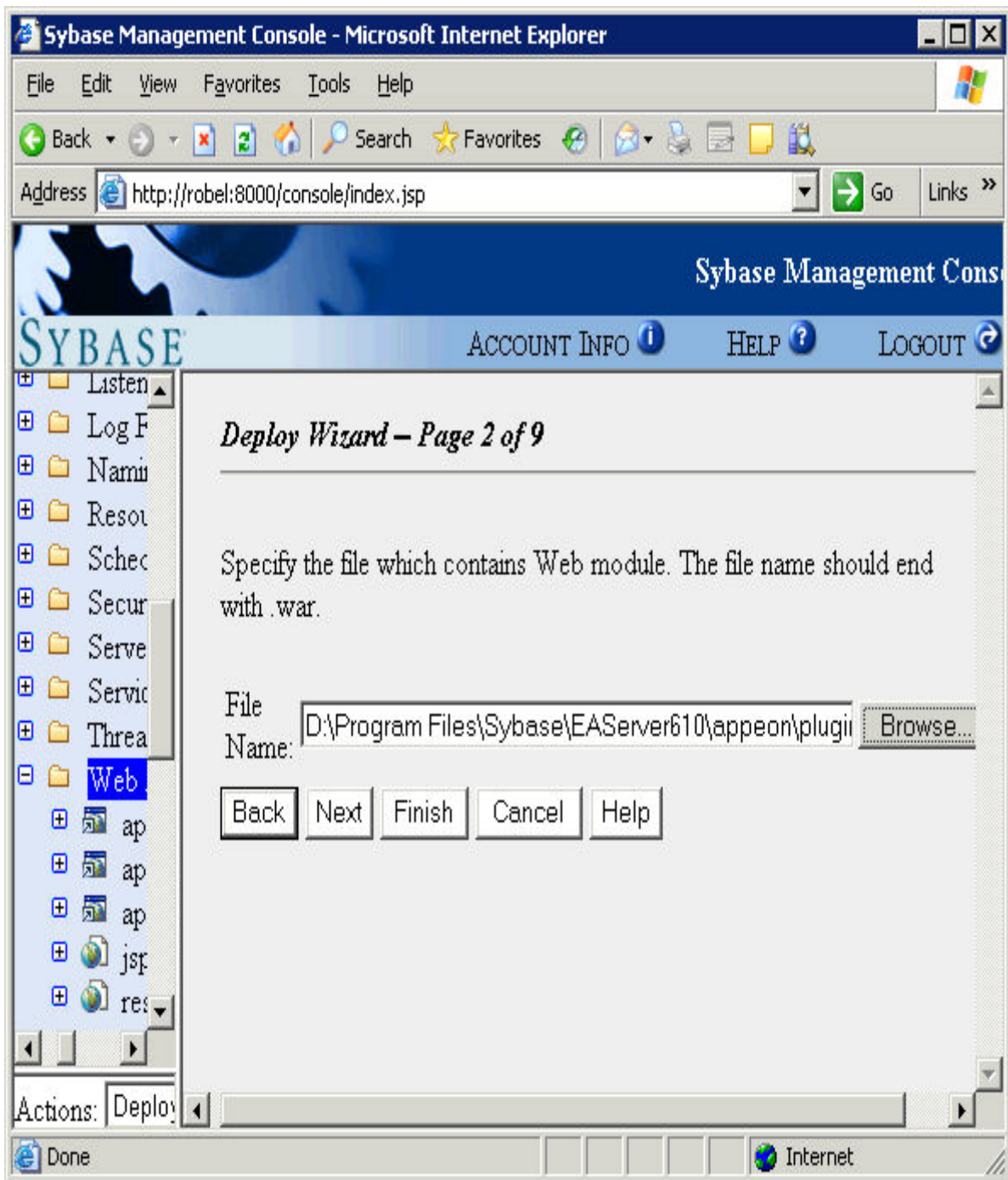
Step 1: Open **Sybase Management Console**, click **Web Applications** and select **Deploy** from the **Actions** list box and click the **Go** button.

Figure 2.11:



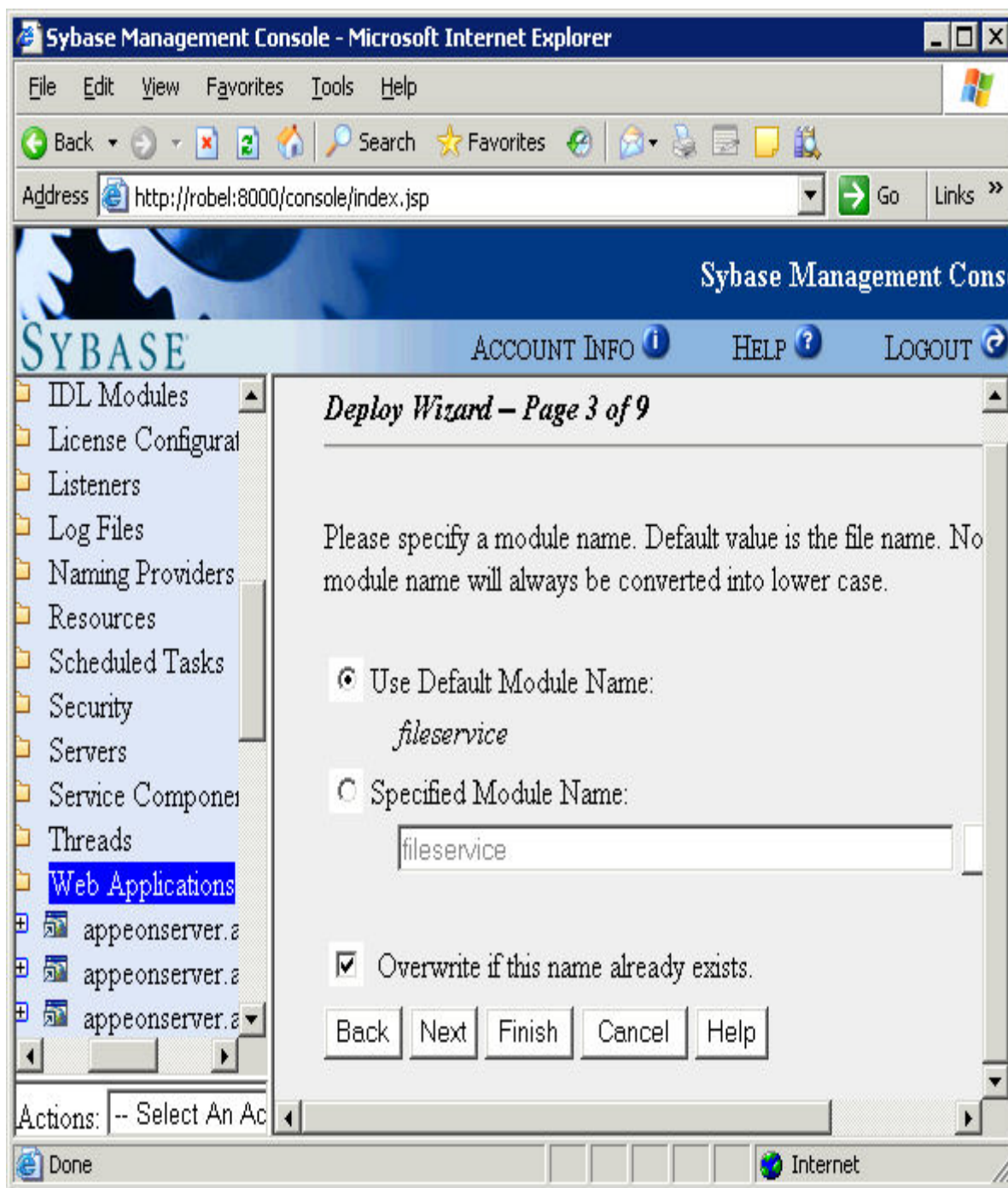
Step 2: Click **Browse** to choose the **fileservice.war** file and then click **Next**.

Figure 2.12:



Step 3: Choose **Use Default Module Name** and click **Next** to continue the deployment.

Figure 2.13:



Step 4: Click the **Finish** button to complete the deployment of Appeon File Server.

2.3.4.1.7 Deploying Appeon File Server to NetWeaver

Configuring the Appeon File Server

Configuring the Appeon File Server in the Java server such as WebLogic, WebSphere, JBoss, JEUS, NetWeaver etc. is the same. Please refer to [Configuring the Appeon File Server](#) for WebLogic.

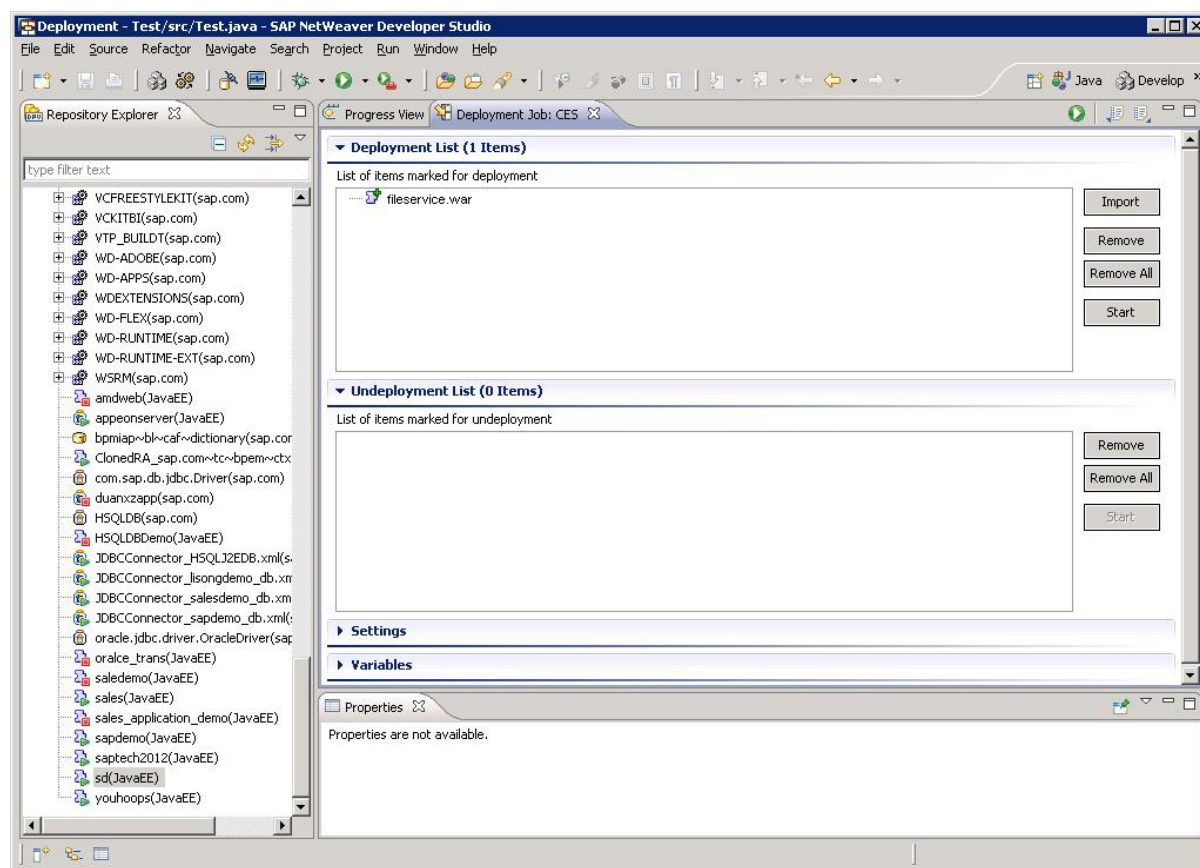
Deploying the Apeon File Server

After configuring the Apeon File Server, follow steps below to deploy it:

Step 1: Open the deployment perspective in the SAP NetWeaver Developer Studio by clicking the menu **Window | Open Perspective | Other** and then selecting **deployment** in the pop-up window. The deployment perspective is shown as below.

Step 2: Click the **Import** button to import the **fileservice.war** file and then click **Start** to deploy the Apeon File Server after the file is imported successfully.

Figure 2.14:



2.3.4.2 Uploading and downloading files

After configuring and deploying the Apeon File Server, you can follow the steps below to upload files to or download files from the Apeon File Server.

Step 1: Add Apeon Workaround PBL to your PowerBuilder project.

Step 2: Define a variable and create an instance for the **ApeonFileService** object.

Step 3: Call [of_logonfileserver](#) to connect to the file server.

Step 4: Call [of_apeonupload](#) or [of_apeondownload](#) to upload or download files.

Step 5: Call [of_logofffileserver](#) to disconnect from the file server.

2.3.4.2.1 ApeonFileService object

ApeonFileService object provides service about the file uploading and downloading.

Functions for AppeonFileService object

AppeonFileService object provides the following functions to upload and download files:

- [of_logonfileservice function](#): Connects to the file server.
- [of_logofffileservice function](#): Disconnects from the file server.
- [of_appeonupload function](#): Uploads file to the file server. You can specify the path where the source file locates and the path where the file will be uploaded.
- [of_appeondownload function](#): Downloads the specified file from the file server. You can specify the path from which the file is downloaded and the path to which the file will be saved.
- [of_FileExists function](#): Reports whether the specified file exists on the file server.

of_logonfileservice

Description

Connects to the Appeon File Server.

Syntax

appeonfileservice.of_logonfileservice (string *fileserviceipaddress*, long *port*, string *connectstring*)

Parameter	Description
appeonfileservice	An instance of an AppeonFileService object
<i>fileserviceipaddress</i>	IP address or machine name of the Appeon File Server.
<i>port</i>	HTTP Port of the application where the Appeon File Server is deployed. HTTPS port is unsupported.
<i>connectstring</i>	User name and password for connecting to the Appeon File Server. Format: "username=username;password=password". Use a semicolon ";" to separate the user name and password. The user name and password should be consistent with the specification in the XML.

Return value

Long.	Return values are: 0 - The logon is successful. -1 - Connection to the file server fails. -2 - The provided user name or password is invalid.
-------	--

of_logofffileservice

Description

Disconnects from the Appeon File Server.

Syntax

appeonfileservice.of_logofffileservice()

Parameter	Description
appeonfileservice	An instance of an AppeonFileService object

Return value

Long.	Return values are: 0 - The logoff is successful. -1 - Connection to the Appeon File Server fails. -2 - The session has timed out or the user has not logged into the file server.
-------	--

of_appeonupload

Description

Uploads file to the Appeon File Server.

Syntax

appeonfileservice.of_appeonupload (string *source*, string *destination*, boolean *isrename*, ref long *errorcode*)

appeonfileservice.of_appeonupload (string *destination*, boolean *isrename*, ref long *errorcode*)

appeonfileservice.of_appeonupload (boolean *isrename*, ref long *errorcode*)

Parameter	Description
appeonfileservice	An instance of an AppeonFileService object.
<i>source</i>	(Optional) The directory of the source file. If no directory is specified here, a dialog box will be displayed prompting the user to select a file to upload.
<i>destination</i>	(Optional) The destination directory where the file is uploaded. The directory specified here is a relative path to the upload file directory on the file server.
<i>isrename</i>	Whether to rename the file if another file with the same name already exists.
<i>errorcode</i>	The error occurred in uploading files. Values are: 0 - Succeeded. -1 - Connection to the file server fails.

Parameter	Description
	-2 - ID error. -3 - The format of the source file is forbidden. -4 - Destination directory is invalid or the file is too large.

Return value

String.	Returns the URL of the uploaded file if it succeeds and returns an empty string if it fails. Returns NULL if any argument is NULL.
---------	---

Usage

If you want to delete an uploaded file, you must restart the file server after deleting it. Otherwise, you will fail to upload the file with the same name.

of_appeondownload

Description

Downloads the specified file from the Appeon File Server.

Syntax

appeonfileservice.of_appeondownload (string *source*)

appeonfileservice.of_appeondownload (string *source*, string *target*)

Parameter	Description
appeonfileservice	An instance of an AppeonFileService object.
<i>source</i>	The directory of the source file. The directory specified here is a relative path to the download file directory on the Appeon File Server.
<i>target</i>	(Optional) The destination directory where the file is saved. If no directory is specified here, a dialog box will be displayed prompting the user to select a location to save the file.

Return value

Long.	Return values are: 0 - The file is successfully downloaded. -1 - The file download failed or exceptions occurred. -3 - File name is empty. NULL if any argument is NULL.
-------	--

of_FileExists

Description

Reports whether the specified file exists on the Appeon File Server.

Syntax

appeonfileservice.of_FileExists (String *filename*)

Parameter	Description
appeonfileservice	An instance of the AppeonFileService object.
<i>filename</i>	A string whose value is the name of a file.

Return value

Long.

1 - The file exists.

0 - The file does not exist on the Appeon File Server.

-1 - Failed to connect to the Appeon File Server.

of_downloadfile (Obsolete)

Obsolete function

of_downloadfile is an obsolete function and will be discontinued in a future release. Please replace it with [of appeondownload](#).

of_uploadfile (Obsolete)

Obsolete function

of_uploadfile is an obsolete function and will be discontinued in a future release. Please replace it with [of appeonupload](#).

Properties for AppeonFileService object

AppeonFileService object provides the following properties (private instances) to upload and download files:

- [is_FileServerName](#) instance: Indicates the name of the file server.
- [is_LogOnParams](#) instance: Indicates the connection string for logging into the file server.
- [il_Id](#) instance: Long. Indicates the ID number returned from the file server when logon has succeeded.

is_FileServerName

Description

Indicates the IP address or machine name of the file server. This property is initialized after the of_LogOnFileServer function is called.

Return value

String.

is_LogOnParams

Description

Indicates the connection string for logging into the file server. This property is initialized after the of_LogOnFileServer function is called.

Return value

String.

il_Id

Description

Indicates the ID number returned from the file server when logon has succeeded. This property is initialized after the of_LogOnFileServer function is called. It is used by the file server to validate the client.

Return value

Long.

2.3.5 Distributed DataWindows (EAServer only)

"Distributed DataWindows" refers to the use of DataWindow/DataStore objects in a distributed environment. In a distributed PowerBuilder application, a DataWindow control at the Client can associate with a DataStore object in EAServer. The Client DataWindow control is responsible for the visual representation of data and user operations, while the DataStore object in EAServer is responsible for transactions. The state of the Client DataWindow control is synchronized with the state of the DataStore object in EAServer and vice versa, using relevant DataWindow functions.

There are two benefits to using distributed DataWindow technology with Appeon:

- Provides more scalability by separating user interface and business logic.
- Works around the Appeon-unsupported DataWindow functions by moving the functions to the server DataStore objects.

2.3.5.1 AppeonDataWindow and AppeonDataStore

AppeonDataWindow and AppeonDataStore are two standard user objects provided by Appeon. The appeondatawindow is inherited from the PowerBuilder system DataWindow control, and the appeondatastore is inherited from the PowerBuilder system DataStore object.

Why is workaround required if you use distributing DataWindows in Appeon

PowerBuilder **GetFullState**, **SetFullState**, **GetChanges** and **SetChanges** functions use BLOB (Binary Large Object) parameters for passing DataWindow or DataStore object specifications. Although Appeon supports BLOB, but it cannot directly interpret the BLOB DataWindow or DataStore object specifications. To work around the unsupported features (the BLOB parameter) in **GetFullState**, **SetFullState**, **GetChanges** and **SetChanges**, you should use *appeondatawindow* and *appeondatastore*.

Functions for AppeonDataWindow and AppeonDataStore

There are six functions provided by AppeonDataWindow and AppeonDataStore.

1. GetFullState, SetFullState, GetChanges and SetChanges functions are derived from corresponding PowerBuilder functions.
2. [AppeonGetFullStateEX](#) and [AppeonSetFullStateEX](#) functions.

2.3.5.1.1 AppeonGetFullStateEX

Description

Retrieves the complete state of the main DataWindow into a blob, excluding the information of its DropDownDataWindow.

Syntax

Long dwcontrol.AppeonGetFullStatusEX(blob *dwastblob*)

Argument	Description
dwcontrol	A reference to an appeondatawindow control and an appeondatastore object.
<i>dwastblob</i>	A variable into which the returned DataWindow will be placed.

Return value

Returns the number of rows in the DataWindow blob if it succeeds and returns -1 if an error occurs. If any argument value is NULL, the method returns NULL.

2.3.5.1.2 AppeonSetFullStateEX

Description

Applies the contents of a DataWindow blob retrieved by AppeonGetFullStateEX to a DataWindow. If the source DataWindow object matches the target DataWindow, DropDownDataWindow information of the target DataWindow will not be changed.

Syntax

Long dwcontrol.AppeonSetFullStatusEX(blob *dwastblob*)

Argument	Description
dwcontrol	An reference to an appeondatawindow control and an appeondatastore
<i>dwastblob</i>	A variable into which the returned DataWindow will be placed.

Return value

Long. Return value are:

- 1 - DataWindow objects match; old data and state overwritten.
- 2 - DataWindow objects do not match; old object, data, and state replaced.

3 - No DataWindow object associated with DataWindow control or DataStore; the DataWindow object associated with the blob is used. The value of the DataObject property remains an empty string.

2.3.5.2 Workaround steps

This section introduces the main workaround steps you should take for using distributed DataWindows in your application.

Step 1: Add the Workaround PBL and DLLs provided by Appeon to your application.

Step 2: Derive all distributed DataWindows and DataStores from *appeondatawindow* and *appeondatastore*.

appeondatastore and *appeondatawindow* are built in *appeon_workarounds.pbl*.

Step 3: Migrate n-Tier DataWindows. In PowerBuilder, deploy the server DataWindows and DataStores that are inherited from *appeondatastore* to Appeon Server.

You also need to deploy *appeondatawindow* objects and *appeondatastore* objects to the Appeon Server.

Step 4: Generate stubs and skeletons for the server DataWindows and DataStores as well as n-Tier NVOs in the application by following the instructions in [How to deploy NVO to EA Server 6.1](#).

A detailed sample for the workaround is provided in [Section 4.6.1, “Appeon GetFullState/SetFullState/GetChanges/SetChanges \(Windows only\)”](#).

2.3.5.3 Workaround limitations

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.

Limitation/ Difference in ...	Limitation/Difference Description
DataWindow styles	The workaround works with DataWindows or DataStores of all styles, except for OLE and Treeview.
Return value of the functions	The return value of Appeon SetFullState may have a different meaning from that of 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 function can return -1 and -3, but cannot return 2 and -2.
DataWindow ImportString function	If using the DW ImportString function in a distributed DataWindow environment, keep the date display format the same at the Client and Appeon Server machines. In addition, the date/time format configuration in AEM should be kept the same as that of Appeon Server.
State information initialization of	In PowerBuilder, the state information of a DataWindow/DataStore is initialized whenever you set its DataObject property. However, if using <i>appeondatawindow</i> and <i>appeondatastore</i> , the state information is

Limitation/ Difference in ...	Limitation/Difference Description
a DataWindow/ DataStore	initialized only when you change the DataObject property to a different DataWindow object.
Truncation of characters in certain cases	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 are truncated, but in PowerBuilder the extra characters are preserved.
Un-modified or modified data	When calling PowerBuilder GetFullState and GetChanges, changed (but not accepted) data in a DataWindow control is treated as un-modified data, but if using the Appeon <i>appeondatawindow</i> and <i>appeondatastore</i> , changed (but not accepted) data is treated as modified data.

2.3.6 Appeon Labels

Overview

Appeon Labels provided in the Appeon Workarounds PBL can reduce the interactions between the client and Appeon Server, thus boosting the performance of Appeon applications.

The Appeon Labels associated functions are contained in the *appeon_nvo_db_update* object in *appeon_workarounds.pbl*.

Appeon Labels and associated functions

Label Name	Label Associated Function	Description
Appeon Commit/Rollback Label	<i>of_autocommitrollback</i>	Automatically commits or rolls back the first database operation statement after the label.
Appeon Commit Label	<i>of_autocommit</i>	Automatically commits the first database operation.
Appeon Rollback Label	<i>of_rollback</i>	Automatically rolls back the first database operation statement if the operation fails.
Appeon Queue Labels	<i>of_startqueue</i> <i>of_commitqueue</i>	The Appeon Queue Labels are designed for use when it is necessary to execute numerous database operation statements on Appeon Server, and the returned values of the statements are not validated or used. Using the pair of labels can dramatically reduce the number of client-server interactions.
Appeon Immediate Call Label	<i>of_imdcall</i>	Immediately commits a database operation.
Appeon Update Label	<i>of_update</i>	Reduces the number of interactions with the server caused by "interrelated updates".

Usage

For detailed instructions on how to take advantages of Appeon Labels to improve performance of the application, please refer to the Section 6.5, “Technique #4: grouping multiple server calls with Appeon Labels” in *Appeon Performance Tuning Guide*.

2.3.6.1 Appeon Commit/Rollback Label

Description

The Appeon Commit/Rollback Label is used to automatically commit or rollback the first database operation statement after the label.

Controls

appeon_nvo_db_update object

Associated functions

of_autocommitrollback

Syntax

objectname.of_autocommitrollback()

Argument	Description
objectname	An instance of the appeon_nvo_db_update object.

Return value

None.

Usage

With the Appeon Commit/Rollback Label, the database operation statement will be sent to the Appeon Server. The server will automatically commit (or roll back) the statement according to the execution result. If the execution succeeds, the result will be committed; if the execution fails, the result will be rolled back.

The first Commit or Rollback statement after the Appeon Commit/Rollback Label will not be submitted to the Appeon Server. Therefore, there must be no more than one database operation statement between the label and the first Commit or Rollback statement. For example, the IF statement should not contain database operation statements, since the executed result will not be committed to the database.

```

gnv_appeonDbLabel.of_autocommitrollback()
UPDATE tab_a .....
IF SQLCA.SQLCODE = 0 THEN
    ..... // non-database related bussiness logic
    COMMIT;
ELSE
    ..... // non-database related bussiness logic
    ROLLBACK;
    .....
END IF
    
```

There must be database related operations after the label.

There must be no labels between the Appeon Rollback Label and the first Commit or Rollback statement.

2.3.6.2 Appeon Commit Label

Description

The Appeon Commit Label is used to automatically commit the first database operation.

Controls

appeon_nvo_db_update object

Associated functions

of_autocommit

Syntax

objectname.of_autocommit()

Argument	Description
objectname	An instance of the appeon_nvo_db_update object.

Return value

none.

Usage

After the label there must be database operations.

With Appeon Commit Label, Appeon Server does not validate the execution result of the database operation statement. Instead, the server automatically commits the statement regardless of the execution result.

The first Commit statement after the Appeon Commit Label will not be submitted to the Appeon Server , however, the first Rollback statement will be submitted to the server. Therefore, there should be no more than one database operation between the label and the first Commit statement. For example, the IF statement should not contain database related business logic, since the executed result will not be committed to the database.

```

gmv_appeonDbLabel.of_autocommit()
SELECT.....INTO .....FROM tab_a;
IF SQLCA.SQLCODE = 0 THEN
    ..... // non-database related business logic
ELSE
    ..... // non-database related business logic
END IF
COMMIT;
    
```

There must be no labels between the Appeon Commit Label and the first Commit statement.

2.3.6.3 Appeon Rollback Label

Description

The Appeon Rollback Label is used to automatically roll back the first database operation statement if the operation fails.

Controls

appeon_nvo_db_update object

Associated functions

of_rollback

Syntax

objectname.of_rollback()

Argument	Description
objectname	An instance of the appeon_nvo_db_update object.

Return value

None.

Usage

After the label there must be database operations.

With the Appeon Rollback Label, Appeon Server only commits an unsuccessful database operation.

The first Rollback statement after the Appeon Rollback Label will not be submitted to the Appeon Server if the execution fails. Therefore, there should be no more than one database operation between the label and the first Commit or Rollback statement.

There must be no labels between the Appeon Rollback Label and the first Rollback statement.

Code Example

```

gmv_appeonDbLabel.of_rollback()
IF dw_1.update() <> 1 THEN
    ROLLBACK ;
    ..... // non-database related business logic
END IF
    
```

2.3.6.4 Appeon Queue Labels

Description

Appeon Queue Labels consist of the Appeon Start Queue Label and the Appeon Commit Queue Label. The Appeon Queue Labels are designed for use when it is necessary to execute numerous database operation statements on Appeon Server, and the returned values of the statements are not validated or used. Using the pair of labels can dramatically reduce the number of client-server interactions.

Controls

appeon_nvo_db_update object

Associated functions

of_startqueue, of_commitqueue

Syntax

objectname.of_startqueue ()

objectname.of_startqueue({integer stopmode})

Argument	Description
<i>objectname</i>	An instance of the <code>appeon_nvo_db_update</code> object.
<i>stopmode</i>	0 - continue executing the remaining SQL scripts when an error occurs; 1 - stop executing the remaining SQL scripts when an error occurs. Note: The <code>of_startqueue</code> function without this argument is preserved for compatibility.

objectname.of_commitqueue()

Argument	Description
<i>objectname</i>	An instance of the <code>appeon_nvo_db_update</code> object.

Return value

none.

Usage

Appeon Queue Labels must be used in the same field.

All the database operations in the labels will be submitted to the Appeon Server.

If there are multiple Appeon Commit Queue Labels used together with an Appeon Start Queue Label, only the first Appeon Commit Queue Label that is executed will be effective. Other Appeon Commit Queue Labels will be ignored.

With the `stopmode` argument, users can choose to continue running or return immediately when an error occurs in the database syntax operation in the queue.

In the Appeon Queue Labels, the `SELECT` statement cannot be used in the condition statements. The following example is incorrect.

Incorrect Example

```

nv_appeonDbLabel.Of_startqueue()
IF.....THEN
  SELECT STATEMENT 1
else
  SELECT STATEMENT 2
END IF
gmv_appeonDbLabel.Of_commitqueue()
    
```

In the Appeon Queue Labels, script that stops the execution of another script cannot be included in some events of the DataWindow object, For example, in the following events of DataWindow, the Return statement should not be used: the RetrieveStart event, the RetrieveEnded event, the RowFocusChanged event, the UpdateStart event, the UpdateEnd event, and etc.

For every RETURN statement, there must be an database operation statement or unexpected errors occur.

You can open a cursor in the Appeon Queue Labels.

Using multiple Appeon Queue Labels

Appeon Queue Labels can be embedded in other Appeon Queue Labels. However, only the outer Appeon Queue Labels take effect.

Using non-queue labels together with Appeon Queue Labels

When there are multiple non-queue labels embedded in the Appeon Queue Labels, only the first non-queue label takes effect.

When the other Appeon Labels is embedded in Appeon Queue Labels, the format should be the same as the following code example. Please note that only Commit or Rollback statements are involved in the condition statements.

Code Example

```

nv_appeonDbLabel.of_startqueue()
dw_1.update()
gmv_appeonDbLabel.of_autocommitrollback() // the label takes effect
gmv_appeonDbLabel.of_imdcall() // The label takes no effect
IF dw_2.update() = 1 THEN
    COMMIT;
ELSE
    ROLLBACK;
END if

nv_appeonDbLabel.of_startqueue(1) // Stop immediately when an error occurs
dw_1.update()
gmv_appeonDbLabel.of_autocommitrollback() // the label takes effect
gmv_appeonDbLabel.of_imdcall() // The label takes no effect
if dw_2.update() = 1 THEN
    COMMIT;
ELSE
    ROLLBACK;
END if

nv_appeonDbLabel.of_autocommitrollback()
UPDATE tab_a.....
if SQLCA.SQLCODE = 0 THEN
    COMMIT;
ELSE
    ROLLBACK;
END IF
INSERT tab_b.....
    COMMIT;
gmv_appeonDbLabel.of_commitqueue()
    
```

2.3.6.5 Appeon Immediate Call Label

Description

The Appeon Immediate Call Label is used to immediately commit a database operation.

Controls

appeon_nvo_db_update object

Associated functions

of_imdcall

Syntax

objectname.of_imdcall()

Argument	Description
objectname	An instance of the appeon_nvo_db_update object.

Return value

none.

Usage

Appeon Immediate Call Label cannot be used alone, it must be used in Appeon Queue Labels.

With the Appeon Immediate Call Label, the first database operation statement will be sent to the server and executed immediately.

Code Example

```

gmv_appeonDbLabel.of_startretrievequeue()
dw_1.retrieve()
gmv_appeonDbLabel.of_imdcall()
SELECT .....INTO :var_1,:var_2...
IF var_1 > 0 THEN
    para = "ok"
ELSE
    para = "false"
END IF
dw_2.retrieve(para)
gmv_appeonDbLabel.of_endretrievequeue()
    
```

2.3.6.6 Appeon Update Label

Description

The Appeon Update Label is used to reduce the number of interactions with the server caused by "interrelated updates".

Controls

appeon_nvo_db_update object

Associated functions

of_update

Syntax

objectname.of_update (integer transactionflag, powerobject *obj_1*, powerobject *obj_2*)

objectname.of_update (integer transactionflag, powerobject *obj_1*, powerobject *obj_2*, powerobject *obj_3*)

objectname.of_update (integer transactionflag, powerobject *obj_1*, powerobject *obj_2*, powerobject *obj_3*, powerobject *obj_4*)

objectname.of_update (powerobject *obj*)

objectname.of_update (powerobject *obj_1*, powerobject *obj_2*)

objectname.of_update (powerobject *obj_1*, powerobject *obj_2*, powerobject *obj_3*)

objectname.of_update (powerobject *obj_1*, powerobject *obj_2*, powerobject *obj_3*, powerobject *obj_4*)

Argument	Description
objectname	An instance of the <code>appeon_nvo_db_update</code> object.
transactionflag	0 - transaction is automatically committed; 1 - transaction is not automatically committed. Note: the <code>of_update()</code> function without this argument is preserved for compatibility.
<i>obj</i>	The name of the DataWindow, DataStore or DataWindowChild that needs to update.
<i>obj_1</i>	The name of the DataWindow, DataStore or DataWindowChild that needs to update.
<i>obj_2</i>	The name of the DataWindow, DataStore or DataWindowChild that needs to update.
<i>obj_3</i>	(optional) The name of the DataWindow, DataStore or DataWindowChild that needs to update.
<i>obj_4</i>	(optional) The name of the DataWindow, DataStore or DataWindowChild that needs to update.

Return value

Integer.	Return values are: 1 - Succeed in update -101 - Fail to update the first DataWindow/DataStore/DataWindowChild -102 - Fail to update the second DataWindow/DataStore/DataWindowChild -103 - Fail to update the third DataWindow/DataStore/DataWindowChild -104 - Fail to update the fourth DataWindow/DataStore/DataWindowChild
----------	---

Usage

The update operations of the DataWindows, DataStores, or DataWindowChild will be submitted to the Appeon Server together. If the operation of a DataWindow, DataStore or DataWindowChild fails, Appeon Server will stop processing the update operation. Users can also use `transactionflag` argument to control whether to commit or rollback the Database update.

The following script has the same function. However, by using the Appeon Update Label the number of client-server interactions is reduced to one.

Using Appeon Update Label

```
l_rtn = gnv_appeonDb.of_update(0, dw_1,dw_2)
IF l_rtn = 1 THEN
    MessageBox("Success","Update success!")
ELSEIF l_rtn= -102 THEN
    MessageBox("Failure","Update all failure!")
ELSE
    MessageBox("Failure","Update dw_1 failure!")
END IF
```

Without using Appeon Update Label

```
IF dw_1.Update() = 1 THEN
    IF dw_2.Update() = 1 THEN
        COMMIT;
        MessageBox("Success","Update success!")
    ELSE
        ROLLBACK;
        MessageBox("Failure","Update all failure!")
    END IF
ELSE
    ROLLBACK;
    MessageBox("Failure","Update dw_1 failure!")
END IF
```

2.3.7 Calling EJB Component

In order to implement calling EJB components for applications deployed with Appeon, Appeon provides a customized object ([EJBObject object](#)), a DLL (EonEJBClient.dll) and a bridge ([Appeon Bridge](#)) in Appeon workarounds. With the two features you can call an EJB component either in the PowerBuilder application or in the deployed application. The solution is not suitable for applications that are deployed to Appeon Server for .NET.

Compared to the EJB solution of PowerBuilder, Appeon EJB solution can support more complex parameters such as Structure.

EonEJBClient.dll

Connects to and communicates with the Appeon Bridge. The DLL should exist on the machine on which the Client/Server application runs.

2.3.7.1 Appeon Bridge

Overview

Appeon Bridge is a standard J2EE-compliant Web application that can be deployed to any J2EE compliant application server regardless of whether EJB components exist on the server. It functions as the medium between the clients and EJB components, and can be deployed by installing the appeonbridge.war file, which is located in *%AppeonServer%\plugin\appeonbridge*.

Bundle EJB Client Proxies and Application Client to Appeon Bridge

Before deploying Appeon Bridge to a J2EE-compliant application server, you must bundle EJB client proxies (JAR file) and Application clients (JAR file) to Appeon Bridge to

implement the communication to EJB components. An Application client (for example, websphere.jar for WebSphere) is only required if the J2EE-compliant application server and Java Web server are different (for example, using EAServer as the J2EE-compliant application server and WebSphere as the Java Web Server).

Two ways of bundling the EJB client proxies and Application Clients to Appeon Bridge:

1. Add the proxies and clients to the lib directory in the appeonbridge.war file. Since the appeonbridge.war is a ZIP file, you can check the directory in the WAR file with a third party tools such as WinZip. For more details, take EAServer as an example, shown as below:

- Locate to *%EAServer%\appeon\plugin\appeonbridge* folder and open *appeonbridge.war* with the third party tool such as WinZip.
- Locate to *%EAServer%\deploy\ejbjars\md5generator\com\desta* folder.
- Add full *md5generator\com\desta* folder to *appeonbridge.war\WEB-INF\classes\com* directory.
- Login on EAServer console and locate to **EAServer Manager > Local Server > Web Applications**.
- Delete *appeonbridge*.
- Deploy *%EAServer%\appeon\plugin\appeonbridge\appeonbridge.war*.
- Run your application and test again.

2. Move the EJB Client Proxies and Application Clients to the Lib directory in the Java Web server, and add the directory, where the proxies and clients are stored, to the CLASSPATH environment variable of the machine that hosts the Java Web Server.

Deploy Appeon Bridge to J2EE-compliant application server

Deploying Appeon Bridge (*appeonbridge.war*) is the same as deploying the file server. Please refer to the corresponding **Deploying the Appeon File Server** section under [Configuring and deploying the file server](#). (Or you can refer to documents provided by the corresponding server vendors for the deployment instructions, as deploying Appeon Bridge is the same as deploying any other applications.)

2.3.7.2 EJLObject object

Appeon EJLObject object implements the interaction between client and Appeon Bridge. To use this object, you need to load **appeon_workaround.pbl** into your application and make sure the **EonEJBClient.dll** is stored on the machine where the Client/Server application will be run.

EJLObject object provides the following functions to perform the relevant actions:

Function	Description
ConnectServer function	Connecting and disconnecting to Appeon Bridge

Function	Description
DisConnection function	
LookUpJndi function	Obtaining the home interface of an EJB component
CreateRemoteInstance function	Creating and destroying the instance for an EJB component
DestroyRemoteInstance function	
Registering parameter functions	Registering parameters
Invoking component functions	Invoking EJB components
InitLocalLanguage function	Setting the language of the error message

A detail [code example](#) of how to call these functions is also provided.

2.3.7.2.1 ConnectServer function

Description

Connects a client application to Appeon Bridge.

Syntax

`EJBObject.ConnectServer(string url, string properties[])`

Parameter	Description
EJBObject	A reference of an EJBObject.
<i>url</i>	URL with port, where Appeon Bridge is installed.
<i>properties[]</i>	Properties of Appeon Bridge.

Return value

String.

Returns empty string ("") if it succeeds.

2.3.7.2.2 DisConnection function

Description

Disconnects a client application from Appeon Bridge.

Syntax

`EJBObject.DisConnection()`

Parameter	Description
EJBObject	A reference of an EJBObject.

Return value

String.

Returns empty string ("") if it succeeds.

2.3.7.2.3 LookUpJndi function

Description

Obtains the home interface of an EJB component in order to create an instance for the component.

Syntax

EJBObject.lookupjndi (string *jndiname*, ref long *objid*)

Parameter	Description
EJBObject	A reference of an EJBObject.
<i>jndiname</i>	The JNDI name of the EJB component.
<i>objid</i>	The handle to the EJB home interface.

Return value

String.

Returns empty string ("") if it succeeds.

2.3.7.2.4 InitLocalLanguage function

Description

Sets the language of the error message in PowerBuilder IDE.

Syntax

EJBObject.initlocallanguage (long *nlocalcode*)

Parameter	Description
EJBObject	A reference of an EJBObject.
<i>nlocalcode</i>	A long value representing different languages. 0 - English (default) 1 - Japanese 2 - Korean 3 - Simplified Chinese 4 - Traditional Chinese

Return value

Long.

2.3.7.2.5 Invoking component functions

Invokeretblob	Invokeretblobarray	Invokeretbool	Invokeretboolarray
Invokeretchar	Invokeretchararray	Invokeretdate	Invokeretdatearray
Invokeretdatetime	Invokeretdatetimearray	Invokeretdouble	Invokeretdoublearray
Invokeretint	Invokeretintarray	Invokeretlong	Invokeretlongarray
Invokeretreal	Invokeretrealarray	Invokeretstring	Invokeretstringarray
Invokerettime	Invokerettimearray	Invokeretuint	Invokeretuintarray
Invokeretulong	Invokeretulongarray	Invokeretstru	Invokeretstruarray
Invokeretvoid			

Description

Invoke an EJB component which returns a particular data type. All methods will share the same parameters, syntax and return value.

Syntax

string EJBObject.Invokeblob(long *objid*, string *methodname*, boolean *autoremove*, ref blob *retval*)

Parameter	Description
EJBObject	A reference of an EJBObject.
<i>objid</i>	The handle to the component method.
<i>methodname</i>	The name of the invoking component method.
<i>autoremove</i>	Unsupported. Input false.
<i>retval</i>	The return value of invoking the EJB component method. This parameter does not provided in the Invokeretvoid method. The data type of the <i>retval</i> argument keeps the consistence with the data type used in the invoking methods, except the Invokeretstru and Invokeretstruarray, the data type of the <i>retval</i> arguments for the two methods are both blob.

Return value

String.

Returns empty string ("") if it succeeds.

Usage

Variables cannot be null for structure and array.

For a structure to be registered, variables can be:

1. char, string, boolean, int, unit, long, ulong, real, double, datetime, date or time.

2. an array of the above types. The maximum dimension is 3.
3. a structure or a 1-dimensional structure array. And the array must be a fixed array.

For a return value of structure type, variables can be:

1. char, string, boolean, blob, int, unit, long, ulong, real, double, datetime, date or time.
2. a multidimensional array of the above types.
3. a structure or a 1-dimensional structure array. And the array must be a fixed array.

2.3.7.2.6 Registering parameter functions

Register functions are provided to register parameters with different data type. Except RegStruct and RegstructArray, all functions will share the same parameters, syntax, return value.

RegChar	RegCharArray	RegDate	RegDateArray
RegDateTime	RegDateTimeArray	RegDouble	RegDoubleArray
RegInt	RegIntArray	RegLong	RegLongArray
RegReal	RegRealArray	RegString	RegStringArray
RegBlob	RegBlobArray	RegBool	RegBoolArray
RegTime	RegTimeArray	RegUInt	RegUIntArray
RegULong	RegULongArray	RegStruct	RegStructArray

Description

Registers a parameter with a certain data type. Syntax below takes RegBlob as an example.

Syntax

EJLObject.RegBlob(blob *data*)

Parameter	Description
EJLObject	A reference of an EJLObject.
<i>data</i>	The parameter to be registered. Its name can be user-defined, but its type must be consistent with the data type specified in the Register method.

Return value

String.

Returns empty string ("") if it succeeds.

RegStruct and RegStructArray functions

Description

Registers a structure or structure array. The two functions will share the same parameters, syntax, return value. Syntax below takes regstructarray as an example.

Syntax

EJLObject.regstructarray (any *data []*, string *javaclassname*, readonly classdefinition *cdef*)

Parameter	Description
EJLObject	A reference of an EJLObject.
<i>data</i>	The parameter to be registered and the name is user-defined.
<i>javaclassname</i>	The name of the corresponding Java class in application server.
<i>cdef</i>	ClassDefinition property of structure.

Return value

String.

Returns empty string ("") if it succeeds.

Usage

Variables cannot be null for structure and array.

For a structure to be registered, variables can be:

1. char, string, bool, int, unit, long, ulong, real, double, datetime, date or time.
2. an array of the above types. The maximum dimension is 3.
3. a structure or a 1-dimensional structure array. And the array must be a fixed array.
4. (Only for RegStruct method) For the javaclassname parameter, input the full name of the Java class on EJB server corresponding to the structure you defined in PowerBuilder. For example, a.b.c.d.myclassName.

2.3.7.2.7 CreateRemoteInstance function

Description

Creates the instance for an EJB component.

Syntax

EJLObject.CreateRemoteInstance(string *jndiname*, string *homename*, string *methodname*, ref long *beanid*)

Parameter	Description
EJLObject	A reference of an EJLObject.
<i>jndiname</i>	The JNDI name of the EJB component.
<i>homename</i>	The name of the home interface of an EJB component.
<i>methodname</i>	The name of the method.

Parameter	Description
<i>beanid</i>	The handle to the EJB component.

Return value

String.

Returns empty string ("") if it succeeds.

2.3.7.2.8 DestroyRemoteInstance function

Description

Destroys the instance for an EJB component.

Syntax

EJBObject.DestroyRemoteInstance(long *objid*)

Parameter	Description
EJBObject	A reference of an EJBObject.
<i>objid</i>	The handle to the EJB component.

Return value

String.

Returns empty string ("") if it succeeds.

2.3.7.2.9 Code Example

This section gives you a detail example of how to invoke EJB components in PowerBuilder.

Step 1: Declare object and connect to Appeon Bridge.

```

ejbobject lo_ejb
long ll_bean1 = 0
long ll_homeHandle = 0
string ls_prop[4]
string ls_serurl,ls_msg
ls_serurl = "http://192.0.2.182:8080/appeonbridge/Dispatch"
ls_prop[1]= "applicationA"
ls_prop[2]= "javax.naming.Context.INITIAL_CONTEXT_FACTORY=
'com.sybase.ejb.InitialContextFactory'"
ls_prop[3]= "javax.naming.Context.PROVIDER_URL='iiop://192.0.2.182:9989'"
ls_prop[4]= "username=jagadmin"
ls_prop[5]= "password="
ls_msg = lo_ejb.connectserver(ls_serurl, ls_prop)

if ls_msg = "" then
    MessageBox("connectserver succeed!","srvId =" +string(lo_ejb.il_srvid))
else
    MessageBox("wrong!",ls_msg)
end if
    
```

Step 2: Obtain the home interface of an EJB component in a J2EE-compliant server.

```

ls_msg = lo_ejb.lookupjndi ("TestSBeanLess",ref ll_homeHandle)
    
```

Step 3: Create instance for EJB component.

```
ls_msg = lo_ejb.createremoteinstance("AllDataType","AllDataType","create", ref ll_bean1)
```

Step 4: Invoke the EJB components method.

Example one:

```
string ls_msg char c_val c_val = 'a' lo_ejb.regchar(c_val) char retval
ls_msg = lo_ejb.invokeretchar (ll_bean1, "getChar",true, ref retval)
```

Example two:

```
string ls_msg
boolean b_val
b_val = false
lo_ejb.regbool(b_val)
boolean retval
ls_msg = lo_ejb.invokeretbool (ll_bean1, "getBoolean", true, ref retval)
```

Step 5: Destroy the instance.

```
ls_msg = lo_ejb.destroyremoteInstance(ll_homeHandle)
If ls_msg <> "" Then
    MessageBox("Destroy Remote Instance Failed",ls_msg)
Return
End If
```

2.3.7.3 Appeon requirements for EJB development

1. Appeon Bridge maps datatypes (except structure) between Java and PowerBuilder is shown as below.

PowerBuilder data types	Java data types
Char	char
String	String
Boolean	boolean
Int	short
Unit	int
Long	int
Ulong	long
LongLong	long
Real	float
Double	double
Decimal	java.math.BigDecimal
Number	double
Timestamp	java.sql.Timestamp
Datetime	java.sql.Timestamp
Date	java.util.Date
Time	java.sql.Time
Blob	byte[]

2. With Appeon EJB solution, Structure data can be passed when invoking EJB components. To implement this, you need to define a Java class in the EJB components. There are two necessary elements in Java Class: 1) **private static String PBMap[]** and 2) **implementing java.io.Serializable interface**. In PBMap array you need to map the members with the identical order and datatype to a PowerBuilder Structure.

Following is an example of defining a Java class (please note that the member variables should be in lower case.)

```

package test;
import java.io.Serializable;

public class Simple implements Serializable {
    private short l_int;
    private boolean b_bool;
    private String s_string;

    private static String PBMap[] = {"l_int", "b_bool", "s_string"};

    public String[] getPMap() {
        return PBMap;
    }

    public boolean isB_bool() {
        return b_bool;
    }

    public short getL_int() {
        return l_int;
    }

    public String getS_string() {
        return s_string;
    }

    public void setB_bool(boolean b_bool) {
        this.b_bool = b_bool;
    }

    public void setL_int(short l_int) {
        this.l_int = l_int;
    }

    public void setS_string(String s_string) {
        this.s_string = s_string;
    }
}

```

2.3.8 Calling .NET/COM server components (.NET only)

Applies to

Appeon Server for .NET.

Supported server component types

- .NET components: All valid .NET components, including executable files (.exe) and DLL files (.dll).

Supported parameters: primitive type parameters, such as int, vlong char, and boolean.

Non-primitive type parameters, such as class, are unsupported.

Supports reference parameters.

- COM components: COM/COM+ components

Supported parameters: primitive type parameters, such as byte, int, long, and float.

Supports reference parameters.

Description

To call .NET/COM components, Appeon provides a non-autoinstantiated NVO - [AppeonDotNetComponent](#) - as the proxy object to call the server-side components. The user can either create a local instance of `AppeonDotNetComponent` for each server component, or directly use an existing instance of `AppeonDotNetComponent`. The user must specify the properties of the instance, such as the component type, the library name and the class name, to bind the instance with the server component, or change the instance properties during runtime to dynamically bind with a different component.

It provides a universal single interface and a set of parameters which determines which component and methods will be called.

Note:

The script to call `AppeonDotNetComponent` takes effect only after the PowerBuilder application is deployed, and has no effect when the PowerBuilder application is run.

Register

The COM component must be registered using the `regsvr32` tool.

Storage location

The components must reside in the `%appeon%/AEM/components` folder on the Appeon Server machine. You only need to place the `.tlb` library files and `.dll` files of the COM components to the folder. `%appeon%` indicates the installation directory of Appeon Server.

2.3.8.1 AppeonDotNetComponent object

2.3.8.1.1 Properties

Properties for `AppeonDotNetComponent`.

Properties	Type	Description
ComponentType	String	The type of the component to be called. "1" indicates a .NET Assembly to be called. "2" indicates an unmanaged-code COM component to be called. "3" indicates a managed-code COM component to be called. "4" indicates a built-in Appeon Workaround .NET Assembly to be called.

Properties	Type	Description
TypeLib	String	The name of the component library. Appeon Server uses this name to find the component.
ClassDescript	String	The class name.
ReturnValue	Any	Read-only. The return value of functions. The value and value type varies from function to function.
ErrorText	String	Read-only. The error message of functions. The message varies from function to function. Empty string if no error.

2.3.8.1.2 Functions

of_execinterface

Description

Calls the function in the binding component.

Syntax

of_execinterface (string *interfacename* {, ref any *paralist*[]})

Argument	Description
<i>interfacename</i>	The name of the component function.
<i>paralist</i> []	Optional. Arrays of Any type. Specifies the parameter arrays for the component function.

Return value

Long.

0 – Call succeeded. Gets the value from the ReturnValue property of the proxy object.

-1 – Call failed. Gets the error message from the ErrorText property of the proxy object.

Usage

Before calling this function, use the proxy object properties to bind with the target component. If the component function contains no parameters, simply specify the function name. If the component function contains parameters, define an Any type array before the call, then place the argument to the array, finally pass the array as the second parameter of the function.

Examples

Example 1: the interface contains no parameters.

```

AppeonDotNetComponent lu_apf

lu_apf = create AppeonDotNetComponent lu_apf
lu_apf.ComponentType = "2"
lu_apf.TypeLib = "test.dll"
lu_apf.ClassDescript = "testclass"

ll_ret = lu_apf.of_ExecInterface("test")
    
```

Example 2: the interface contains four parameters, their types are: string, int, long, and string.

```
// Define the array variable
AppeonDotNetComponent lu_apf
any la_1[]

la_1[1] = "Appeon"
la_1[2] = 100
la_1[3] = 256
la_1[4] = "Sybase"

lu_apf = create AppeonDotNetComponent lu_apf
lu_apf.ComponentType = "1"
lu_apf.TypeLib = "testdotnet.dll"
lu_apf.ClassDescript = "interface1"

ll_ret = lu_apf.of_ExecInterface("test_dotnet", la_1)
```

2.3.8.1.3 Events

Constructor

Description

It will be triggered when you create an instance from a user-defined proxy object inherited from AppeonDotNetComponent.

Event ID

pbm_constructor

Argument

None

Return values

Long

Usage

Do not write scripts to this event directly, because the scripts will be abandoned when the application is deployed. Instead, define and inherit an object from AppeonDotNetComponent, and add the scripts to the Constructor event of the new object. The usage is the same as that of the PowerBuilder system object. For example, you can initialize the property value of this event, or define relevant information objects.

Destructor

Description

It will be triggered when you explicitly call Destroy to destroy the instance of a user-defined proxy object inherited from AppeonDotNetComponent.

Event ID

pbm_destructor

Argument

None

Return value

Long

Usage

Do not write scripts to this event directly, because the scripts will be abandoned when the application is deployed. Instead, define and inherit an object from `AppeonDotNetComponent`, and add the scripts to the `Destructor` event of the new object. The usage is the same as that of the PowerBuilder system object. For example, you can add scripts to release the instances related with the proxy object.

2.3.8.1.4 Code Examples

Example 1:

```
long lRet
int iResult
string strError
appeondotnetcomponent comcaller

// create appeondotnetcomponent instance and set properties

comcaller = create appeondotnetcomponent
comcaller.componenttype = '1'
comcaller.typelib = 'DotNetDll.dll'
comcaller.classdescript = 'DotNetClass'

// invoke component method

lRet = comcaller.of_execinterface("GetInt")
if lRet = 0 then
iResult = comcaller.ReturnValue
else
strError = comcaller.ErrorText
end if

// Bind with a component and call the component method

comcaller.componenttype = '2'
comcaller.typelib = 'comfordotnet.dll'
comcaller.classdescript = 'ifdotnet'
comcaller.of_execinterface("getint")
```

Example 2:

```
// Call a method with reference parameters

any paralist[]
long refparam1 = 32764
long refparam2 = 32763

paralist[1] = refparam1
paralist[2] = refparam2

comcaller.componenttype = '1'
comcaller.typelib = 'DotNetDll.dll'
comcaller.classdescript = 'DotNetClass'
comcaller.of_execinterface("GetIntAndRefInt",paralist)

refparam1 = paralist[1]
rafparam2 = paralist[2]
```


2.3.9 Calling Web Service

Description

To call Web services, Appeon provides a non-autoinstantiated NVO – AppeonWebServiceComponent – as the proxy object to call Web services. The user can either create a local instance of AppeonWebServiceComponent for Web service, or directly use an existing instance of AppeonWebServiceComponent. The user must specify the properties of the instance, such as the proxy type, the Web service location and the class name, to bind the instance with the Web service, or change the instance properties during runtime to dynamically bind with a different Web service.

It provides a universal single interface and a set of parameters which determines which Web service and methods will be called.

Note:

1. The script to call AppeonWebServiceComponent takes effect on both the Web and the mobile, and has no effect in the PowerBuilder application.
2. You need to restart the IIS after deploying a Web service with a new method.

2.3.9.1 appeonwebservicecomponent object

2.3.9.1.1 Properties

Properties for appeonwebservicecomponent.

Properties	Type	Description
CallType	String	The proxy type of the Web service to be called. "1" indicates the proxy type is Dynamic Proxy. "2" indicates the proxy type is DLL Proxy. Appeon for J2EE editions support only CallType="1".
ProxyDllOrUrl	String	If CallType="1", it indicates the URL of the Web service to be called; If CallType="2", it indicates the DLL name of the proxy used by the Web service to be called
ClassDescript	String	If CallType="1", it can be null; If CallType="2", it indicates class name.

Properties	Type	Description
ReturnValue	Any	Read-only. The return value of functions. The value and value type varies from function to function.
ErrorText	String	Read-only. The error message of functions. The message varies from function to function. Empty string if no error.

2.3.9.1.2 Functions

of_callwebservice

Description

Calls the function in the binding Web service.

Syntax

of_callwebservice (value string *methodname* {, ref any *paralist*[]})

Argument	Description
methodname	The name of the Web service method.
<i>paralist</i> []	Optional. Arrays of Any type. Specifies the parameter arrays for the Web service function.

Return value

Long.

0 – Call succeeded. Gets the value from the ReturnValue property of the proxy object.

-1 – Call failed. Gets the error message from the ErrorText property of the proxy object.

Usage

Before calling this function, use the proxy object properties to bind with the target Web service. If the Web service method contains no parameters, simply specify the method name. If the Web service method contains parameters, define an Any type array before the call, then place the argument to the array, finally pass the array as the second parameter of the function.

Examples

Example 1: the interface contains no parameters.

```

appeonwebservicecomponent caller
caller= create appeonwebservicecomponent
caller.calltype="1"
caller.proxydllorurl="http://localhost/webservice.asmx"
caller.classdescript=""
integer IRet
IRet=caller.of_callwebservice("GetUserName")
    
```

Example 2: the interface contains two parameters, their types are any.

```
any paralist[]
appeonwebservicecomponent caller
caller=create appeonwebservicecomponent
caller.calltype="1"
caller.proxydllorurl="http://localhost/webservice.asmx"
caller.classdescript=""
paralist[1]="param1"
paralist[2]="param2"
IRet=caller.of_callwebservice("GetUserName",paralist)
```

2.3.9.1.3 Events

Constructor

Description

It will be triggered when you create an instance from a user-defined proxy object inherited from AppeonWebServiceComponent.

Event ID

pbm_constructor

Argument

None

Return values

Long

Usage

Do not write scripts to this event directly, because the scripts will be abandoned when the application is deployed. Instead, define and inherit an object from AppeonWebServiceComponent, and add the scripts to the Constructor event of the new object. The usage is the same as that of the PowerBuilder system object. For example, you can initialize the property value of this event, or define relevant information objects.

Destructor

Description

It will be triggered when you explicitly call Destroy to destroy the instance of a user-defined proxy object inherited from AppeonWebServiceComponent.

Event ID

pbm_destructor

Argument

None

Return value

Long

Usage

Do not write scripts to this event directly, because the scripts will be abandoned when the application is deployed. Instead, define and inherit an object from `ApeonWebServiceComponent`, and add the scripts to the Destructor event of the new object. The usage is the same as that of the PowerBuilder system object. For example, you can add scripts to release the instances related with the proxy object.

2.3.9.1.4 Code Examples

Example 1:

```
long IRet
int iResult
string strError
apeonwebservicecomponent caller

// create apeonwebservicecomponent instance and set properties

caller=create apeonwebservicecomponent

// if proxy type is DynamicProxy, the value of calltype is 1;
// if proxy type is DllProxy type, the value of calltype is 2.
// Apeon for J2EE editions support only CallType="1".
// Apeon for .NET edition supports both.

caller.calltype="1"

// since proxy type is DynamicProxy,the value of proxydllorurl is the address of
// the webservice to be called;or else, the value of proxydllorurl is the name
// of DLL.
caller.proxydllorurl="http://localhost/webservice.asmx"
caller.classdescript=""

// Invoke webservice method

IRet=caller.of_callwebservice("GetUserName")
if IRet=0 then
    iResult=caller.ReturnValue
else
    strError=caller.ErrorText
end if
```

Example 2:

```
long IRet
int iResult
string strError
any paralist[]
apeonwebservicecomponent caller

// create apeonwebservice instance and set properties

caller=create apeonwebservicecomponent
caller.calltype="1"
caller.proxydllorurl="http://localhost/webservice.asmx"
caller.classdescript=""
paralist[1]="param1"
paralist[2]="param2"

// invoke webservice method

IRet=caller.of_callwebservice("GetUserName",paralist)
```

Example 3:

```

long IRet
int iResult
string strError
appeonwebservicecomponent caller

// create appeonwebservicecomponent instance and set properties

caller=create appeonwebservicecomponent
caller.calltype="2"
caller.proxydllorurl="DotDllForJava"
caller.classdescript="MyJavaWebService"

// invoke webservice method

IRet=caller.of_callwebservice("GetUserName")
if IRet=0 then
    iResult=caller.ReturnValue
else
    strError=caller.ErrorText
end if
    
```

2.4 Mobile Device API

The APIs listed in this section are specific for the mobile environment including iOS and Android. These APIs allow developers to take advantage of the iOS SDK APIs and Android SDK APIs to access the native features of the target mobile device.

Note: For each category of APIs, there are two objects with almost the same name -- one with the "ex" suffix and the other without the suffix, e.g. `eon_mobile_awsex` and `eon_mobile_aws` objects. The former is the extension of the latter, and you are recommended to directly use the object with the "ex" suffix.

2.4.1 Appeon Workspace

Manipulates the various features of the Appeon mobile application and the Appeon Workspace, such as the screen orientation, the title bar, the assistive touch bar, the log, the project information, the version, etc.

Note: The Appeon Workspace API takes effect in both the mobile app running in Appeon Workspace and the native standalone mobile applications.

2.4.1.1 `eon_mobile_awsex` object

2.4.1.1.1 Function

`of_clearlog`

Description

Clears the mobile application log information.

Supported on mobile client only.

Syntax

```
aws.of_clearlog ()
```

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_log

Description

Adds the log to the mobile client log, and the default log level is INFO.

Supported on mobile client only.

Syntax

aws.of_log (value integer *ai_level*, value string *as_info*)

aws.of_log (value string *as_info*)

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.
integer	<i>ai_level</i>	1 - FATAL level. 2 - ERROR level. 3 - WARNING level. 4 - INFO level. (Default value) 5 - DEBUG level.
string	<i>as_info</i>	The log information.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

Usage

For the log path of mobile applications (that run in Appeon Workspace), please refer to Section 5.4.6.8, “Client Logs” in *Appeon Server Configuration Guide for .NET* or *Appeon Server Configuration Guide for J2EE*.

of_getassistivetouchmode

Description

Gets the current assistive touch mode. For more about assistive touch mode, refer to Section 2.2, “Event-handling model” in *Mobile UI Design & Development Guide*.

Supported on mobile client only.

Syntax

```
aws.of_getassistivetouchmode ( )
```

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.

Return value

Integer.

- 0 - The assistive touch mode is Left-click.
- 1 - The assistive touch mode is Right-click.
- 2 - The assistive touch mode is Drag.
- 1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_setassistivetouchmode

Description

Sets the current assistive touch mode.

Supported on mobile client only.

Syntax

```
aws.of_setassistivetouchmode ( value integer ai_mode )
```

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.
integer	<i>ai_mode</i>	0 - Sets to the Left-click mode. 1 - Sets to the Right-click mode. 2 - Sets to the Drag mode.

Return value

Integer.

- 1 - Success.
- 1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getassistivetouchbtnvisible

Description

Detects if the assistive touch bar is visible.

Supported on mobile client only.

Syntax

`aws.of_getassistivetouchbtnvisible ()`

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.

Return value

Integer.

1 - The assistive touch bar is visible.

0 - The assistive touch bar is invisible.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_setassistivetouchbtnvisible

Description

Sets whether the assistive touch bar is visible.

Supported on mobile client only.

Syntax

`aws.of_setassistivetouchbtnvisible (value integer ai_mode)`

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.
integer	<i>ai_mode</i>	1 - Sets the assistive touch bar to be visible. 0 - Sets the assistive touch bar to be invisible.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getapporientation

Description

Detects if the screen orientation of the Appeon mobile application is Landscape or Portrait.

Supported on mobile client only.

This function is only valid for the current running application.

Syntax

`aws.of_getapporientation ()`

Argument Type	Argument	Description
<code>eon_mobile_awsex</code>	<code>aws</code>	The name of the <code>eon_mobile_awsex</code> object.

Return value

Integer.

0 - Unknown.

1 - Portrait.

2 - Landscape.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

`of_setapporientation`

Description

Sets the screen orientation of the Appeon mobile application.

The screen orientation is set regardless of current screen rotation.

For apps running on the Android device, this function not only sets the screen orientation, but also locks the screen orientation. Therefore, you no longer need to call [of_setapprotationlock](#) separately to lock the screen orientation.

For apps running on the iOS device, this function only sets the screen orientation.

You will need to call [of_setapprotationlock](#) separately to lock the screen orientation.

And [of_setapprotationlock](#) should be called after the screen orientation is set by `of_setapporientation`, otherwise `of_setapporientation` will have no effect and return -1.

Supported on mobile client only.

Syntax

`aws.of_setapporientation (value integer ai_mode)`

Argument Type	Argument	Description
<code>eon_mobile_awsex</code>	<code>aws</code>	The name of the <code>eon_mobile_awsex</code> object.
integer	<code>ai_mode</code>	1 - Sets screen orientation to Portrait. 2 - Sets screen orientation to Landscape.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getaprotationlock

Description

Detects if the screen rotation of the Appeon mobile application is locked.

Supported on mobile client only.

Not supported on Appeon Workspace Emulator, and this function is only valid for the current running application. For more about the emulator, refer to Chapter 9, *Launching Appeon Workspace Emulator* in *Appeon Developer User Guide*.

Syntax

```
aws.of_getaprotationlock ( )
```

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.

Return value

Integer.

1 - Locked.

0 - Unlocked.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_setaprotationlock

Description

Sets whether to lock the screen orientation of the Appeon mobile application, so the screen orientation will not change with the screen rotation of the device.

For apps running on the Android device, the screen orientation is locked once the screen orientation is set using [of_setapporientation](#), therefore, you do not need to call this function separately to lock the screen orientation. And this function has no effect and returns -1 when executed on the Android device.

For apps running on the iOS device, you will need to call [of_setapporientation](#) to set the screen orientation first and then call this function to lock the screen orientation.

Supported on mobile client only.

Syntax

```
aws.of_setaprotationlock ( value integer ai_mode )
```

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.
integer	<i>ai_mode</i>	1 - Locks the screen orientation of Appeon mobile application, so it will not rotate when the screen

Argument Type	Argument	Description
		orientation of the device changes. 0 - Unlocks the screen orientation of Appeon mobile application.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_gettitlebarvisible

Description

Detects if the Appeon Workspace title bar is visible or invisible.

Supported on mobile client only.

Syntax

`aws.of_gettitlebarvisible()`

Argument Type	Argument	Description
<code>eon_mobile_awsex</code>	<code>aws</code>	The name of the <code>eon_mobile_awsex</code> object.

Return value

Integer.

1 - The title bar is visible.

0 - The title bar is invisible.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_settitlebarvisible

Description

Sets whether the Appeon Workspace title bar is visible or invisible.

Supported on mobile client only.

Syntax

`aws.of_settitlebarvisible (value integer ai_mode)`

Argument Type	Argument	Description
<code>eon_mobile_awsex</code>	<code>aws</code>	The name of the <code>eon_mobile_awsex</code> object.
integer	<code>ai_mode</code>	1 - Sets the title bar to be visible.

Argument Type	Argument	Description
		0 - Sets the title bar to be invisible.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getappinfo

Description

Gets the information of the current Appeon mobile application.

Supported on mobile client only.

Syntax

`aws.of_getappinfo (ref string as_appname, ref string as_appurl)`

Argument Type	Argument	Description
<code>eon_mobile_awsex</code>	<code>aws</code>	The name of the <code>eon_mobile_awsex</code> object.
<code>string</code>	<code>as_appname</code>	Returns the name of the current Appeon mobile application. Returns empty string if it is called in PowerBuilder or Appeon Web or if there is any error.
<code>string</code>	<code>as_appurl</code>	Returns the value of the "App URL" setting of the current Appeon mobile application. Returns empty string if it is called in PowerBuilder or Appeon Web or if there is any error.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getversion

Description

Gets the version number of Appeon Mobile.

Supported on mobile client only.

Syntax

aws.of_getversion (ref string *as_version*)

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.
string	<i>as_version</i>	Returns the Appeon Mobile version number. Returns empty string if it is called in PowerBuilder or Appeon Web, or if there is any error.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_sendmail

Description

Sends an email.

Supported on mobile client only.

Syntax

aws.of_sendmail (value eon_mobile_str_mailcontent *astr_content*, value string *as_recipient*[], value string *as_cc*[], value string *as_bcc*[], value string *as_attachmentfile*[])

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.
eon_mobile_str_mailcontent	<i>astr_content</i>	The structure of the email content. See variable list of eon_mobile_str_mailcontent .
string	<i>as_recipient</i> []	Specifies the recipient list of the mail.
string	<i>as_cc</i> []	Specifies the Cc recipient list of the mail.
string	<i>as_bcc</i> []	Specifies the Bcc recipient list of the mail.
string	<i>as_attachmentfile</i> []	Attachment file path list.

Return value

Integer.

1 - Send the email successfully.

0 - Cancel sending the email.

-1 - Failed to send the email, or It is called in PowerBuilder or Appeon Web, or there is an error.

-2 - The email account is not configured.

-100 - System error.

Usage

The following example shows you how to generate a PDF file and send it via email.

Step 1: Generate the PDF file.

- Generate the PDF file by calling the SaveAs function of DataWindow Control, DataWindowChild object or DataStore object on the mobile device.

Syntax:

```
saveas(filename, PDF!, colheading{, encoding})
```

For example:

```
dw_1.saveas("test.pdf", PDF!, True)
```

Note:

1. filename cannot be empty string or null value.
 2. The file path will be ignored, for the generated file is automatically redirected to the *%cachedir%/plugin/* directory (**cachedir**: the Cache directory that is used by the current mobile application). You can get the cache directory by calling the [of_getcachedir function](#).
- Download the PDF file on the Appeon File service server to the mobile device by calling the [appeonfileservice object](#). These PDF files will be automatically redirected to the *%appcachedir%/plugin/* directory.

Syntax:

```
appeonfileservice lnv_ftp
lnv_ftp = create appeonfileservice

string ls_server
integer li_port
string ls_connectstring

...
ls_connectstring = "username=appeon;password=appeon"

...
ll_re = lnv_ftp.of_appeondownload("test.pdf","") // download pdf file

...
```

Step 2: Send the PDF file.

- Call the `of_sendmail` function of `eon_mobile_awsex` object to send the PDF file as attachment.

Syntax:

```

eon_mobile_awsex lnv_aws

eon_mobile_str_mailcontent lstr_content
string ls_recipient[]
string ls_cc[]
string ls_bcc[]
string ls_attachmentfile[]
integer li_re
...
lnv_aws = create eon_moible_awsex
lstr_content.s_subject = "test mail"
lstr_content.s_notetext = "This is a test mail"
ls_recipient[1] = "support@appeon.com"
ls_attachmentfile[1] = appeongetcachedir()+"/plugin/test.pdf"
...
li_re =
lnv_aws.of_sendmail(lstr_content.ls_recipient.ls_cc.ls_bcc.ls_attachmentfile)
    
```

Note: You should own an account already configured on iPad or iPhone to send the PDF file via email as attachment.

of_getwindowlisticonvisible

Description

Detects if the window list icon is visible or not. For more about the window list icon, refer to the section called “Window title bar, menu, & toolbar” in *Mobile UI Design & Development Guide*.

Supported on mobile client only.

Syntax

```
aws.of_getwindowlisticonvisible()
```

Argument Type	Argument	Description
eon_mobile_awsex	aws	The name of the eon_mobile_awsex object.

Return value

Integer.

- 1 - The window list icon is visible.
- 0 - The window list icon is invisible.
- 1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_setwindowlisticonvisible

Description

Sets whether the window list icon is visible or not.

Supported on mobile client only.

Syntax

`aws.of_setwindowlisticonvisible (value integer ai_mode)`

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.
integer	<i>ai_mode</i>	1 - Sets the window list icon to be visible. 0 - Sets the window list icon to be invisible.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getcloseappiconvisible

Description

Detects if the close app icon is visible or not. For more about the close app icon, refer to the section called “Window title bar, menu, & toolbar” in *Mobile UI Design & Development Guide*.

Supported on mobile client only.

Syntax

`aws.of_getcloseappiconvisible ()`

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.

Return value

Integer.

1 - The close app icon is visible.

0 - The close app icon is invisible.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_setcloseappiconvisible

Description

Sets whether the close app icon is visible or not.

Supported on mobile client only.

Syntax

```
aws.of_setcloseappiconvisible ( value integer ai_mode, value string as_title, value string as_message )
```

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.
integer	<i>ai_mode</i>	1 - Sets the close app icon to be visible. 0 - Sets the close app icon to be invisible.
integer	<i>as_title</i>	The title that displays in the dialog box which pops up when the Close App icon is tapped.
string	<i>as_message</i>	The message that displays in the dialog box which pops up when the Close App icon is tapped. A Yes/No button will be displayed in the dialog box. When Yes is tapped, the app will be closed. When No is tapped, the app will not be closed and will continue running. If <i>as_message</i> is null or blank, then no message box will be displayed and the app will be closed immediately.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getdwmousemovemode

Description

Detects if the pbm_dwnmousemove event ID of DataWindow is supported.

Supported on mobile client only.

Syntax

`aws.of_getdwmousemovemode ()`

Argument Type	Argument	Description
<code>eon_mobile_awsex</code>	<code>aws</code>	The name of the <code>eon_mobile_awsex</code> object.

Return value

Integer.

1 - Supported.

0 - Unsupported.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

`of_setdwmousemovemode`

Description

Sets whether to support the `pbm_dwnmousemove` event ID of DataWindow.

Supported on mobile client only.

Syntax

`aws.of_setdwmousemovemode (value integer ai_mode)`

Argument Type	Argument	Description
<code>eon_mobile_awsex</code>	<code>aws</code>	The name of the <code>eon_mobile_awsex</code> object.
integer	<code>ai_mode</code>	1 - Supported. 0 - Unsupported.

Return value

Integer.

1 - Success

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

2.4.1.1.2 Structure

`eon_mobile_str_mailcontent`

Description

Struct.

The content information of an email.

Property

Type	Variable Name	Description
string	s_subject	Specifies the subject of the mail.
string	s_notetext	Specifies the content of the mail body.
boolean	b_html	Sets if the mail format is HTML.

2.4.2 Barcode

Creates or reads the information of a barcode.

Usage example

- Archives administration:
Facilitate the separation and indexing of documents that have been imaged in batch scanning applications.
- Healthcare industry:
Patient identification (to access patient data, including medical history, drug allergies, etc.)
- Pharmaceutical industry:
Identify, secure and track products to deal with competitors who want to market counterfeit or fake goods.
- Retail industry:
Help track items and also reduce instances of shoplifting involving price tag swapping.
- Tag reader:
Use barcodes to connect customers from your offline marketing materials to information, entertainment, and interactive experiences on their smartphones.

2.4.2.1 eon_mobile_barcodeex object

2.4.2.1.1 Function

of_create

Description

Generates an image containing the barcode according to the data and format specified by the user, and returns the image path.

Supported on mobile client only.

Syntax

```
barcode.of_create ( value string as_data, value integer ai_format, ref string as_filepath )
```

barcode.of_create (value string *as_data*, value integer *ai_format*, ref string *as_filepath*, ref blob *ablb_data*)

Argument Type	Argument	Description
eon_mobile_barcodeex	<i>barcode</i>	The name of the eon_mobile_barcodeex object.
string	<i>as_data</i>	Sets the data of a barcode.
integer	<i>ai_format</i>	Sets the format of barcode.
string	<i>as_filepath</i>	Returns the path of the generated barcode image.
blob	<i>ablb_data</i>	Returns the data of the generated barcode image.

Following is the list of supported barcode formats:

Data	Format	Specification
0	Unknown	
1	EAN-2, GS1 2-digit add-on	It is a supplement to the EAN-13 and UPC-A barcodes. It is often used on magazines and periodicals to indicate an issue number.
2	EAN-5, GS1 5-digit add-on	It is a supplement to the EAN-13 barcode used on books. It is used to give a suggestion for the price of the book.
3	EAN-8	It is the short version of EAN-13. An EAN-8 number contains 7 digits of message plus 1 check digit. The first two or three digits identify the numbering authority, and the remaining 4 or 5 digits identify the product.
4	UPC-E4	It is the short form representation of a UPC-A number. It reduces the data length from 12 digits to 6 digits by compressing the extra zeros. It is suited for identifying products in small packages.
5	ISBN-10, from EAN-13	It is printed in human-readable form above the EAN-13 bar code.
6	UPC-A	It is the most common and well-known symbology in the US, which consists of four areas: 1) The number system; 2) The manufacturer code; 3) The product code; 4) The check digit.
7	EAN-13	It is a superset of UPC-A. Any software or hardware capable of reading an EAN-13 symbol will automatically be able to read an UPC-A symbol.

Data	Format	Specification
8	ISBN-13, from EAN-13	It is printed in human-readable form above the EAN-13 bar code.
9	COMPOSITE, EAN/UPC composite	It is the specification for a 2D barcode symbol included above a DataBar, UPC, EAN or Code 128 barcode to encode additional data. Composite symbols are not supported by all barcode scanners and implementation can be more difficult than other options.
10	Interleaved 2 of 5	It is based on Standard 2 of 5 symbology, but uses both bar and space width to encode information so the density of ITF25 is much higher. It is primarily used in the warehouse industry.
11	CODE 128	A high-density linear symbology that encodes text, numbers, numerous functions and the entire 128 ASCII character set. It enables you to store diversified information in the barcode.
12	CODE 93	An alphanumeric, variable length symbology designed to provide a higher density and data security enhancement to Code 39.
13	CODE 39	The first alpha-numeric symbology that originally encoded 39 characters (now 43) . It is not as compact as the Code 128 symbology, but it is still heavily used in the automotive industry and by the US Department of Defense.
14	ITF	A mix between the POS (Point-of-sale) codes and the logistics barcode Code 128. It's mainly being used in trading, but only on products that need no POS interaction.
20	QR CODE	A "2D Matrix" symbology. Because QR Code requires camera based scanners it is currently restricted for use with applications that will involve imaging scanners within mobile devices and not for POS processing.
21	DataMatrix	A 2D barcode symbology with very high data density. It is spreading in the area of mobile marketing, in such applications the DataMatrix barcode is also known under the name SemaCode.
22	AZTEC	It is a matrix symbology which supports the entire ASCII character set and offers several error checking modes. Aztec symbols are square and may be read at any orientation.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getfiledata

Description

Reads data from the specified image file.

Syntax

barcode.of_getfiledata (string as_filepath)

Argument Type	Argument	Description
eon_mobile_barcodeex	<i>barcode</i>	The name of the eon_mobile_barcodeex object.
string	<i>as_filepath</i>	The full path of the file you want to read.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_read

Description

Executes this function when you need to scan an image containing the barcode information.

The camera will be opened automatically by Appeon, and you need to point the camera at the image to be scanned. The camera will automatically close after a successful scanning, and will return the data and format contained in the barcode.

Supported on mobile client only.

Syntax

barcode.of_read (ref string as_data, ref integer ai_format)

Argument Type	Argument	Description
eon_mobile_barcodeex	<i>barcode</i>	The name of the eon_mobile_barcodeex object.
string	<i>as_data</i>	Returns the data of a barcode.
integer	<i>ai_format</i>	Returns the format of barcode.

Following is the list of supported barcode formats:

Data	Format	Specification
0	Unknown	
1	EAN-2, GS1 2-digit add-on	It is a supplement to the EAN-13 and UPC-A barcodes. It is often used on magazines and periodicals to indicate an issue number.
2	EAN-5, GS1 5-digit add-on	It is a supplement to the EAN-13 barcode used on books. It is used to give a suggestion for the price of the book.
3	EAN-8	It is the short version of EAN-13. An EAN-8 number contains 7 digits of message plus 1 check digit. The first two or three digits identify the numbering authority, and the remaining 4 or 5 digits identify the product.
4	UPC-E4	It is the short form representation of a UPC-A number. It reduces the data length from 12 digits to 6 digits by compressing the extra zeros. It is suited for identifying products in small packages.
5	ISBN-10, from EAN-13	It is printed in human-readable form above the EAN-13 bar code.
6	UPC-A	It is the most common and well-known symbology in the US, which consists of four areas: 1) The number system; 2) The manufacturer code; 3) The product code; 4) The check digit.
7	EAN-13	It is a superset of UPC-A. Any software or hardware capable of reading an EAN-13 symbol will automatically be able to read an UPC-A symbol.
8	ISBN-13, from EAN-13	It is printed in human-readable form above the EAN-13 bar code.
9	COMPOSITE, EAN/UPC composite	It is the specification for a 2D barcode symbol included above a DataBar, UPC, EAN or Code 128 barcode to encode additional data. Composite symbols are not supported by all barcode scanners and implementation can be more difficult than other options.
10	Interleaved 2 of 5	It is based on Standard 2 of 5 symbology, but uses both bar and space width to encode information so the density of ITF25 is much higher. It is primarily used in the warehouse industry.
11	CODE 128	A high-density linear symbology that encodes text, numbers, numerous functions and the entire

Data	Format	Specification
		128 ASCII character set. It enables you to store diversified information in the barcode.
12	CODE 93	An alphanumeric, variable length symbology designed to provide a higher density and data security enhancement to Code 39.
13	CODE 39	The first alpha-numeric symbology that originally encoded 39 characters (now 43) . It is not as compact as the Code 128 symbology, but it is still heavily used in the automotive industry and by the US Department of Defense.
14	ITF	A mix between the POS (Point-of-sale) codes and the logistics barcode Code 128. It's mainly being used in trading, but only on products that need no POS interaction.
20	QR CODE	A "2D Matrix" symbology. Because QR Code requires camera based scanners it is currently restricted for use with applications that will involve imaging scanners within mobile devices and not for POS processing.
21	DataMatrix	A 2D barcode symbology with very high data density. It is spreading in the area of mobile marketing, in such applications the DataMatrix barcode is also known under the name SemaCode.
22	AZTEC	It is a matrix symbology which supports the entire ASCII character set and offers several error checking modes. Aztec symbols are square and may be read at any orientation.

Return value

Integer.

1 - Success.

0 - Cancel.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

2.4.3 Camera

Captures or views a photo or video using the device's camera.

2.4.3.1 eon_mobile_cameraex object

2.4.3.1.1 Function

of_getfiledata

Description

Reads data from the specified image file. If the file is over 4 MB, calling this function may cause memory issue in PB 9 or earlier version. You can upgrade your source code to PB 10 or later, and then call the function of the Appeon Workarounds PBL for the corresponding PB version.

Syntax

```
camera.of_getfiledata ( value string as_filepath )
```

Argument Type	Argument	Description
eon_mobile_cameraex	<i>camera</i>	The name of the eon_mobile_cameraex object.
string	<i>as_filepath</i>	The full path of the file you want to read.

Return value

Blob.

of_openalbums

Description

Opens the album for the user to select a photo or video.

Supported on mobile client only.

Syntax

```
camera.of_openalbums ( ref string as_filepath, ref blob ablb_data )
```

```
camera.of_openalbums ( ref string as_filepath )
```

Argument Type	Argument	Description
eon_mobile_cameraex	<i>camera</i>	The name of the eon_mobile_cameraex object.
string	<i>as_filepath</i>	Returns the full path pointing to this file in the sandbox cache directory. The selected video or photo file will be copied to the cache directory of current application's sandbox. The files in this cache directory will be removed when the application exits.
blob	<i>ablb_data</i>	Returns the file data of the photo file or video file.

Return value

Integer.

1 - Select a photo file successfully, and *as_filepath* will return the full path pointing to the photo file in the sandbox cache directory.

2 - Select a video file successfully, and *as_filepath* will return the full path pointing to the video file in the sandbox cache directory.

0 - Cancel selecting a file from the album.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

Code example 1

```
eon_mobile_camera ln_camera
String ls_filepath
Integer li_filetype
Integer li_return

li_filetype = 2 // Simultaneously displays the photo or video in the album
ln_camera = create eon_mobile_camera
li_return = ln_camera.of_openalbums (ls_filepath)
```

Code example 2

```
eon_mobile_camera ln_camera
String ls_filepath
Integer li_filetype
Integer li_return
Blob lb_data

li_filetype = 2 // Simultaneously displays the photo or video in the album
ln_camera = create eon_mobile_camera
li_return = ln_camera.of_openalbums (ls_filepath, lb_data)
```

of_takefile

Description

Opens the default camera application and uses it to take a photo or record a video.

Once the photo or video is taken, the camera application automatically closes with all the files saved in the album, and returns to the current application.

The supported video file format is MOV, and the supported photo file format is JPG.

Supported on mobile client only.

Syntax

```
camera.of_takefile ( value integer ai_filetype, value boolean ab_allowedit,
ref string as_filepath )
```

```
camera.of_takefile ( value eon_mobile_str_cameraoption astr_option, ref string
as_filepath )
```

```
camera.of_takefile ( value eon_mobile_str_cameraoption astr_option, ref string
as_filepath, ref blob ablb_data )
```

```
camera.of_takefile ( value integer ai_filetype, value boolean ab_allowedit,
ref string as_filepath, ref blob ablb_data )
```

Argument Type	Argument	Description
eon_mobile_cameraex	<i>camera</i>	The name of the eon_mobile_cameraex object.
eon_mobile_str_cameraoption	<i>astr_option</i>	Sets the camera parameters. See variable list of eon_mobile_str_cameraoption .
integer	<i>ai_filetype</i>	Sets the type of file to take. The camera application will be set to this type by default when it is opened. Users can also manually switch the type in the camera application. 1 - Takes a photo. 2 - Records a video.
boolean	<i>ab_allowedit</i>	Sets if the file is editable after picture-taking or video-recording.
string	<i>as_filepath</i>	Returns the full path pointing to the file in the sandbox cache directory. The video and photo files will be saved into device's album, and will be copied to the cache directory of current application's sandbox. The files in this cache directory will be removed when the application exits.
blob	<i>abl_data</i>	Returns the file data of the photo file or video file.

Return value

Integer.

1 - Take a photo successfully, and *as_filepath* will return the full path of the photo file, and return empty string if the picture-taking is cancelled, or if there is any error.

2 - Record a video successfully, and *as_filepath* will return the full path of the video file, and return empty string if the picture-taking is cancelled, or if there is any error.

0 - Cancel taking photo or video.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

Code example 1

```
eon_mobile_str_cameraoption lstr_option
eon_mobile_cameraex ln_camera
String ls_filepath
Integer li_return

lstr_option.i_filetype = 1 // Takes the photo
lstr_option.b_allowedit = true // The photo is editable after picture-taking

ln_camera = create eon_mobile_camera
li_return = ln_camera.of_takefile (lstr_option, ls_filepath)
```

Code example 2

```
eon_mobile_str_cameraoption lstr_option
eon_mobile_cameraex ln_camera
String ls_filepath
Integer li_return
blob lb_data
lstr_option.i_filetype = 1 // Takes the photo
lstr_option.b_allowedit = true // The photo is editable after picture-taking

ln_camera = create eon_mobile_camera
li_return = ln_camera.of_takefile (lstr_option, ls_filepath, lb_data)
```

2.4.3.1.2 Structure

eon_mobile_str_cameraoption

Description

Struct.

Optional parameters to customize the camera settings.

Property

Type	Variable Name	Description
integer	i_filetype	Sets the type of file to take. The camera application will be set to this type by default when it is opened. Users can also manually switch the type in the camera application. 1 - Takes a photo. 2 - Records a video.
boolean	b_allowedit	Sets if the file is editable after picture-taking or video-recording.

2.4.4 Connection

Obtains the network state and connection type.

2.4.4.1 eon_mobile_connectionex object

2.4.4.1.1 Function

of_getconnectioninfo

Description

Gets the device's network connection information.

Supported on mobile client only.

Syntax

connection.of_getconnectioninfo ()

Argument Type	Argument	Description
eon_mobile_connectionex	<i>connection</i>	The name of the eon_mobile_connectionex object.

Return value

Integer.

0 - No connection.

1 - Cell network: 2G/3G/4G.

2 - Wi-Fi network.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

2.4.5 Device

Obtains the device specific information, such as the device type, the device DPI/PPI, the OS version, the device's memory, the device's OS, the screen resolution, etc.

2.4.5.1 eon_mobile_deviceex object

2.4.5.1.1 Function

of_getdeviceid

Description

Gets the device's unique identifier (only supported on iOS 6.0 or later), which depends on the device manufacturer, product type, and operating system.

Supported on mobile client only.

Syntax

device.of_getdeviceid (ref string *as_id*)

Argument Type	Argument	Description
eon_mobile_deviceex	<i>device</i>	The name of the eon_mobile_deviceex object.

Argument Type	Argument	Description
string	<i>as_id</i>	Returns the device unique identifier. Returns empty string if it is called in PowerBuilder or Appeon Web or, if there is any error.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

Usage

Use the `of_getdeviceid()` function to only allow the application to run on specific mobile devices that are authorized. In this way, if unauthorized users install the app on their own mobile device and somehow steals a valid username/password they still won't be able to run the app or your data.

of_getdevicetype

Description

Gets the device type from the manufacturer.

Supported on mobile client only.

Syntax

`device.of_getdevicetype (ref string as_type)`

Argument Type	Argument	Description
<code>eon_mobile_deviceex</code>	<i>device</i>	The name of the <code>eon_mobile_deviceex</code> object.
string	<i>as_type</i>	Returns the device type; e.g., "iPod touch", "iPhone", "iPad", "GT-P5210", "Lenovo A3000-H" etc. The value of device type is set by the manufacturer. Returns empty string if it is called in PowerBuilder or Appeon Web or if there is any error.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getdpi

Description

Gets the device DPI value.

Supported on mobile client only.

Syntax

device.of_getdpi ()

Argument Type	Argument	Description
eon_mobile_deviceex	<i>device</i>	The name of the eon_mobile_deviceex object.

Return value

Integer.

>0 - Return the device DPI value.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getppi

Description

Gets the device PPI value.

Supported on mobile client only.

Syntax

device.of_getppi ()

Argument Type	Argument	Description
eon_mobile_deviceex	<i>device</i>	The name of the eon_mobile_deviceex object.

Return value

Integer.

>0 - Return the device PPI value.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_gettotalmemory

Description

Gets the total number (in KB) of the device memory.

Supported on mobile client only.

Syntax

device.of_gettotalmemory ()

Argument Type	Argument	Description
eon_mobile_deviceex	<i>device</i>	The name of the eon_mobile_deviceex object.

Return value

Long.

>0 - Return the total number of the device memory.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getfreememory

Description

Gets the total number (in KB) of the device's current available free memory.

Supported on mobile client only.

Syntax

device.of_getfreememory ()

Argument Type	Argument	Description
eon_mobile_deviceex	<i>device</i>	The name of the eon_mobile_deviceex object.

Return value

Long.

>0 - Return the total number of the device free memory.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getname

Description

Gets the device's machine name.

Supported on mobile client only.

Syntax

device.of_getname (ref string *as_name*)

Argument Type	Argument	Description
eon_mobile_deviceex	<i>device</i>	The name of the eon_mobile_deviceex object.
string	<i>as_name</i>	Returns the device's machine name. Returns empty string if it is called in PowerBuilder or Appeon Web, or if there is any error.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getorientation

Description

Detects if the device screen orientation is Landscape or Portrait.

Supported on mobile client only.

Syntax

device.of_getorientation()

Argument Type	Argument	Description
eon_mobile_deviceex	<i>device</i>	The name of the eon_mobile_deviceex object.

Return value

Integer.

0 - Unknown.

1 - Portrait, Home button at the bottom.

2 - Portrait, Home button on the top.

3 - Landscape, Home button to the right.

4 - Landscape, Home button to the left.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getplatform

Description

Gets the device's OS name.

Supported on mobile client only.

Syntax

device.of_getplatform (ref string as_platform)

Argument Type	Argument	Description
eon_mobile_deviceex	<i>device</i>	The name of the eon_mobile_deviceex object.
string	<i>as_platform</i>	Returns the device's OS name; e.g., returns "iPhone OS" if the device is iPhone, iPad, or iPod touch, and returns "Android" if the

Argument Type	Argument	Description
		device is Android tablets or smartphones. The value of OS name is obtained from the mobile device. Returns empty string if it is called in PowerBuilder or Appeon Web, or if there is any error.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getresolution

Description

Gets the device's resolution.

Supported on mobile client only.

Syntax

device.of_getresolution (ref integer ai_height, ref integer ai_width)

Argument Type	Argument	Description
eon_mobile_deviceex	<i>device</i>	The name of the eon_mobile_deviceex object.
integer	<i>ai_height</i>	Returns the device's screen height (in pixels).
integer	<i>ai_width</i>	Returns the device's screen width (in pixels).

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getosversion

Description

Gets the OS version.

Supported on mobile client only.

Syntax

device.of_getosversion (ref string as_version)

Argument Type	Argument	Description
eon_mobile_deviceex	<i>device</i>	The name of the eon_mobile_deviceex object.
string	<i>as_version</i>	Returns the device's OS version. For example, returns "6.0" if the iOS version is 6.0; returns "6.0.1" if the iOS version is 6.0.1. The value of OS version is obtained from the mobile device. Returns empty string if it is called in PowerBuilder or Appeon Web, or if there is any error.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getstatusbarvisible

Description

Detects if the system status bar is visible.

Supported on mobile client only.

Syntax

device.of_getstatusbarvisible()

Argument Type	Argument	Description
eon_mobile_deviceex	<i>device</i>	The name of the eon_mobile_deviceex object.

Return value

Integer.

1 - The system status bar is visible.

0 - The system status bar is invisible.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_setstatusbarvisible

Description

Sets whether the iOS system status bar is visible. This API takes no effect for the Android device and returns -1.

Supported on the iOS mobile client only.

Syntax

device.of_setstatusbarvisible (value integer *ai_mode*)

Argument Type	Argument	Description
eon_mobile_deviceex	<i>device</i>	The name of the eon_mobile_deviceex object.
integer	<i>ai_mode</i>	1 - Sets the iOS system status bar to be visible. 0 - Sets the iOS system status bar to be invisible.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

2.4.6 Geolocation

Gets the current position and opens the geolocation functionality.

Usage example

- Express & delivery services:
Track the detailed parcel delivery progress: the time and place of departure, the real-time positioning of the delivery men, passage and arrival, etc.
- Geomarketing:
The online advertising can use this type of marketing, relying on IP addresses to locate the user logged into the Website, and then customizing advertising based on this information.
- Public transportation:
If a user doesn't know the best way to get to somewhere, the user can use the Geolocation API to find out nearby bus stops and metro locations.
- Tourism:
By combining with Map API, travel agencies can provide better vacation tips for the current location and season.

2.4.6.1 eon_mobile_geolocationex object

2.4.6.1.1 Instance variable

powerobject ipo_bindevent

The object to bind with the oe_locationchanged and oe_error.

string is_bindwithlocationchangedevent

The name of the event bound with the oe_locationchanged of the powerobject ipo_bindevent.

string is_bindwitherrorevent

The name of the event bound with the `oe_error` of the powerobject `ipo_bindevent`.

integer ii_errorcode

Error code. When the `oe_error` is triggered, it will be updated and the device will return this message.

```
10 - kCLErrorLocationUnknown, // location is currently unknown, but CL will keep
    trying
11 - kCLErrorDenied, // CL access has been denied (eg, user declined location use)
12 - kCLErrorNetwork, // general, network-related error
13 - kCLErrorHeadingFailure, // heading could not be determined
14 - kCLErrorRegionMonitoringDenied, // Location region monitoring has been denied
    by the user
15 - kCLErrorRegionMonitoringFailure, // A registered region cannot be monitored
16 - kCLErrorRegionMonitoringSetupDelayed, // CL could not immediately initialize
    region monitoring
17 - kCLErrorRegionMonitoringResponseDelayed, // While events for this fence will
    be delivered, delivery will not occur immediately
18 - kCLErrorGeocodeFoundNoResult, // A geocode request yielded no result
19 - kCLErrorGeocodeFoundPartialResult, // A geocode request yielded a partial
    result
20 - kCLErrorGeocodeCanceled, // A geocode request was canceled
21 - kCLErrorDeferredFailed, // Deferred mode failed
22 - kCLErrorDeferredNotUpdatingLocation, // Deferred mode failed because location
    updates disabled or paused
23 - kCLErrorDeferredAccuracyTooLow, // Deferred mode not supported for the
    requested accuracy
24 - kCLErrorDeferredDistanceFiltered, // Deferred mode does not support distance
    filters
25 - kCLErrorDeferredCanceled, // Deferred mode request canceled a previous request
```

string is_errortext

Error message. When the `oe_error` is triggered, it will be updated and the device will return this message.

2.4.6.1.2 Event**oe_error****Description**

It will be triggered if there is an error when using the geolocation functionality.

Supported on mobile client only.

Note: This event will update the value of [is_errortext](#) and [ii_errorcode](#) instance variables.

Syntax

None.

Return value

None.

oe_locationchanged

Description

It will be triggered when the position (longitude and/or latitude) changes after getting the geolocation information.

Supported on mobile client only.

Note: The user can obtain the new position information by calling the [of_getcurrentposition function](#). WPARAM and LPARAM are both empty.

Syntax

None.

Return value

None.

2.4.6.1.3 Function

of_close

Description

Closes the geolocation function.

Supported on mobile client only.

Syntax

`gps.of_close()`

Argument Type	Argument	Description
eon_mobile_geolocationex	<i>gps</i>	The name of the eon_mobile_geolocationex object.

Return value

Integer.

1 - Success.

-1 - If it is called in PowerBuilder or Appeon Web, or there is an error.

of_getcurrentposition

Description

Gets the current position information.

Supported on mobile client only.

Syntax

`gps.of_getcurrentposition (ref eon_mobile_str_coordinates astr_coordinates)`

Argument Type	Argument	Description
eon_mobile_geolocationex	<i>gps</i>	The name of the eon_mobile_geolocationex object.
eon_mobile_str_coordinates	<i>astr_coordinates</i>	The returned detailed position information. See variable list of eon_mobile_str_coordinates .

Return value

Integer.

1 - Success.

-1 - If it is called in PowerBuilder or Appeon Web, or there is an error.

of_isenabled

Description

Detects if the geolocation services can be used.

Supported on mobile client only.

Syntax

gps.of_isenabled()

Argument Type	Argument	Description
eon_mobile_geolocationex	<i>gps</i>	The name of the eon_mobile_geolocationex object.

None.

Return value

Integer

1 - Enabled.

0 - Disabled.

-1 - If it is called in PowerBuilder or Appeon Web, or there is an error.

of_open

Description

Opens the geolocation function.

Supported on mobile client only.

Syntax

gps.of_open (value integer ai_locationaccuracy, value integer ai_distancefilter)

Argument Type	Argument	Description
eon_mobile_geolocationex	<i>gps</i>	The name of the eon_mobile_geolocationex object.
integer	<i>ai_locationaccuracy</i>	Location accuracy update, recommended to use 0 or 1 for the automatic selection. 0 - Use the highest-level of accuracy. 1 - Use the highest possible accuracy and combine it with additional sensor data. This level of accuracy is intended for use in navigation applications that require precise position information at all times and are intended to be used only while the device is plugged in. >1 - User-defined accuracy (in meters).
integer	<i>ai_distancefilter</i>	Location filter, used to control the location update message frequency (in meters). 0 - Notifies by every update. >0 - Updates only when the location change exceeds this value.

Return value

Integer.

1 - Success.

-1 - If it is called in PowerBuilder or Appeon Web, or there is an error.

of_register

Description

Registers the object and the event to be bound with the oe_error and oe_locationchanged events.

Supported on mobile client only.

Syntax

gps.of_register (value powerobject *apb_bind*, value string *as_changedevent*, value string *as_errorevent*)

Argument Type	Argument	Description
eon_mobile_geolocationex	<i>gps</i>	The name of the eon_mobile_geolocationex object.
powerobject	<i>apb_bind</i>	The object to be bound with the oe_error and oe_locationchanged event.
string	<i>as_changedevent</i>	The event to be bound with the oe_locationchanged event .
string	<i>as_errorevent</i>	The event to be bound with the oe_error event .

Return value

Integer.

1 - Success.

-1 - If it is called in PowerBuilder or Appeon Web, or there is an error.

2.4.6.1.4 Structure

eon_mobile_str_coordinates

Description

Struct.

The detailed position information.

Property

Type	Variable Name	Description
decimal{6}	dec_longitude	The longitude value of the current position.
decimal{6}	dec_latitude	The latitude value of the current position.
decimal{2}	dec_altitude	The altitude value of the current position.
decimal{2}	dec_accuracy	The latitude and longitude positioning accuracy.
decimal{2}	dec_altitudeaccuracy	The altitude positioning accuracy.
decimal{2}	dec_heading	The degrees clockwise from true north (0 to 359.99 degrees).

Type	Variable Name	Description
decimal{2}	dec_speed	The displacement velocity (m/sec).
datetime	dt_timestamp	The timestamp to get the current position information.

2.4.6.1.5 Code Example

- To get the GPS information of the photo you take:

```
eon_mobile_cameraex inv_camera
eon_mobile_geolocationex inv_gps

inv_gps = create eon_mobile_geolocationex
inv_camera = create eon_mobile_cameraex

// opens the geolocation function
inv_gps.of_open(3, 0)

// take a photo
li_return = inv_camera.of_takefile(1,false,ls_filepath)

if li_return = 1 then
...
    // get current location info for the photo
    inv_gps.of_getcurrentposition( istr_coordinates)
...
end if
```

- To mark the movement on the map:

1. Adds an eon_mobile_mapex object, uo_map, to the Window.
2. Declares an instance variable.

```
eon_mobile_geolocationex inv_gps
```

3. Registers the GPS service in the Open event of the Window.

```
inv_gps = create eon_mobile_geolocationex
eon_mobile_str_mapoption lstr_mapoption

// Sets the default argument of the map
lstr_mapoption.b_allowmove = true
lstr_mapoption.b_allowzoom = true
lstr_mapoption.b_locatetocurrentlocation = true
lstr_mapoption.i_mapaccuracy = 5
// Opens the map
uo_map.of_open(lstr_mapoption)

// The ue_gps event will be triggered when the positioning accuracy is 3
// meters and the location update range is beyond 100 meters.
inv_gps.of_open(3, 100)

// Bind the "ue_gps" event and "ue_gperror" event of the Window to
// the "oe_locationchanged" event and "oe_error" event of Geolocation object.
inv_gps.of_register( this, "ue_gps","ue_gperror")
```

4. Records the movement and adds it as an annotation to the map.

```
// ue_gps event

eon_mobile_str_coordinates lstr_coordinates
eon_mobile_str_annotation lstr_annotation_appeon

// Get the gps data of the current position
inv_gps.of_getcurrentposition( lstr_coordinates)

// Displays the latitude and longitude of the position in the static text
st_latitude.text = string( lstr_coordinates.dec_latitude)
st_longitude.text = string( lstr_coordinates.dec_longitude)

// Records the gps information of the current position to an annotation
lstr_annotation_appeon.dec_latitude= lstr_coordinates.dec_latitude
lstr_annotation_appeon.dec_longitude = lstr_coordinates.dec_longitude
lstr_annotation_appeon.i_pincolor = 1
lstr_annotation_appeon.s_title= string(lstr_coordinates.dt_timestamp,
"hh:mm:ss" )

// Adds this annotation to the map
uo_map.of_addannotation(lstr_annotation_appeon, true)
```

5. Captures the GPS error.

```
// ue_gpserror event
// Displays the error number and the error message
messagebox("Error "+string(inv_gps.ii_errorcode) , inv_gps.is_errortext)
```

2.4.7 Map

Accesses the information about the longitude, latitude and address as well as processes some annotation operations.

Usage example

- **City planning:**
Use the Street View maps to implement a better city planning and design.
- **Fire department:**
The fire control center can find out the specific position of the man who made the emergency call according to the system, and then quickly plan their arrangement to send the nearest firefighters.
- **Mountaineering:**
For mountain-climbing lovers, they can identify a location according to the known longitude and latitude and vice versa.
- **Tourism:**
By combining with Geolocation API, travel agencies can provide better vacation tips for the current location and season.
- **Turn-by-turn route navigation:**

Help the user navigate along a route by providing detailed turn-by-turn directions. Map API does this by registering with the Geolocation API to receive repeated location updates of the user's position.

2.4.7.1 eon_mobile_mapex object

2.4.7.1.1 Instance variable

integer ii_errorcode

Error code. It will be updated when the oe_error is triggered, and this message will be returned from the device. For example:

```
1 - MKErrorUnknown
2 - MKErrorServerFailure, // The map service is turn off
3 - MKErrorLoadingThrottled,
4 - MKErrorPlacemarkNotFound, // address is not found
-1 - NSErrorUnknown
-999 - NSErrorCancelled,
-1001 - NSErrorTimedOut,
-1002 - NSErrorCannotFindHost,
-1004 - NSErrorCannotConnectToHost,
-1005 - NSErrorNetworkConnectionLost,
-1009 - NSErrorNotConnectedToInternet
```

string is_errortext

Error message. It will be updated when the oe_error is triggered, and this message will be returned from the device.

eon_mobile_str_annotation istr_clickedannotation

The annotation (added by [of addannotation](#)) that the user clicks on.

2.4.7.1.2 Event

oe_annotationclicked

Description

The oe_annotationclicked event will be automatically triggered when an annotation (added by [of addannotation](#)) is clicked.

Supported on mobile client only.

Note: This event cannot be triggered via the code, and can only be triggered by clicking the annotation on the map.

Syntax

None.

Return value

None.

The information of the clicked annotation is stored to the istr_clickedannotation instance variable.

oe_loadstart**Description**

The oe_loadstart event will be automatically triggered when starting loading the map. LPARAM and WPARAM are empty.

Supported on mobile client only.

Note: This event can only be automatically triggered after calling the [of_open function](#).

Syntax

None.

Return value

None.

oe_loadend**Description**

The successful completion of loading the map. LPARAM and WPARAM are empty.

Supported on mobile client only.

Note: This event can only be automatically triggered after calling the [of_open function](#).

Syntax

None.

Return value

None.

oe_error**Description**

Failure in loading the map. WPARAM is the error number.

Supported on mobile client only.

Note: This event can only be automatically triggered after calling the [of_open function](#).

Syntax

None.

Return value

None.

oe_changed**Description**

The map display area has changed.

Note: This event can only be automatically triggered after calling the [of_open function](#).

Syntax

None.

Return value

None.

2.4.7.1.3 Function

of_addannotation

Description

Adds a map annotation.

Supported on mobile client only.

Syntax

map.of_addannotation (ref eon_mobile_str_annotation *astr_annotation*, value boolean *ab_move*)

Argument Type	Argument	Description
eon_mobile_mapex	<i>map</i>	The name of the eon_mobile_mapex object.
eon_mobile_str_annotation	<i>astr_annotation</i>	The annotation to be added. (For the relevant information, refer to the property of the eon_mobile_str_annotation structure .) When the annotation is added successfully, this function will generate and store the unique identifier of the annotation to the <i>l_id</i> property in the structure. And the <i>l_id</i> property will be used when this annotation is removed or selected.
boolean	<i>ab_move</i>	True: To automatically move the map so as to make the new annotation visible. False: Not to move the map.

Return value

Integer.

1 - Success.

-1 - Failure or it is called in PowerBuilder or Appeon Web, or there is an error.

Code example

```

eon_mobile_str_annotation istr_annotation_appeon
istr_annotation_appeon.dec_latitude= 22.54993
istr_annotation_appeon.dec_longitude = 113.94977
istr_annotation_appeon.i_pincolor = 1
istr_annotation_appeon.s_pinimage="apb.png"
istr_annotation_appeon.s_subtitle="Appeon Mobile"
istr_annotation_appeon.s_title="Appeon"
uo_map.of_addannotation(istr_annotation_appeon, true)
    
```

of_adresstocoordinate

Description

Obtains the longitude and latitude according to the address.

Supported on mobile client only.

Syntax

map.of_adresstocoordinate (value string *as_address*, ref decimal *dec_longitude*, ref decimal *adec_latitude*)

Argument Type	Argument	Description
eon_mobile_mapex	<i>map</i>	The name of the eon_mobile_mapex object.
string	<i>as_address</i>	Address, a human-readable string to specify locations.
decimal	<i>dec_longitude</i>	The returned longitude, which is defined using numerals within a comma-separated text string that have a precision to 6 decimal places.
decimal	<i>adec_latitude</i>	The returned latitude, which is defined using numerals within a comma-separated text string that have a precision to 6 decimal places.

Return value

Integer

1 - Success.

-1 - Failure or it is called in PowerBuilder or Appeon Web, or there is an error.

of_close

Description

Closes the default map application.

Supported on mobile client only.

Syntax

map.of_close ()

Argument Type	Argument	Description
eon_mobile_mapex	<i>map</i>	The name of the eon_mobile_mapex object.

Return value

Integer.

1 - Success.

-1 - Failure or it is called in PowerBuilder or Appeon Web, or there is an error.

of_coordinatetoaddress

Description

Converts the longitude and latitude into the corresponding address.

Supported on mobile client only.

Syntax

map.of_coordinatetoaddress (value decimal *dec_longitude*, value decimal *adec_latitude*, ref string *as_address*)

Argument Type	Argument	Description
eon_mobile_mapex	<i>map</i>	The name of the eon_mobile_mapex object.
decimal	<i>dec_longitude</i>	Longitude, which is defined using numerals within a comma-separated text string that have a precision to 6 decimal places.
decimal	<i>adec_latitude</i>	Latitude, which is defined using numerals within a comma-separated text string that have a precision to 6 decimal places.
string	<i>as_address</i>	The returned address, which is converted from longitude and latitude values.

Return value

Integer.

1 - Success.

-1 - Failure or it is called in PowerBuilder or Appeon Web, or there is an error.

of_open

Description

Opens the default map application at the specified position.

Supported on mobile client only.

Syntax

*map.of_open (value eon_mobile_str_mapoption *astr_mapoption*)*

*map.of_open (value eon_mobile_str_mapoption *astr_mapoption*, ref eon_mobile_str_annotation *astr_clickedannotation*)*

Argument Type	Argument	Description
eon_mobile_mapex	<i>map</i>	The name of the eon_mobile_mapex object.
eon_mobile_str_mapoption	<i>astr_mapoption</i>	Relevant information about the map. (For the relevant information, refer to the property of the eon_mobile_str_mapoption structure)
eon_mobile_str_annotation	<i>astr_clickedannotation</i>	The annotation that the user clicks on. (For the relevant information, refer to the property of the eon_mobile_str_annotation structure .)

Return value

Integer.

1 - Success.

-1 - Failure or it is called in PowerBuilder or Appeon Web, or there is an error.

Code example

```

eon_mobile_str_mapoption lstr_mapoption
lstr_mapoption.b_allowmove = true
lstr_mapoption.b_allowzoom = true
lstr_mapoption.b_locatetocurrentlocation = true
lstr_mapoption.i_mapaccuracy = 5
uo_map.of_open(lstr_mapoption)
    
```

of_refresh

Description

Refreshes the map according to the specified information.

Supported on mobile client only.

Syntax

map.of_refresh (value eon_mobile_str_mapoption astr_mapoption)

Argument Type	Argument	Description
eon_mobile_mapex	<i>map</i>	The name of the eon_mobile_mapex object.
eon_mobile_str_mapoption	<i>astr_mapoption</i>	Relevant information about the map. (For the relevant information, refer to the property of the eon_mobile_str_mapoption structure)

Return value

Integer.

1 - Success.

-1 - Failure or it is called in PowerBuilder or Appeon Web, or there is an error.

of_removeannotation

Description

Removes a map annotation that is added by [of_addannotation](#).

Supported on mobile client only.

Syntax

map.of_removeannotation (value eon_mobile_str_annotation astr_annotation)

map.of_removeannotation (value long al_id)

Argument Type	Argument	Description
eon_mobile_mapex	<i>map</i>	The name of the eon_mobile_mapex object.
eon_mobile_str_annotation	<i>astr_annotation</i>	The annotation to be removed. (For the relevant information, refer to the property of the eon_mobile_str_annotation structure .) This function will remove the annotation according to the unique identifier of the annotation stored in the l_id property of the structure.
long	<i>al_id</i>	The unique identifier of the annotation to be removed.

Return value

Integer.

1 - Success.

-1 - Failure or it is called in PowerBuilder or Appeon Web, or there is an error.

of_removeallannotation

Description

Removes all map annotations that are added by [of_addannotation](#).

Supported on mobile client only.

Syntax

`map.of_removeallannotation()`

Argument Type	Argument	Description
eon_mobile_mapex	<i>map</i>	The name of the eon_mobile_mapex object.

Return value

Integer.

1 - Success.

-1 - Failure or it is called in PowerBuilder or Appeon Web, or there is an error.

of_selectannotation

Description

Selects a map annotation that is added by [of_addannotation](#).

Supported on mobile client only.

Syntax

`map.of_selectannotation (value eon_mobile_str_annotation astr_annotation)`

`map.of_selectannotation (value long al_id)`

Argument Type	Argument	Description
eon_mobile_mapex	<i>map</i>	The name of the eon_mobile_mapex object.
eon_mobile_str_annotation	<i>astr_annotation</i>	The annotation to be selected. (For the relevant information, refer to the property of the eon_mobile_str_annotation structure .) This function will select the annotation according to

Argument Type	Argument	Description
		the unique identifier of the annotation stored in the l_id property of the structure.
long	<i>al_id</i>	The unique identifier of the annotation to be selected.

Return value

Integer.

1 - Success.

-1 - Failure or it is called in PowerBuilder or Appeon Web, or there is an error.

2.4.7.1.4 Structure

eon_mobile_str_mapoption

Description

Struct.

A map annotation. The user can click on the text markup.

Property

Type	Variable Name	Description
boolean	b_locatetocurrent location	Moves to the current location.
decimal{6}	dec_longitude	The longitude of the map central location (only takes effect when the value of bLocateToCurrentLocation is False).
decimal{6}	dec_latitude	The latitude of the map central location.
boolean	b_allowmove	Allows the user to move or not (the default value is True).
boolean	b_allowzoom	Allows the user to zoom or not (the default value is True).
integer	i_mapaccuracy	The map display accuracy or amplification factor (unit: meter).
integer	i_maptype	The map type (0: Default type, 1: Satellite view, 3: Hybrid map).

eon_mobile_str_annotation

Description

Struct.

Relevant information about the initialized map.

Property

Type	Variable Name	Description
decimal{6}	dec_longitude	The longitude of the added or obtained annotation.
decimal{6}	dec_latitude	The latitude of the added or obtained annotation.
string	s_title	The title of the added or obtained annotation.
string	s_subtitle	The subtitle of the added or obtained annotation.
integer	i_pincolor	The pin color. Only takes effect when there is no user-defined pin icon (the supported colors are: Red = 0, Green = 1, Purple = 2).
string	s_pinimage	The pin image (If it is NULL, use the default icon).
long	l_id	The unique identifier of the annotation.

2.4.8 Media

Plays a media file using the device's media player.

2.4.8.1 eon_mobile_mediaex object

2.4.8.1.1 Function

of_play

Description

Uses the default media player application to play the media files.

The supported audio file formats are .aac, .mp3, .aiff, and .wav; the supported video file formats are .m4v, .mp4, and .mov.

Not all the media files with these suffixes can be played; if some media files cannot be played, they can be transcoded via iTunes.

Supported on mobile client only.

Syntax

media.of_play (value string *as_filepath*)

Argument Type	Argument	Description
eon_mobile_mediaex	<i>media</i>	The name of the eon_mobile_mediaex object.
string	<i>as_filepath</i>	Sets the full path of a media file to be played.

Return value

Integer.

1 - Play the media file successfully.

0 - Cancel playing the media file.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

2.4.9 Notification

Sends a notification to the notification bar.

Usage example

- Airlines:

Because of the weather effects and other force majeure, the airlines can send such warm notifications to inform passengers of the latest flight dynamics, so as to reassure them and reduce some troublesome cases .

- Banking and finance:

The e-notification service is a very practical solution that allows quick and safe transmission of any personalized message via electronic mail or SMS. It is especially helpful as a form of automatic notification of, for example, change in account balance, issuance of an invoice, deadline for installment payment, etc. The client is provided with the latest information and always stays updated.

- Healthcare:

Reminding patients of their dentist appointments is an important part of any dentistry which helps build and strengthen the relationship between patient and dentist. When patients receive those reminders, they are not only reminded of their appointment they are also made aware of the quality and care of the business.

2.4.9.1 eon_mobile_notificationex object

2.4.9.1.1 Function

of_addmessage

Description

Sends a notification to the notification bar.

Supported on mobile client only.

Syntax

notification.of_addmessage (value string *as_message*)

Argument Type	Argument	Description
eon_mobile_notificationex	<i>notification</i>	The name of the eon_mobile_notificationex object.
string	<i>as_message</i>	The message to be displayed on the notification bar.

Return value

Long.

>0 - Return the current notification handle if sending notification successfully.

[of_removemessage function](#) can use this handle to delete the notification.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_removemessage

Description

Removes a notification from the notification bar.

Supported on mobile client only.

Syntax

notification.of_removemessage (value long *al_handle*)

Argument Type	Argument	Description
eon_mobile_notificationex	<i>notification</i>	The name of the eon_mobile_notificationex object.
long	<i>al_handle</i>	Specifies the handle of the notification, which is created by the of_addmessage() function.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

2.4.10 Offline

Checks and applies the updates of the current application.

2.4.10.1 eon_offlineex object

2.4.10.1.1 Function

of_applydbupdate

Description

Updates the local database files of the current application.

Supported on mobile client only.

Syntax

offline.of_applydbupdate ()

Argument Type	Argument	Description
eon_offlineex	<i>offline</i>	The name of the eon_offlineex object.

Return value

Integer.

0 - The DB structure of the current application is the same as that in the server, and no need to update.

1 - Success.

-1 - Failure or it is called in PowerBuilder.

-2 - It is called in Appeon Web and this operation is not supported.

of_checkappeonserver

Description

Detects if Appeon Server exists.

Supported on mobile client only.

Syntax

offline.of_checkappeonserver ()

Argument Type	Argument	Description
eon_offlineex	<i>offline</i>	The name of the eon_offlineex object.

Return value

Integer.

1 - Appeon Server exists.

-1 - Appeon Server does not exist or there is an error in the checking of the Appeon Server, or it is called in PowerBuilder.

of_applyupdate

[Syntax 1](#)

[Syntax 2](#)

Description 1

Updates the current application from the server.

Supported on mobile client only.

Syntax 1

offline.of_applyupdate (boolean *ab_showdialog*)

Argument Type	Argument	Description
eon_offlineex	<i>offline</i>	The name of the eon_offlineex object.
boolean	<i>ab_showdialog</i>	This argument controls whether to display a dialog box that prompts the end user to restart the app after updated. The prompt message is "After the app is updated, you need to restart the app to make the new version take effect. Please confirm your update." The end user can choose OK to update the app, or choose Cancel not to update the app.
		True - Displays the dialog box.
		False - Do not display the dialog box.

Return value 1

Integer.

1 - Success.

0 - Latest app and no need to update.

-1 - Failure or it is called in PowerBuilder.

-2 - It is called in Appeon Web and this operation is not supported.

Description 2

Updates the current application from the server according to the return value of [of_checkupdate](#).

Supported on mobile client only.

Syntax 2

offline.of_applyupdate ()

Argument Type	Argument	Description
eon_offlineex	<i>offline</i>	The name of the eon_offlineex object.

Return value 2

Integer.

1 - Success.

0 - Latest app and no need to update.

-1 - Failure or it is called in PowerBuilder.

-2 - It is called in Appeon Web and this operation is not supported.

Note: The following application configuration information will be synchronized when running the offline mobile application for the first time or when calling the `of_applyupdate` function.

- AEM configuration
 1. Application > Local Database
 2. Application > PB Features > INI Files
 3. Application > PB Features > Decimal Precision
 4. Application > Client Features > [codeexamples] > Application Title
 5. Application > Client Features > [codeexamples] > Retina Display (Mobile Only)
 6. Application > Data Transfer > Encoding
 7. Mobile UI Resizing
- Application configuration
- Data source
- Embedded SQL
- DW SQL
- Local database files

of_checkupdate

Description

Detects if there is any update in the current application.

Supported on mobile client only.

Syntax

`offline.of_checkupdate ()`

Argument Type	Argument	Description
eon_offlineex	<i>offline</i>	The name of the eon_offlineex object.

Return value

Integer.

0 - There is no update.

- 1 - There is an update only on the application.
- 2 - There is an update only on the DB file.
- 3 - There are updates both on the DB file and application.
- 1 - Failure in the update check or it is called in PowerBuilder.
- 2 - It is called in Appeon Web and this operation is not supported.

2.4.11 Textchecker

Checks spelling in a text field.

2.4.11.1 eon_mobile_textcheckex object

2.4.11.1.1 Function

of_completionsforpartialword

Description

Returns a list of words that are possible valid replacements for a misspelled word.
Supported on mobile client only.

Syntax

```
textchecker.of_completionsforpartialword ( value
eon_mobile_str_textcheckoption astr_textcheckoption, ref string
as_completions[ ] )
```

Argument Type	Argument	Description
eon_mobile_textcheckerex	<i>textchecker</i>	The name of the eon_mobile_textcheckerex object.
eon_mobile_str_textcheckoption	<i>astr_textcheckoption</i>	Check option. See variable list of eon_mobile_str_textcheckoption .
string	<i>as_completions[]</i>	Returns an array of strings, each of which is a completion of a partially entered word represented by range in string. If no possible completions are found, the method returns an empty array.

Return value

Integer.

- 1 - Found possible completions.
- 0 - Found no possible completions.
- 1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getignoredwords

Description

Returns the words that the receiver ignores when spell-checking.

The spell checker excludes ignored words as misspelled words during the current spell-checking session only.

Supported on mobile client only.

Syntax

textchecker.of_getignoredwords (ref string *as_words*[])

Argument Type	Argument	Description
eon_mobile_textcheckerex	<i>textchecker</i>	The name of the eon_mobile_textcheckerex object.
string	<i>as_words</i> []	Returns an array of strings, each of which specifies a word the receiver ignores.

Return value

Integer.

1 - Found any ignored words.

0 - Found no ignored words.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_setignoredwords

Description

Sets the list of words that the receiver should ignore, and the ignored words are not treated as misspelled words.

Supported on mobile client only.

Syntax

textchecker.of_setignoredwords (value string *as_words*[])

Argument Type	Argument	Description
eon_mobile_textcheckerex	<i>textchecker</i>	The name of the eon_mobile_textcheckerex object.
string	<i>as_words</i> []	Sets an array of strings, each of which specifies a word the receiver should ignore.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getlanguages

Description

Gets the languages for which the text checker can perform spell-checking.

Supported on mobile client only.

Syntax

textchecker.of_getlanguages (ref string *as_languages[]*)

Argument Type	Argument	Description
eon_mobile_textcheckerex	<i>textchecker</i>	The name of the eon_mobile_textcheckerex object.
string	<i>as_languages[]</i>	Returns an array of strings representing ISO 639-1 language codes or combined ISO 639-1 language codes and ISO 3166-1 regional codes (for example, en_US).

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getmisspelledword

Description

Gets the range of the first misspelled word encountered.

Supported on mobile client only.

Syntax

textchecker.of_getmisspelledword (value eon_mobile_str_textcheckoption *astr_textcheckoption*, ref long *al_start*, ref long *al_length*)

Argument Type	Argument	Description
eon_mobile_textcheckerex	<i>textchecker</i>	The name of the eon_mobile_textcheckerex object.
eon_mobile_str_textcheckoption	<i>astr_textcheckoption</i>	Check option. See variable list of eon_mobile_str_textcheckoption .
long	<i>al_start</i>	Returns a long whose value is the starting position of the first misspelled word.
long	<i>al_length</i>	Returns a long whose value is the length of the first misspelled word.

Return value

Integer.

1 - Found the misspelled word.

0 - Found no misspelled words.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_guessesforward

Description

Returns an array of strings that are possible completions for a partially entered word.

Supported on mobile client only.

Syntax

textchecker.of_guessesforward (value eon_mobile_str_textcheckoption *astr_textcheckoption*, ref string *as_guesses[]*)

Argument Type	Argument	Description
eon_mobile_textcheckerex	<i>textchecker</i>	The name of the eon_mobile_textcheckerex object.
eon_mobile_str_textcheckoption	<i>astr_textcheckoption</i>	Check option. See variable list of eon_mobile_str_textcheckoption .
string	<i>as_guesses[]</i>	Returns an array of strings each of which might be a correct substitute (that is, a guess) for a misspelled word in the given range of the string. If no possible guesses are found, the method returns an empty array.

Return value

Integer.

1 - Found possible guesses.

0 - Found no possible guesses.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_haslearnedword

Description

Detects if the text checker has learned the specified word.

Supported on mobile client only.

Syntax

textchecker.of_haslearnedword (value string *as_word*)

Argument Type	Argument	Description
eon_mobile_textcheckerex	<i>textchecker</i>	The name of the eon_mobile_textcheckerex object.
string	<i>as_word</i>	A string representing a word.

Return value

Integer.

1 - The text check has learned the word.

0 - The text check hasn't learned the word.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_learnword

Description

Tells the text checker to learn the specified word so that it is not evaluated as misspelled.

Supported on mobile client only.

Syntax

textchecker.of_learnword (value string *as_word*)

Argument Type	Argument	Description
eon_mobile_textcheckerex	<i>textchecker</i>	The name of the eon_mobile_textcheckerex object.
string	<i>as_word</i>	A string representing the word for the text checker to learn.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_unlearnword

Description

Tells the text checker to unlearn the specified word.

Supported on mobile client only.

Syntax

textchecker.of_unlearnword (value string *as_word*)

Argument Type	Argument	Description
eon_mobile_textcheckerex	<i>textchecker</i>	The name of the eon_mobile_textcheckerex object.

Argument Type	Argument	Description
string	<i>as_word</i>	A string representing the word for the text checker to unlearn.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

2.4.11.1.2 Structure

eon_mobile_str_textcheckoption

Description

Struct.

Checks spelling.

Property

Type	Variable Name	Description
string	<i>s_source</i>	The string which you want to check.
long	<i>l_start</i>	A long indicating where the check will begin in <i>s_source</i> .
long	<i>l_length</i>	Starting from the position specified by <i>l_start</i> , the number of characters needed to be checked.
string	<i>s_language</i>	The language of the words to be checked for correct spelling.

2.4.12 NFC

Provides access to Near Field Communication (NFC) functionality, allowing the applications to read NDEF message from the Android device.

Note: iOS devices have no NFC features at all, therefore, the NFC APIs are for the Android-powered device only, however, not every Android-powered device provides NFC functionality. For more information, see the [Near Field Communication](#) guide.

There are two methods to read/write data from the card. Pay attention to the functions listed below, because they are effective for one method only. The functions not listed below are effective for both methods, such as [of_open](#), [of_close](#), [of_getblocksize](#), [of_getmaxblockcount](#), [of_getmaxsectorcount](#), [of_getsize](#), [of_gettagid](#).

- Method 1: Read/write data in NDEF (NFC Data Exchange Format)

The following functions are effective for this method only:

[of_getndefrecordcount](#)

- [of_getndefrecord](#)
- [of_addrrecord](#)
- [of_writerecord](#)

- Method 2: Read/write data in bytes from blocks

The following functions are effective for this method only:

- [of_settimeout](#)
- [of_gettimeout](#)
- [of_setkey](#)
- [of_startblockoperate](#)
- [of_stopblockoperate](#)
- [of_readblock](#)
- [of_writeblock](#)

See code example for method 1 and 2 in [Code example](#).

2.4.12.1 eon_mobile_nfcex object

2.4.12.1.1 Event

oe_newtagfound

Description

Occurs when the card is detected by the reader.

Syntax

nfc.oe_newtagfound (integer *ai_type*)

Argument	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.
integer	<i>ai_type</i>	0 - MifareClassic. Requires the encryption key, or uses the default key. 1 - MifareUltralight. Requires no encryption key. 2 - NfcV. Requires no encryption key. 3 - NDEF Only. Supports read/write NDEF records but does not support read/write data (in bytes) from the block.

Argument	Argument	Description
		4 - UNKNOWN. Unrecognizable card types. Cannot read/write from the card, but can get card identifier.

Return value

None

2.4.12.1.2 Function

of_addrecord

Description

Adds the NDEF record.

Supported on mobile client only.

Syntax

nfc.of_addrecord (eon_mobile_str_nfcrecord astr_nfcrecord)

Argument Type	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.
eon_mobile_str_nfcrecord	<i>astr_nfcrecord</i>	The record to be added. See variable list of eon_mobile_str_nfcrecord .

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_close

Description

Closes the NFC communication.

Supported on mobile client only.

Syntax

nfc.of_close ()

Argument Type	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getblocksize

Description

Gets the size (in bytes) of the block.

Supported on mobile client only.

Syntax

nfc.of_getblocksize ()

Argument Type	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.

Return value

Integer.

>0 - Returns the size (in bytes) of the block.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getmaxblockcount

Description

Gets the total number of blocks.

Supported on mobile client only.

Syntax

nfc.of_getmaxblockcount ()

Argument Type	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.

Return value

Integer.

>0 - Returns the total number of blocks.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getmaxsectorcount

Description

Gets the total number of sectors.

Supported on mobile client only.

Syntax

nfc.of_getmaxsectorcount ()

Argument Type	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.

Return value

Integer.

>0 - Returns the total number of sectors.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getndefrecord

Description

Gets the NDEF record.

Supported on mobile client only.

Syntax

nfc.of_getndefrecord (integer *ai_index*, ref eon_mobile_str_nfcrecord *astr_nfcrecord*)

Argument Type	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.
integer	<i>ai_index</i>	Block indexes which is obtained through of_getMaxBlockCount . Block indexes are sequential numbers starting from 1.
eon_mobile_str_nfcrecord	<i>astr_nfcrecord</i>	The record to be read. See variable list of eon_mobile_str_nfcrecord .

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getndefrecordcount

Description

Gets the total number of the NDEF records.

Supported on mobile client only.

Syntax

nfc.of_getndefrecordcount ()

Argument Type	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.

Return value

Integer.

>0 - Returns the total number of the NDEF records.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_getsize

Description

Gets the total capacity of the card.

Supported on mobile client only.

Syntax

nfc.of_getsize ()

Argument Type	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.

Return value

Integer.

>0 - Returns the total capacity of the card.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_gettagid

Description

Gets the unique identifier of the card.

Supported on mobile client only.

Syntax

nfc.of_gettagid (ref string *as_tagid*)

Argument Type	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.
string	<i>as_tagid</i>	Returns the unique identifier of the card; or returns

Argument Type	Argument	Description
		empty string if it is called in PowerBuilder or Appeon Web or if there is any error.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_gettimeout

Description

Gets the timeout value for reading/writing data.

Supported on mobile client only.

Syntax

nfc.of_gettimeout ()

Argument Type	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.

Return value

Integer.

>0 - Returns the timeout value for reading/writing data.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_open

Description

Starts the NFC communication.

Supported on mobile client only.

Syntax

nfc.of_open ()

Argument Type	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_readblock

Description

Reads data (in bytes) from the block.

Supported on mobile client only.

Syntax

nfc.of_readblock (integer *ai_block*, ref blob *abl_data*)

Argument Type	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.
integer	<i>ai_block</i>	Block indexes which is obtained through of_getMaxBlockCount . Block indexes are sequential numbers starting from 1.
blob	<i>abl_data</i>	Data that are read from the block.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_register

Description

Registers the object and the event to be bound with the oe_newtagfound events.

Supported on mobile client only.

Syntax

nfc.of_register (powerobject *apb_bind*, string *as_changedevent*)

Argument Type	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.
powerobject	<i>apb_bind</i>	The object to be bound with the oe_newtagfound event.
string	<i>as_changedevent</i>	The event to be bound with the oe_newtagfound event.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_setkey

Description

Sets the read/write encryption key.

Supported on mobile client only.

Syntax

nfc.of_setkey (string as_key)

Argument Type	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.
string	<i>as_key</i>	Encryption key for reading/writing data. Or use the system default key. MIFARE Classic cards require the key. Card types can be obtained through oe_newtagfound event.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_settimeout

Description

Sets the timeout value for reading/writing data.

Supported on mobile client only.

Syntax

nfc.of_settimeout (integer ai_timeout)

Argument Type	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.
integer	<i>ai_timeout</i>	Timeout value for reading/writing data. Normally, this parameter needs not to be set. Only if the read/write operation would take a long time, then this parameter should be set, otherwise the

Argument Type	Argument	Description
		operation will time out and disconnect automatically.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_startblockoperate

Description

Starts reading/writing data from the block.

Supported on mobile client only.

Syntax

`nfc.of_startblockoperate ()`

Argument Type	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_stopblockoperate

Description

Stops reading/writing data from the block.

Supported on mobile client only.

Syntax

`nfc.of_stopblockoperate ()`

Argument Type	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_triggerevent

Description

Triggers the object events bound with the `oe_newtagfound` event.

Supported on mobile client only.

Syntax

`nfc.of_triggerevent (string as_event)`

Argument Type	Argument	Description
<code>eon_mobile_nfcex</code>	<code>nfc</code>	The name of the <code>eon_mobile_nfcex</code> object.
<code>string</code>	<code>as_event</code>	The event name bound with oe_newtagfound event.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_writeblock

Description

Writes data (in bytes) to the block.

Supported on mobile client only.

Syntax

`nfc.of_writeblock (integer ai_block, blob abl_data)`

Argument Type	Argument	Description
<code>eon_mobile_nfcex</code>	<code>nfc</code>	The name of the <code>eon_mobile_nfcex</code> object.
<code>integer</code>	<code>ai_block</code>	Block indexes which is obtained through of_getMaxBlockCount . Block indexes are sequential numbers starting from 1.
<code>blob</code>	<code>abl_data</code>	Data to be written to the block. Block size should be obtained through of_getBlockSize first. Block size can be 4 bytes or 16 bytes. Data exceeding the block size will be automatically truncated.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

of_writerecord

Description

Writes the NDEF record. [of_addrecord](#) should be called first to add one or more records, then of_writerecord should be called to write all records at one time.

Supported on mobile client only.

Syntax

`nfc.of_writerecord ()`

Argument Type	Argument	Description
eon_mobile_nfcex	<i>nfc</i>	The name of the eon_mobile_nfcex object.

Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or Appeon Web, or there is an error.

2.4.12.1.3 Structure

eon_mobile_str_nfcrecord

Description

Structure.

The record to be added.

Property

Type	Variable Name	Description
int	<code>i_messagetype</code>	0 - URL message 1 - Mime message 2 - App message 3 - Text message 4 - Poster message 5 - Alternative carrier message 6 - Handover carrier message

Type	Variable Name	Description
		7 - Handover request 8 - Handover select 9 - External message
string	s_Domain	This parameter is required when i_messagetype is External message.
string	s_Type	Data type or MIME type. This parameter is required when i_messagetype is Mime message or External message.
string	s_Payload	Valid data.

2.4.12.1.4 Code example

Code example for method 1: Read/write data in NDEF (NFC Data Exchange Format)

```
eon_mobile_nfcex  Inv_mobile_nfc  //create nfc object
eon_mobile_str_nfcrecord  istr_nfcrecord

//open nfc object
Inv_mobile_nfc.of_open()

//read nfc card record, li_index is the index of the block which you want to get.
Inv_mobile_nfc.of_getNdefRecord(li_index,istr_nfcrecord)

//write nfc card record
Inv_mobile_nfc.of_addrecord(istr_nfcrecord)
Inv_mobile_nfc.of_writeRecords()

//close nfc object
Inv_mobile_nfc.of_close()
```

Code example for method 2: Read/write data in bytes from blocks

```
eon_mobile_nfcex  Inv_mobile_nfc  //create nfc object
eon_mobile_str_nfcrecord  istr_nfcrecord

//open nfc object
Inv_mobile_nfc.of_open()

//if necessary
Inv_mobile_nfc.of_setTimeout(li_timeout)
Inv_mobile_nfc.of_setkey(ls_key)

Inv_mobile_nfc.of_startBlockOperate()

//read nfc card record
Inv_mobile_nfc.of_readBlock(li_block,ibl_data)

//write nfc card record
Inv_mobile_nfc.of_writeBlock(li_block,ibl_data)

Inv_mobile_nfc.of_stopBlockOperate()

//close nfc object
Inv_mobile_nfc.of_close()
```

3 Appeon Resize PBL Reference

This chapter lists the syntax and code examples of the programming interfaces provided by the Appeon Resize PBL, to resize and move the controls within a window. For more about how Appeon Resize PBL works, refer to Section 2.4.2, “Appeon Resize PBL” in *Mobile UI Design & Development Guide*.

3.1 eon_cst_resize object

The eon_cst_resize object is basically the same as the PFC n_cst_resize object with a little bit modifications and the usage method almost remains the same. The modifications will be discussed below and those that are not discussed here are the same as that of the PFC n_cst_resize object.

3.1.1 Functions

3.1.1.1 of_checkSizeType

Description

Determines whether the specified object is resizable in the current UI layout.

Note: this is not a public function and the internal algorithm may be modified in future by Appeon, therefore please do not use this function directly.

Syntax

```
of_checkSizeType ( eon_cst_resizeattrib a_attr, string as_filterstart, string
as_origstart, string as_origlength )
```

3.1.1.2 of_getComponentName

Description

Returns the name of the container that contains the current object (usually the container is a window, user object, or tab control).

Syntax

```
of_getComponentName ( )
```

3.1.1.3 of_init

Description

Initializes all the registered objects, calculates their sizes and locations, and completes the registration.

Note: this function is called in of_init (userobject) and of_init (window), so please do not directly call this function.

Syntax

```
of_init ( )
```

Return value

1 - success.

3.1.1.4 of_init (userobject)

Description

Analyzes and registers the internal object of the user object.

Syntax

```
of_init ( userobject a_obj )
```

Parameter

userobject - the user object to be analyzed.

Return value

1 - success.

3.1.1.5 of_init (window)

Description

Analyzes and registers the internal object of the window.

Syntax

```
of_init ( window a_obj )
```

Parameter

window - the window to be analyzed.

Return value

1 - success.

3.1.1.6 of_register

Description

Same as `of_register` in the PFC `n_cst_resize` object with the following modifications in algorithm:

- *awo_control* is already registered and will not directly return any value, but will find and re-register the corresponding object according to the current parameters. Therefore, users can call the registration functions to re-register the specified object according to their needs, even if the object has been automatically registered via the Apeon resizing algorithm.
- Determines whether the object is resizable or unresizable according to the type of the *awo_control* object.

Syntax

```
of_register ( windowobject awo_control, boolean ab_scale, integer ai_movex,  
integer ai_movey, integer ai_scaleWidth, integer ai_scaleHeight)
```

Parameter

Same as that of the PFC `n_cst_resize` object.

Return value

Same as that of the PFC `n_cst_resize` object.

3.1.1.7 of_setResizeType

Description

Determines the resizing behavior of the object according to the resize type (resizable or unresizable) of surrounding controls.

Note: this is not a public function, and the internal algorithm may be modified in future by Appeon, therefore, please do not use this function directly.

Syntax

`of_setResizeType (powerobject awo_control, boolean ab_leftsizetype, boolean ab_upsizetype, boolean ab_selfsizable)`

3.1.1.8 of_typeOf

Description

Same as `of_typeOf` in the PFC `n_cst_resize` object but with additional evaluation to the `MonthCalender` and the `DatePicker` objects in PowerBuilder 12.5 or later versions.

Syntax

`of_typeOf (powerobject awo_control)`

Parameter

Same as that of the `n_cst_resize` object.

Return value

Same as that of the `n_cst_resize` object.

3.1.2 Properties

The added properties are as follows:

Table 3.1: Added properties

Type	Name	Description
userObject	<code>iuo_component</code>	The component that contains this object is a user object.
powerObject	<code>ipo_component</code>	The component that contains this object is a window.
int	<code>i_tabIdx[]</code>	Saves the subscript of the Tab control in the component object for adjusting the location and size of the Tab control.

3.2 eon_cst_resizeattrib object

This object is based on the PFC `n_cst_resizeattrib` object and it provides more variables for the `eon_cst_resize` object. Only the internal functions of the `eon_cst_resize` object will use these variables.

3.3 eon_cst_resizecompose object

Scans and initializes the window and the tab controls in the window and triggers the resize event to make the controls in the window adjust their sizes and locations correctly.

3.3.1 Functions

3.3.1.1 of_init

Description

Scans, analyzes, and registers the controls of the specified window.

Syntax

```
of_init ( window a_obj )
```

Return value

1 - success.

3.3.1.2 of_register (a_obj, as_method)

Description

Specifies the location and the resizing rule of a control in the window by calling the corresponding internal interfaces of the `eon_cst_resize` object and its functionality is the same as that of the PFC `n_cst_resize` interface.

Syntax

```
of_register ( dragobject a_obj, string as_method )
```

Parameter

Same as that of the PFC `n_cst_resize` object.

Return value

Same as that of the PFC `n_cst_resize` object.

3.3.1.3 of_register (a_component, a_obj, as_method)

Description

Specifies the location and the resizing rule of a control in the tab control.

Syntax

```
of_register ( userobject a_component, dragobject a_obj, string as_method )
```

Parameter

a_component - Tab control;

a_obj and *as_method* are the same as that of the PFC *n_cst_resize* object.

Return value

Same as that of the PFC *n_cst_resize* object.

3.3.2 Properties

The added properties are as follows:

Table 3.2: Added properties

Type	Name	Description
<i>eon_cst_resize</i>	<i>inv_winResize</i>	Saves the registrations of each control in the window object that needs to resize and adjust location.
<i>eon_cst_resize</i>	<i>inv_userObjResize[]</i>	Saves the registrations of each control in the tab control that needs to resize and adjust the location.

3.3.3 Event

3.3.3.1 pfc_resize

Description

Triggers the resize event of the window and the tab control of the window to automatically adjust the layout.

Parameter

Same as that of the *n_cst_resize* object.

Return value

Same as that of the *n_cst_resize* object.

4 Workarounds for Unsupported Features

Not all of the PowerBuilder features can be supported by Appeon. The unsupported features, if not modified, will be commented out in the generated application files, as a result, the code that contains the unsupported features and other code that is dependent on those unsupported features will stop working.

This chapter provides suggestions to work around the unsupported features that have functional impact on the running of the application. Some cosmetic features, such as **BorderStyle=StyleShadowBox!** property, can be simply ignored if they will not affect the application.

4.1 Objects & Controls

4.1.1 External user object

Description

The external user object is unsupported.

Workaround

Choose one of the following methods:

Method #1: Encapsulate the functions using DLLs.

Method #2: Encapsulate the functions using a user-defined OCX.

4.1.2 UserObject object

4.1.2.1 AddItem function (EAServer only)

Description

AddItem function is unsupported in a user defined object.

Workaround

Call the AddItem function of controls such as ListBox, DropDownListBox, PictureListBox, DropDownPictureListBox, or ListView.

4.1.2.2 DeleteItem function

Description

DeleteItem function is unsupported in a user defined object.

Workaround

Call the DeleteItem function of controls such as ListBox, DropDownListBox, PictureListBox, DropDownPictureListBox, or ListView instead of calling the DeleteItem function of a user-defined object.

4.1.3 System Objects

4.1.3.1 PipeLine object

Description

The Pipeline object including its properties, functions and events is unsupported.

Workaround

Add a server NVO and deploy it to EA Server to work around this issue. The detailed steps are:

Step 1: Encapsulate the Pipeline object including its properties, functions and events in a server NVO, and deploy the NVO to EA Server component.

Step 2: Call the interface function of the component and pass the Pipeline object name and relevant parameters to it.

Step 3: Call the functions, properties or events of the Pipeline object in the component to implement the same functionalities as the original Pipeline object.

The limitation of this workaround is that only the system Pipeline object can be used in the NVO. That is to say, user-defined objects inherited from Pipeline objects are unsupported in the NVO.

4.1.3.2 Application object

4.1.3.2.1 DWMMessageTitle property

Description

The DWMMessageTitle is unsupported.

Workaround

Use a global variable to record the value of the DWMMessageTitle property. Read and write the global variable instead of using the DWMMessageTitle property in the Script.

4.1.3.2.2 FreeDBLibraries property

Description

The FreeDBLibraries property is unsupported.

Workaround

It is a useless property for a Web based application. The property can be simply commented out causing no functionality difference for the deployed application.

4.1.4 System Controls

4.1.4.1 ListView control

4.1.4.1.1 GetItemAtPointer function

Description

GetItemAtPoint is unsupported for ListView control.

Workaround

Replace the use of GetItemAtPointer by using Index argument of RightClicked since they return the same value.

4.1.4.1.2 ItemActivate event

Description

The ItemActivate event for ListView control is unsupported.

Workaround

Copy the script in the ItemActivate event to the Clicked or DoubleClicked event.

4.1.4.2 TreeView control

4.1.4.2.1 SetDropHighlight function

Description

The SetDropHighlight function for TreeView control is unsupported.

Workaround

Use other functions to highlight the item specified in SetDropHighlight as the drop target. For example, change the font of the item or modify the item label.

4.1.4.3 Tab control

4.1.4.3.1 TabPostEvent function

Description

TabPostEvent function of tab control is unsupported.

Workaround

Use TabTriggerEvent instead.

4.2 System Functions

4.2.1 DDE Server functions

Description

The DDE Server functions are unsupported.

Workaround

Use a DLL to work around this issue.

1. Encapsulate the DDE in a DLL.
2. Create a DLL proxy to call the DDE in the DLL.

4.2.2 FileLength64 function

Description

The FileLength64 function is unsupported.

Workaround

Replace the FileLength64 function with FileLength function.

4.2.3 FileReadEx function

The FileReadEx function is supported since Appeon 6.5.

Description

The FileReadEx function is unsupported in Appeon 5.x, 6.0, 6.1 and 6.2.

Workaround

Replace the FileReadEx function with FileRead function.

4.2.4 FileSeek64 function

Description

The FileSeek64 function is unsupported.

Workaround

Replace the FileSeek64 function with FileSeek function.

4.2.5 FileWriteEx function

The FileWriteEx function is supported since Appeon 6.5.

Description

The FileWriteEx function is unsupported in Appeon 5.x, 6.0, 6.1 and 6.2.

Workaround

Replace the FileWriteEx function with FileWrite function.

4.2.6 FromAnsi function

Description

The FromAnsi function is unsupported.

Workaround

Replace the FromAnsi function with String function.

Example

The original code:

```
FromAnsi ( blob )
```

The modified code in PowerBuilder 9:

```
String(blob)
```

The modified code in PowerBuilder 10:

```
String(blob, EncodingANSI!)
```

4.2.7 FromUnicode function

Description

The FromUnicode function is unsupported.

Workaround

Replace the FromUnicode function with String function.

Example

The original code:

```
FromUnicode(blob)
```

The modified code:

```
String(blob)
```

4.2.8 Garbage Collection functions

Description

The Garbage Collection functions (GarbageCollect, GarbageCollectGetTimeLimit & GarbageCollectSetTimeLimit) are unsupported.

Workaround

Appeon has its own mechanism to delete unused objects and classes. This function can be simply commented out, causing no loss to the deployed application.

4.2.9 GetLibraryList function

Description

The GetLibraryList function is unsupported.

Workaround

Create a new table holding all the library information in the database. Use this table to check the information in the library instead of using the GetLibraryList function.

4.2.10 Help functions

Description

Using the Help functions such as ShowHelp or ShowPopupHelp to display a HLP file or a CHM (HTML Help) file is unsupported.

Workaround

To enable you to display a HLP file or a CHM file, replace the Help functions by applying a StaticHyperLink, PictureHyperLink or HyperLinkToURL:

Step 1: Upload the Help file to your Web Server.

Step 2: Create a StaticHyperLink or PictureHyperLink control in the application and set the link of the control to the URL of the Help file. You can also apply the HyperLinkToURL function in the application for accessing the URL of the Help file.

Example

This example uses Appeon Help as the HLP file. First upload the HLP file to the Web Server (in this example Appeon Help is located in <http://www.appeon.com/support>).

Workaround #1: Open the HLP or CHM file with a StaticHyperLink control.

Create a StaticHyperLink control in the application. Add the Appeon Help URL (<http://www.appeon.com/support>) to the URL property of StaticHyperLink control.

Workaround #2: Apply the HyperLinkToURL function to access the URL of the Help file.

Add the following code to the relevant location in the application:

```
Inet internet
Internet = create Inet
internet.HyperLinktoURL("http://www.appeon.com/support/manuals")
```

4.2.11 HyperLinkToURL function

Description

While the HyperLinkToURL function is supported, it cannot link to the URL of an Appeon application. This is because an Internet Explorer process only supports one Appeon application at a time. Executing the HyperLinkToURL function from one Appeon application to open another will cause Internet Explorer to shut down.

Workaround

Method 1: Using the Run function to open the second application

Step 1: Replace the call to the HyperLinkToURL function with a call to the Run system function. Comment out the script in the HyperLinkToURL button and add the following script:

```
run("IEXPLORE.EXE http://192.0.2.189:8080/b237293/",Maximized!);
```

In the code example, "http://192.0.2.189:8080/b237293" can be replaced with the URL of an Appeon application.

Step 2: At the Web server, copy the IEXPLORE.EXE to *C:\WINNT\system* or *C:\WINNT\system32*. Alternatively, add the directory *C:\Program Files\Internet Explorer* in the PATH variable.

Method 2: Using the PictureHyperLink URL property to open the second application

Step 1: Add a PictureHyperLink control to the first application, from which the second application will be opened.

Step 2: Specify the URL of the second application as the URL of the PictureHyperLink control.

4.2.12 LibraryDirectory function

Description

The LibraryDirectory PowerScript function is unsupported.

Workaround

The LibraryDirectory function is used to get a list of all objects or just objects of a specified type in a PowerBuilder library.

Case #1

If the LibraryDirectory function is used in other unsupported Library functions, there is no workaround available. Comment out the script related to the Library functions and the relevant functionality will be lost.

Case #2

If the LibraryDirectory function is used to get the list of all objects in the PowerBuilder Library and check whether a particular object exists, the workaround steps are:

Step 1: Create a table in the database to store the object names of the PowerBuilder library that is passed to the LibraryDirectory function.

Step 2: Use a SQL statement to check whether a particular object exists in the table.

4.2.13 PopulateError function

Description

The PopulateError function is unsupported.

Workaround

Appeon has its own error-handling mechanism. This function can be directly commented out, causing no loss to the deployed application.

4.2.14 Registry function

Description

System registry functions can read and write registry entries, keys, and values on a Windows PC. But unlike Windows system, iOS system has no such registry, so these functions cannot be directly executed in iOS system.

Workaround

Appeon offers an option of "Appeon emulation registry" in AEM to allow the mobile application to read and write the mock registry file stored in the Appeon Server database. For detailed information, please refer to **Appeon Server Configuration Guide for .NET | AEM User Guide | Application | PB Features | Registry Mode**.

Note: Mobile applications can only read Appeon emulation registry.

4.2.15 Shared Object functions

Description

Shared Object functions like SharedObjectGet, SharedObjectRegister are unsupported by Appeon, so they cannot be applied to show the process of an event with a progress bar, or control the progress of two different processes.

Workaround

The following is the workaround for showing the process of an event to the users:

Before the execution of the event, prompt a message box at the client to inform the user that the event (such as the retrieve of a large amount of data) is taking place and the event may take relatively long.

4.2.16 SignalError function

Description

The SignalError function is unsupported.

Workaround

Appeon has its own error-handling mechanism. This function can be directly commented out, causing no functionality loss to the deployed application.

4.3 PowerScript Reference

4.3.1 PowerScript Topics

4.3.1.1 Calling functions and events

4.3.1.1.1 Overriding system function

Description

Overriding system functions is unsupported.

Workaround

Create a user-defined function and rewrite the user code in the system function into the user-defined function.

Example

In the original application, the SetTransObject function of the *u_dw* object contains the following user code:

```
trans_current = atrans_current return super::settransobject(itrans_current)
```

In the converted Web application, remove the preceding user code from the SetTransObject function, then create a user-defined function *uf_settransobject* and rewrite the following code:

```
itrans_current = atrans_current settransobject(itrans_current)
```

When the SetTransObject function of the *u_dw* object is called in the Web application, the user code is executed in the newly created *uf_settransobject* function.

4.3.1.1.2 Passing arguments to functions and events

Duplicate arguments for a function

Description

Repetitively referring objects as arguments for a function is unsupported. Use the following workaround and example to change it into a supported format.

Workaround

Assign the repetitively referenced object to multiple different variables and pass these variables to a function.

Example

The original script:

```
w_1.wf_1(dw_1,dw_1)
```

Re-write it using the following format:

```
u_dw ldw_1, ldw_2
ldw_1 = dw_1
ldw_2 = dw_1
w_1.wf_1(ldw_1,ldw_2)
```

Passing Menu object as a reference parameter

Description

Passing Menu object as a reference parameter is unsupported.

Workaround

Pass the Menu object by value or as read-only instead of by reference. The reason is, when passing by value or as read-only, if you change the properties of the object by value or as read-only, you are changing the original object, which is the same as passing by reference.

Reference parameter

Description

The return values of functions and properties of objects cannot be directly used as reference parameters for functions.

Workaround

Follow the steps below to work around this issue:

Step 1: Create a variable and assign the return value of a function or the property of an object to the variable.

Step 2: Call the variable where the return value or the property is called in the original code.

Step 3: Assign the return value of calling the variable to the original property or object.

Code example

```
// The original code
poptags(theobject.item[theitem])
// After modification
menu lm_menu
lm_menu = theobject.item[theitem]
poptags(lm_menu)
```

4.3.1.2 Declarations

4.3.1.2.1 Shared variables

Description

Shared variables are unsupported. For example, In the *pfcmmain.pbl* of a PFC application, the *pf_c_u_dw* uses a shared variable *snv_property* with its type being *n_cst_dwsrv_property*. The *snv_property* is unsupported.

Workaround

Follow the steps below to work around this issue.

Step 1: Change the variable *snv_property* as an instance variable of *n_cst_appmanager* (usually declared as *gnv*).

Step 2: Open the *pfc_u_dw*, and replace all the "snv_property" with the "gnv_app.snv_property".

Note

Since the parent code is changed, when deploying a new PFC, the same modification needs to be done again.

Other shared variables can also be worked around in this way.

4.3.1.3 Language basics

4.3.1.3.1 Null values

Description

The Null value calculation in Appeon is quite different from that in PowerBuilder. This is caused by the different calculation methods used in PowerScript and JavaScript.

Workaround

When using expressions with Null value calculations, you should add conditional statements to ensure that you get the correct result.

Example #1

A and/or B is/are likely to carry a Null value in the following assignment:

```
C = A + B
```

Re-write it using the following format:

```
IF IsNull(A) or IsNull(B) THEN
  SetNull(C)
ELSE
  C = A + B
END IF
```

Example #2

Another example for relational operations:

```
IF A = B then
  MessageBox ("Return Value", "True")
ELSE
  MessageBox ("Return Value", "False")
END IF
```

Re-write it using the following format:

```
IF IsNull(A) or IsNull(B) then
  MessageBox ("Return Value", "False")
ELSEIF A = B then
  MessageBox ("Return Value", "True")
ELSE
  MessageBox ("Return Value", "False")
END IF
```

Note

The following table shows the different return values that PowerBuilder and Appeon will produce when an expression contains at least one null value. In these examples, the values of variables A and B are both null:

Expressions	Return Value in PowerBuilder	Return Value in Apeon
A+1	Null	1
A+B	Null	Null
A*B	Null	0
A=1 (relational)	Null	False
A<>1	Null	True
NOT (A=1)	Null	True
A=A (relational)	Null	True
A=B	Null	True
IsNull(A=1)	True	False

4.3.2 PowerScript Statements

4.3.2.1 GOTO statement

Description

PowerScript GOTO statements and Label are unsupported. Using GOTO is not recommended in structured programming.

Workaround

Analyze the code that uses GOTO and re-write the code in a structured way by applying IF ... THEN ... statements.

If the statement that the GOTO label is associated with has a return value, place the statement in a user function, and in place of the GOTO statement, call the user function.

If the statement that the GOTO label is associated with has no return value, place the statement directly in place of the GOTO statement.

Example

The original script:

```
IF sle_1.text = "" THEN GOTO hide_sle_1
sle_1.text = ""
hide_sle_1:
sle_1.visible = false
MessageBox ("","SingleLineEdit sle_1 is cleared and hidden.")
```

Re-write it using IF ... THEN... statement:

```
IF sle_1.text = "" THEN
sle_1.visible = false
MessageBox ("","SingleLineEdit sle_1 is cleared and hidden.")
ELSE
sle_1.text = ""
sle_1.visible = false
MessageBox ("","SingleLineEdit sle_1 is cleared and hidden.")
END IF
```

Re-write it using CHOOSE CASE statement:

```
CHOOSE CASE sle_1.text
CASE ""
sle_1.visible = false
MessageBox ("", "SingleLineEdit sle_1 is cleared and hidden.")
CASE ELSE
sle_1.text = ""
sle_1.visible = false
MessageBox ("", "SingleLineEdit sle_1 is cleared and hidden.")
END CHOOSE
```

4.3.3 SQL Statements

4.3.3.1 Stored procedure with Null output

Description

It is currently unsupported to use NULL value in an OUTPUT parameter when declaring a stored procedure.

Workaround

Set a variable and set it to NULL. Then use the variable in the OUTPUT parameter to provide the same functionality.

Example

Original code:

```
// Declare the procedures
Declare SPgetseniva procedure for SPgetseniva &
@codpes = :l_osp_codpes, &
@nomsis = :s_osp_systemname, &
@staobt = :s_osp_systemstatus,&
@numseniva = null output ;
```

Modified code:

```
integer li_return
SetNull(li_return)
// Declare the procedures
Declare SPgetseniva procedure for SPgetseniva &
@codpes = :l_osp_codpes, &
@nomsis = :s_osp_systemname, &
@staobt = :s_osp_systemstatus,&
@numseniva = :li_return output ;
```

4.3.4 PowerScript Events

4.3.4.1 Help event

Description

The Help event is unsupported.

Workaround

Copy the script of the Help event to the Key event. In the Key event, use the KeyDown function to check whether the user has pressed the F1 key and if the KeyDown function returns true, execute the script that is originally in the Help event.

4.3.4.2 Other event

Description

The Other event is unsupported.

Workaround

Move the script in the Other event to the supported events.

4.4 DataWindow Reference

4.4.1 DataWindow operators

4.4.1.1 DataWindow operator precedence

Description

The AND and OR operators in a DataWindow expression have the same precedence in PowerBuilder, but in Appeon, the AND operator has higher precedence.

Workaround

When there are both AND and OR operators in a DataWindow expression, you should use parentheses to get the correct precedence effect.

Example

The following code examples are for a DataWindow expression that sets the column text color. The OR operator will be evaluated first in PowerBuilder, but in Appeon, the following script will evaluate the AND operator first.

Original code:

```
If(Left(GetText(), 1) = 'V' OR Left(GetText(), 1) = 'A' AND Mod(GetRow(), 2) = 1, 236, 243433) // incorrect
```

To have the OR operator evaluated first, add a pair of parentheses to the OR expression:

```
If((Left(GetText(), 1) = 'V' OR Left(GetText(), 1) = 'A') AND Mod(GetRow(), 2) = 1, 236, 243433) // correct
```

4.4.2 DataWindow Object Properties

4.4.2.1 Retrieve.AsNeeded

Description

The Retrieve.AsNeeded property of DataWindow object is unsupported.

Workaround

Write script to have the data displayed in batches instead of at one time. The detailed steps are:

Step 1: Create a user object. The user object contains four Picture buttons. The Picture buttons represents First page, Previous page, Next page and Last page respectively.

Step 2: Place the user object in the window that contains the DataWindow in which the large quantity of data will be displayed.

Step 3: Define window functions and modify the original scripts to have them work along with the user object to provide the functionality of displaying data in batches.

4.4.3 DataWindow control

4.4.3.1 Functions

4.4.3.1.1 CanUndo function

Description

CanUndo function is unsupported.

Workaround

Replace the CanUndo function with ModifiedCount function.

Example

The original code:

```
if dw_1.CanUndo() Then
dw_1.Undo()
end if
```

The modified code:

```
if dw_1.ModifiedCount() > 0 Then
dw_1.Undo()
End if
```

4.4.3.1.2 DBCancel function

Description

DBCancel function is unsupported.

Workaround

Appeon has its own mechanism to retrieval data. This function can be simply commented out.

Functionality difference

All the retrieved data will be returned at the same time.

4.4.3.1.3 GenerateResultSet function

Description

The ResultSet system object and GenerateResultSet method are unsupported.

Workaround

To work around the GenerateResultSet method, we have the following two methods:

Method #1: Use GetFullState to retrieve data from a DataWindow and then use SetFullState to apply the blob returned from GetFullState to another DataWindow.

Method #2: Use the datawindow.data property to retrieve data from a DataWindow into a string and insert data into the DataWindow from the string by ImportString.

4.4.3.1.4 GetTrans function

Description

The DataWindow GetTrans function is unsupported.

Workaround

Comment out the unsupported script. Instead use the SetTransObject connection method, to assign a programmer-specified transaction object or a global transaction object called SQLCA to a DataWindow control or DataStore.

To use SetTransObject, write code that performs the following tasks:

1. Set up the transaction object by assigning values to its fields (usually in the application's Open event).
2. Connect to the database using the SQL CONNECT statement and the transaction object (in the Open event for the application or window).
3. Call SetTransObject to associate the transaction object with the DataWindow control or DataStore (usually in the window's Open event).
4. Check the return value from the Update method and follow it with an SQL COMMIT or ROLLBACK statement, as appropriate.

4.4.3.1.5 ReselectRow function

Description

The DataWindow ReselectRow method is unsupported.

Workaround #1

If the DataWindow's source table has a primary key, to work around this issue, follow the steps below:

Step 1: Use GetItem() to get the value of the primary column in the current row.

Step 2: Use SQL statement to retrieve data for the current row according to the value of the primary key.

Step 3: Use SetItem() to assign values to each column in the current row.

Step 4: Change the status of the current row to "NotModified!"

Workaround #2

Replace the ReselectRow function with the Retrieve function.

4.4.3.1.6 ResetTransObject function

Description

The ResetTransObject function of a DataWindow control or a DataStore is unsupported.

Workaround

Replace the ResetTransObject function with SetTransObject function.

Example

The original script:

```
dw_1.ResetTransObject()
```

The modified script:

```
dw_1.SetTransObject(transaction)
```

4.4.3.1.7 SaveAsAscii function

Description

SaveAsAscii function for DataWindow is unsupported.

Workaround

Use a server NVO to work around the SaveAsAscii function.

Step 1: Encapsulate the SaveAsAscii function into a NVO and deploy the NVO to EAServer.

Step 2: Call to the NVO SaveAsAscii function, and store the generated ASCII text file to the Web Server web root.

Step 3: Send the URL of the generated files to client side using the HyperLinkToURL function of Inet object.

Note

The DataWindow that calls the SaveAsASCII must be deployed to EAServer as well.

All the DataWindow SaveAs types that are unsupported can be worked around in this way.

4.4.3.1.8 Scroll function

Description

The Scroll function of DataWindow control is unsupported.

Workaround

Replace the Scroll function with ScrollToRow, ScrollPriorPage or ScrollNextRow function.

Functionality difference

The event sequence of the corresponding function will be triggered.

Example

The original script:

```
dw_1.Scroll(3)
```

The modified script. Please note that the event sequence of the ScrollToRow function will be triggered.

```
dw_1.ScrollToRow(dw_1.GetRow() + 3)
```

4.4.3.1.9 SetTrans function

Description

Using SetTrans to establish a connection to the transaction object is unsupported.

Workaround

Use the SetTransObject method.

Example

The original script:

```
i = ids_main.SetTrans(itr_sql)
```

Re-write it using the following format:

```
ids_main.setTransObject(itr_sql)
```

4.4.3.2 Events

4.4.3.2.1 ScrollVertical event

Description

ScrollVertical event is unsupported.

Workaround

You can choose either of the following two ways to work around the ScrollVertical event.

To work around the ScrollVertical event for a single DataWindow

Step 1: Define a user-defined event such as ue_scrollvertical to replace ScrollVertical.

Step 2: Place the same code that you plan to put in ScrollVertical in the user-defined event.

Step 3: In the Timer event of the window that hosts the DataWindow, trigger the user-defined event periodically with the following code:

```
long ll_new_firstrow
ll_new_firstrow = long(dw_2.Object.DataWindow.firstRowOnPage) // Assuming the
DataWindow is dw_2
if ll_new_firstrow <> il_old_first_row then
    dw_2.trigger event ue_scrollvertical()
    il_old_first_row = ll_new_firstrow
end if
```

Step 4: Call the Timer event in the Open event of the window with the following code:

```
timer(0.005)
```

To work around the ScrollVertical event for ancestor DataWindow

You can code the workaround once in the ancestor DataWindow and then apply it for as many DataWindow as you want.

Supposing the ancestor window is w_sheet and the ancestor DataWindow is u_dw:

Step 1: Start the Timer event in the Open event of w_sheet.

```
string ls_timer_interval
ls_timer_interval = &
ProfileString( gnv_app.of_getappinifile( ), "timer", "interval", "0.005" );
timer(Dec(ls_timer_interval));
```

Step 2: In the pfc_postopen event of w_sheet, add the following code for getting all the DataWindows.

```
of_sb_get_dwobjects( this.control );
```

Step 3 - Add a new function called of_sb_get_dwobjects (windowobject awo_control[]) in w_sheet.

```
int          i;
tab         lt_tab;
userobject  luo_temp;
u_dw       lu_dw

for i = 1 to upperbound( awo_control )
  if( TypeOf( awo_control[i] ) = Tab! ) then
    lt_tab = awo_control[i];
    of_sb_get_dwobjects( lt_tab.control ); // Recursive call
  elseif( TypeOf( awo_control[i] ) = UserObject! ) then
    luo_temp = awo_control[i];
    of_sb_get_dwobjects( luo_temp.control ); // Recursive call
  elseif( TypeOf( awo_control[i] ) = DataWindow! ) then
    // iu_dw is an instance variable that is an array of u_dw datawindow controls
    iu_dw[ upperbound(iu_dw) + 1 ] = awo_control[i];
  end if
next
```

Step 4: In the Timer event for w_sheet, add the following code to check row changes happened to all the DataWindows:

```
int i
for i = 1 to upperbound(iu_dw)
  iu_dw[i].of_sb_verticallscroll()
next
```

Step 5: Add an instance variable for u_dw.

```
long il_old_first_row = -1;
```

Step 6: Add a new function of_sb_verticallscroll() for u_dw.

```
long ll_new_firstrow, ll_counter // Check if only one row per page is being
  displayed in dw
ll_new_firstrow = long( this.object.DataWindow.FirstRowOnPage);
if (ll_new_firstrow <> il_old_first_row) then
  il_old_first_row = ll_new_firstrow
  this.trigger event ue_scrollvertical( )
end if;
```

Step 7: Define a user-defined event for u_dw such as ue_scrollvertical to replace ScrollVertical.

Step 8: Place the same code that you plan to put in ScrollVertical in the user-defined event.

4.4.3.2.2 RetrieveRow event

Description

The RetrieveRow event is unsupported.

Workaround

Move the relevant logic to the RetrieveEnd event.

Example

The original script in the RetrieveRow event:

```
if row < 1 then return
// for every single time, check whether the row should be deleted or not.
if f_find(istr_dwnum.dw,istr_dwnum.id
+"='"+this.getitemstring(row,istr_dwnum.id)+"'")>0 then
  this.deleterow(row)
end if
```

The modified RetrieveEnd event (preceding logic is moved to the RetrieveEnd event):

```
long li_row
// loop all rows retrieved from the database and find out which row should be
deleted.
for li_row = 1 to rowcount
  if f_find(istr_dwnum.dw,istr_dwnum.id+"='"+this.getitemstring(li_row,istr_dwnum.id
+'')>0 then
    this.deleterow(li_row)
    li_row --
  end if
end if
next
```

4.4.3.3 Properties

4.4.3.3.1 LiveScroll property

Description

The LiveScroll property for DataWindow control is unsupported.

Workaround

When the LiveScroll property is enabled, it does not take effect on the Web. A row can only be selected by mouse clicking.

4.4.4 Controls in a DataWindow

4.4.4.1 Large Binary/Text database OLE object

Description

The Large Binary/Text database OLE object for DataWindow is unsupported.

Workaround

This workaround only applies to the BitMap OLE object.

If a column with any large Binary/Text Database OLE object is used for displaying graphs, do the following steps to work around this issue:

Step 1: Replace the column with a Graph control.

Step 2: Retrieve the content of the column in the table related with the DataWindow using a SELECTBLOB SQL statement.

Step 3: Call the SetPicture function of this Graph control.

4.4.4.2 Column control

4.4.4.2.1 Char data type column

Description

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*, but the value you get in PowerBuilder is *char(n)*.

Workaround

Add one more condition while using the returned value.

Example

The original script:

```
string ls_datatype
ls_datatype = dw_1.object.group_id.coltype
if(ls_datatype = "char(50)") then
    ...
else
    ...
end if
```

Add one more conditional statement, as shown in the following modified script.

The modified script:

```
string ls_datatype
ls_datatype = dw_1.object.group_id.coltype
if(ls_datatype = "char(50)" or ls_datatype = "char") then
    ...
else
    ...
end if
```

4.4.4.2.2 Evaluating DataWindow expressions in scripts

Description

When using global functions in DataWindow expressions to dynamically change the attributes of DataWindow objects at run time, this method doesn't work fine with an Appeon application, for the DataWindow expressions are only evaluated once.

Workaround

Modify a computed expression on the DataWindow in order to force the expression to re-evaluate.

Note: Generally speaking, DataWindow expressions will slow-down the initial display or subsequent refresh of DataWindows. As such, Appeon recommend you reduce the usage of DataWindow expressions if possible, especially in the following situations:

- Avoid using DataWindow expressions for computing and setting column properties.
- Avoid setting sort and filter criteria directly for a DataWindow object. Instead, write the sort and filter criteria in the SQL statement of the DataWindow object. As noted previously, it is faster to use SQL statements than DataWindow functionality.

Example

An expression like this will not re-evaluate itself: Expression: f_color()

After making a change that would cause f_color() to return a different value (i.e. selecting a different preferred color from a drop-down), the application has to slightly change the size of the DataWindow in order to force the expression to re-evaluate.

The modified script:

```
integer li_dw_width, li_dw_height
li_dw_width=dw_1.width
li_dw_height=dw_1.height

dw_1.width=li_dw_width-4           // Squeeze the DW
dw_1.height=li_dw_height-4

dw_1.width=li_dw_width           // Restore to original size
dw_1.height=li_dw_height
```

4.4.5 Functions of DataStore object

Description

CategoryCount, Clipboard, CopyRTF, DataCount for DataStore, FindCategory, FindSeries, GetBorderStyle, GetData, GetDataPieExplode, GetDataStyle, GetDataValue, GetSeriesStyle, PrintCancel, ResetDataColors, SeriesCount, SeriesName, SetDataPieExplode, SetDataStyle, SetDetailHeight, SetSeriesStyle.

Workaround

Replace the DataStore with a DataWindow control and call the corresponding functions of the DataWindow control.

4.5 DBParm parameters in Database

4.5.1 ConnectString parameter

Description

The ConnecString parameter is unsupported.

Workaround

Connect to the database via JDBC, and specify the Data Source name in AEM or dynamically set up the database connection. For more details, please refer to the Chapter 4, *Database Connection Setup* in *Appeon Server Configuration Guide for .NET* or *Appeon Server Configuration Guide for J2EE*.

4.6 Workaround Techniques (for EA Server only)

4.6.1 Appeon GetFullState/SetFullState/GetChanges/SetChanges (Windows only)

Description

PowerBuilder provides four functions for synchronizing DataWindows and DataStores in a distributed application:

- GetFullState
- SetFullState
- GetChanges
- SetChanges

These four functions use a BLOB (Binary Large Object) parameter to store the state information of a DataWindow or DataStore, but Web applications that use JavaScript, such as Appeon, do not support the BLOB data type.

Workaround

To work around the unsupported features (the BLOB parameter) in GetFullState, SetFullState, GetChanges and SetChanges, Appeon has provided two standard user objects: the `appeondatawindow` that inherits from the PowerBuilder system DataWindow control, and the `appeondatastore` that inherits from the PowerBuilder system DataStore object.

These two objects have user-defined functions (GetFullState, SetFullState, GetChanges and SetChanges) that override the original DataWindow/DataStore GetFullState, SetFullState, GetChanges and SetChanges, with a parameter of the String data type.

When coding distributed DataWindow/DataStores for Appeon Web migration, always use the descendants of `appeondatawindow` and `appeondatastore`, and the overriding GetFullState, SetFullState, GetChanges and SetChanges functions that take the String data type as the parameter.

Here are detailed steps on how to perform this workaround.

Example

The following workaround for the GetFullState, SetFullState, GetChanges and SetChanges functions used in the application assumes that the distributed application to be migrated is *application_distribute*.

Step 1: Build a new application called `application_datastore` in PowerBuilder to include server DataStores in *application_distribute*.

Step 2: Add the `appeon_workarounds.pbl` file to the Library Search Path of *application_datastore*.

Step 3: Copy the `EonAXNVO.dll` from the `%AppeonDeveloper%\appeon_workaroundsxxx` directory (**xxx** indicates the corresponding PowerBuilder version) to the same directory as *application_distribute*.

Step 4: Search in *application_distribute* for the DataWindow controls that use GetFullState, SetFullState, GetChanges or SetChanges. Change the declarations of these DataWindows to make them inherit from the `appeondatawindow` user object.

In the following example, the unsupported SetFullState function is called in the Constructor event of the `dw_1` DataWindow in the `w_main` Window:

```

long ll_rc
blob lbob_1
nvo_update lnvo_1
ll_rc = myconnect.CreateInstance(lnvo_1, "p_nvo_update1/nvo_update")
IF ll_rc <> 0 THEN
    MessageBox("Create instance failed", ll_rc)
END IF
lbob_1 = lnvo_1.of_getdata1("d_order_master_four")
IF not IsNull(lbob_1) THEN
    dw_1.SetFullState(lbob_1)
ELSE
    MessageBox("", "Getting data failed.")
END IF

```

To re-write the previous unsupported code:

Change the declaration of *dw_1* DataWindow by editing the source code of the *w_main* Window.

Right-click on *w_main* in the System Tree and choose Edit Source from the context menu. The Source Editor window opens and displays the source code of the *w_main* Window.

Figure 4.1:

```

forward
global type w_main from window
end type
type dw_1 from datawindow within w_main
end type
end forward

global type w_main from window
integer width = 2533
integer height = 1408
boolean titlebar = true

```

Replace the script “type *dw_1* from datawindow within *w_main*” with “type *dw_1* from *apeondatawindow* within *w_main*”.

Save the source code and close the Source Editor window for the *w_main* Window object.

Figure 4.2:

```

forward
global type w_main from window
end type
type dw_1 from apeondatawindow within w_main
end type
end forward

global type w_main from window
integer width = 2533
integer height = 1408
boolean titlebar = true

```

Step 5: Search in *apeon_datastore* for the DataStore objects that use *GetFullState*, *SetFullState*, *GetChanges* or *SetChanges*. Change the declarations of these DataStores to make them inherit from the *apeondatastore* user object.

The following is the script for the *of_getdata1* function of the *nvo_update* user object that is called in Step 3. The function creates a DataStore, retrieves data, and uses the unsupported *GetFullState* function to get and return the state of the DataStore:

```
public function blob of_getdata1 (string as_dataobject);
blob lbob_1
long ll_return
datastore lds_1
lds_1 = Create datastore
lds_1.DataObject = as_dataobject
lds_1.SetTransObject(sqlca)
lds_1.Retrieve()
ll_return = lds_1.GetFullState(lbob_1)
IF IsNull(ll_return) or ll_return < 0 THEN
    SetNull(lbob_1)
END IF
Destroy lds_1
Return lbob_1
end function
```

Modify the function return value to be the String data type, and re-declare the *lds_1* DataStore as *appeondatastore*. The changed code is shown as follows:

```
public function string of_getdata1 (string as_dataobject);
string ls_fullstate
long ll_return
appeondatastore lds_1
lds_1 = Create appeondatastore
lds_1.DataObject = as_dataobject
lds_1.SetTransObject(sqlca)
lds_1.Retrieve()
ll_return = lds_1.GetFullState(ls_fullstate)
IF IsNull(ll_return) or ll_return < 0 THEN
    SetNull(ls_fullstate)
END IF
Destroy lds_1
Return ls_fullstate
end function
```

After applying this workaround, migrate the entire application to the Web using Apeon. For detailed steps, refer to the Section 5.3, “Migrating distributed applications with distributed DataWindows” in *Migration Guidelines for Apeon Web*.

4.6.2 Packaging unsupported features in n-Tier NVOs or DLLs

4.6.2.1 Packaging unsupported objects into n-Tier NVOs

Description

The objects shown in the following table are unsupported, but can be supported if they are packaged into n-Tier NVOs.

ADOResultSet	ArrayBounds	ClassDefinition
ClassDefinitionObject	ConnectionInfo	ConnectObject
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	ErrorLogging	Exception
JaguarORB	mailFileDescription	mailMessage
mailRecipient	mailSession	OLECustomControl
OLERuntimeError	OLEStorage	OLEStream
OLETxnObject	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

Workaround

Encapsulate the relevant script into non-visual objects (NVOs) and deploy the NVOs to EAServer.

Note

You can also add almost any type of business logic into N-Tier NVOs. Many of the other unsupported features that are discussed in this Workarounds Guide can be supported in N-Tier NVOs, such as `DataWindow.RetrieveRow` function in a `DataStore`, etc.

Example

The following steps show how to work around the `ArrayBounds` object.

Step 1: Create *NVO n_cst_rs*.

Step 2: Declare variable instance in *nvo(n_cst_rs)*.

```
resultset irs_resultset
    ADOresultset irs_ADOresultset
```

Step 3: Define the function *Of_SetResultSet(ref oleobject aole)*.

```
// create ds_source
// generate a result set from an existing DataStore
ds_source.GenerateResultSet(irs_resultset)

// create a new ADOResultSet object and populate it
// from the generated result set
lrs_ADOresultset = CREATE ADOResultSet
lrs_ADOresultset.SetResultSet(irs_resultset)
// pass the data in the ADOResultSet object
// to an OLEObject you can use as an ADO Recordset
irs_ADOresultset.GetRecordSet(aole)
```

Step 4: Declare variable instance on the Client side.

```
OLEObject ioo_ADOrecordset
N_Cst_RS IN_Cst_RS
```

Step 5: Write script in the Client side objects.

```
ioo_ADOrecordset = CREATE OLEObject
IN_Cst_RS = CREATE N_CST_RS
IN_Cst_RS. Of_SetResultSet(ioo_ADOrecordset)
// call native ADO Recordset methods on the OLEObject
ioo_ADOrecordset.MoveFirst()
```

Step 6: Deploy the NVO to EA Server.

4.6.2.2 Placing unsupported functions into n-Tier NVOs

Description

Not all PowerScript functions are supported. For detailed information on all unsupported PowerScript functions, refer to Chapter 7, *Functions in Supported PB Features for Apeon Mobile* or in *Supported PB Features for Apeon Web*.

Workaround

If an unsupported PowerScript function meets the requirements for functions that can be packaged, encapsulate the function into a PowerBuilder non-visual user object (NVO).

Functions can be packaged into NVOs, provided that they do not:

- use the PowerScript MessageBox function. (EA Server limitation)
- use application global variables. (EA Server limitation)
- use visual controls or objects. (EA Server limitation)
- use Any or visual control/object data types as parameters for NVO functions and/or events. (EA Server limitation)

GetEnvironment, Randomize, and File functions are examples of functions that can be packaged.

For more information on how to use NVOs, refer to Section 5.1, “Moving unsupported features to Appeon Server as n-Tier NVOs” in *Migration Guidelines for Appeon Web*, or perform the steps shown in the following example.

Example

The PowerScript `GetEnvironment` function is used to populate the Environment object that holds information about the computing platform that the PowerBuilder application is running on. The `GetEnvironment()` and Environment object are unsupported by Appeon. To work around these unsupported features, consider moving them into Appeon Server.

Be aware that the `GetEnvironment` function, when executed in EAServer, returns the system information of the server machine.

The following code, written in the `Clicked` event of the `cb_1` CommandButton in the PowerBuilder Client, gets the complete PowerBuilder version number.

```
Environment lenv_obj
Integer li_return
String ls_pbversion = "" // store PB version number
IF GetEnvironment (lenv_obj) = 1 THEN
    li_return = lenv_obj.PBMajorRevision
    ls_pbversion = ls_pbversion + String(li_return)
    li_return = lenv_obj.PBMinorRevision
    ls_pbversion = ls_pbversion + "." + String(li_return)
    li_return = lenv_obj.PBFixesRevision
    ls_pbversion = ls_pbversion + "." + String(li_return)
ELSE
    ls_pbversion = "-1"
END IF
MessageBox ("PB Version", ls_pbversion)
```

To enable the previous unsupported code to run in EAServer, perform the following steps:

Step 1: Create a PowerBuilder custom class user object in the PowerBuilder Client. Name the custom class user object `nvo_environment`.

Step 2: Add the user function `of_GetPBVersion` to `nvo_environment`. Set the function return value as a string, and add the following code to the function:

```
Environment lenv_obj
Integer li_return
String ls_pbversion = "" // store PB version number
IF GetEnvironment (lenv_obj) = 1 THEN
    li_return = lenv_obj.PBMajorRevision
    ls_pbversion = ls_pbversion + String(li_return)
    li_return = lenv_obj.PBMinorRevision
    ls_pbversion = ls_pbversion + "." + String(li_return)
    li_return = lenv_obj.PBFixesRevision
    ls_pbversion = ls_pbversion + "." + String(li_return)
ELSE
    ls_pbversion = "-1"
END IF
Return ls_pbversion
```

Step 3: Create an EAServer Component project in the PowerBuilder Client that contains information on deploying `nvo_environment` to EAServer, then deploy `nvo_environment` to EAServer within the PowerBuilder IDE.

Step 4: Create an EAServer Proxy project in the PowerBuilder Client that acts as the local representation of the deployed `nvo_environment` EAServer NVO component.

Step 5: When the PowerBuilder Client application starts, create a Connection object named *myconnect*, and connect to the EAServer that hosts the *nvo_environment* NVO component.

Step 6: In the Clicked event of the *cb_1* CommandButton in the PowerBuilder Client, comment all the original code that gets the PowerBuilder version and add the following code to the event:

```
// Define local variables
long ll_rc
string ls_pbversion
nvo_environment lnvo_1
// Instantiate remote component
ll_rc = myconnect.CreateInstance(lnvo_1, "nvo_env/nvo_environment")
IF ll_rc <> 0 THEN
    MessageBox("Create instance failed", ll_rc)
END IF
// Call component method
ls_pbversion = lnvo_1.of_GetPBVersion()
MessageBox ("PB Version", ls_pbversion)
```

Step 7: Disconnect from EAServer when the Client application ends.

5 FAQ & Workarounds

This chapter lists some frequently asked questions and workaround tips regarding the Appeon application architecture or product features.

5.1 How to remove the Internet Explorer menu

Description

When a Web application is opened in Internet Explorer, you may see both the Internet Explorer menu and the Web application menu. It is possible to design an HTML file or C++ program to remove the Internet Explorer menu.

Workaround

Method 1: In the application folder under the Web server Web root, create an HTML file named `appeon.html` for loading the Web application. For example,

```
<html>
<head>
  <title>Appeon Web Library </title>
</head>
<script language="javascript">
  function startApp() {
    g_newWindow = window.open("index.html", "_blank",
      "location=no,titlebar=no,toolbar=no,menubar=no,status=no,resizable=yes",false);
  }
</script>
<body>
  <script language="javascript">
    startApp()
  </script>
</body>
</html>
```

Note: In order for `sFeatures` settings to take effect, the `sName` argument in the `Open` method must be “_blank”. This opens a new Internet Explorer window for `index.html` (the index page of the Web application) and hides the Internet Explorer menu.

Instead of loading `Index.html` as the entry page for Appeon Web applications, load the `appeon.html` file using a URL similar to this: `http://host:port/appname/appeon.html`. When the `Open` method in the `appeon.html` file is triggered, the `Index.html` page will be loaded in a new Internet Explorer window.

Method 2: Create a C++ program that utilizes COM API on the Client side.

The sample code in the C++ program is as follows:

```
// Start a new Internet Explorer as a separate process
IWebBrowser2* pIE = NULL;
HRESULT hr;
hr = CoCreateInstance(CLSID_InternetExplorer, NULL, CLSCTX_SERVER,
    IID_IWebBrowser2, (LPVOID*)&pIE); // if open IE OK
if (SUCCEEDED(hr)) {
    pIE->put_Visible(TRUE);
    pIE->put_AddressBar(FALSE);
    pIE->put_MenuBar(FALSE);
    pIE->put_StatusBar(TRUE);
    pIE->put_ToolBar(FALSE);
    pIE->put_FullScreen(FALSE);
    COleVariant vtEmpty;
    CString strURL = "http://apeonserver:81"; // The URL to be opened by the
    program.
    BSTR bstrURL = strURL.AllocSysString();
    pIE->Navigate(bstrURL, &vtEmpty, &vtEmpty, &vtEmpty, &vtEmpty);
    ::SysFreeString(bstrURL);
}
```

If the user runs the C++ program on the Client machine, the Web application will be opened in an Internet Explorer browser and the display mode of the Internet Explorer browser is specified in the C++ program.

5.2 How to deploy NVO to EAServer 6.1

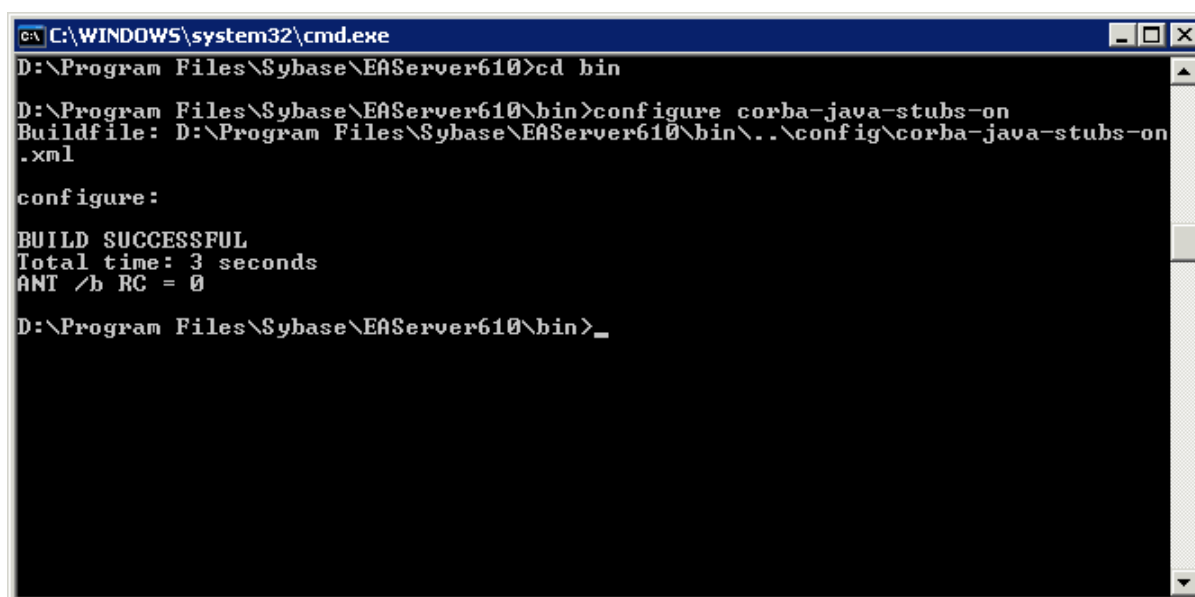
In this section we will demonstrate how to deploy NVO to EAServer 6.1 in PowerBuilder 11. The example NVO used in this section is a simple NVO.

Configuring to Generate Stub and Skeleton

In Windows, run the following command at a prompt window to make sure Stub and Skeleton will be generated during the NVO deployment:

```
%EAServer%\bin\configure corba-java-stubs-on
```

Figure 5.1:



```
C:\WINDOWS\system32\cmd.exe
D:\Program Files\Sybase\EAServer610>cd bin
D:\Program Files\Sybase\EAServer610\bin>configure corba-java-stubs-on
Buildfile: D:\Program Files\Sybase\EAServer610\bin\..\config\corba-java-stubs-on.xml
configure:
BUILD SUCCESSFUL
Total time: 3 seconds
ANT /b RC = 0
D:\Program Files\Sybase\EAServer610\bin>_
```

Adding EAServer profile in PowerBuilder

Use the PowerBuilder component wizard for deploying an NVO to EAServer. Before using the wizard, create an EAServer profile in PowerBuilder, which will connect to the EAServer that hosts Apeon Server.

An EAServer profile stores information on connection settings used to connect to EAServer. The profile you create is used by wizards that require a connection to EAServer.

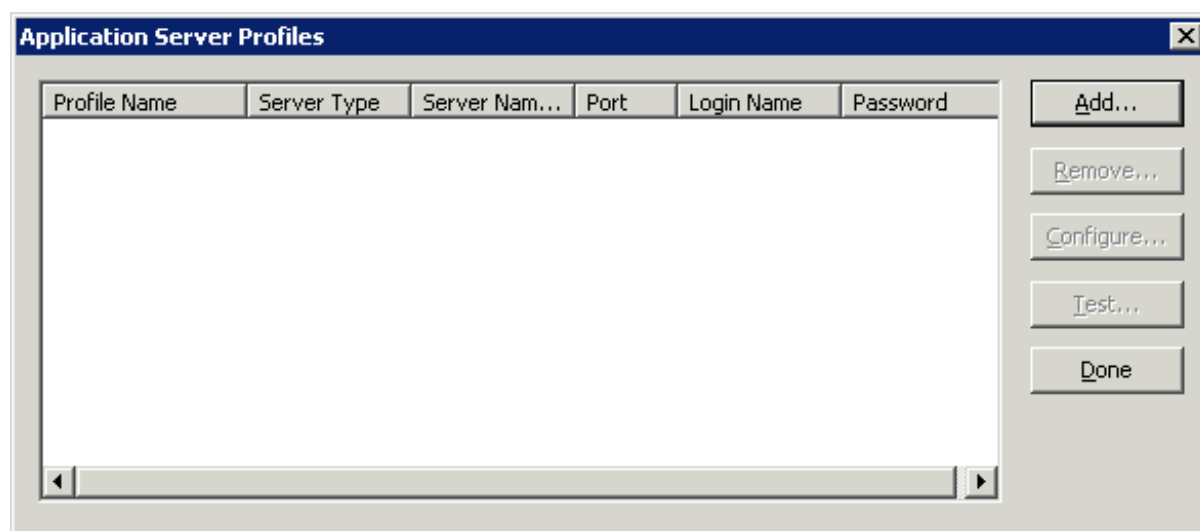
To add an EAServer profile to PowerBuilder:

Step 1: Start EAServer, if it is not already running. Make sure that EAServer is running during the following steps.

Step 2: Click the **Application Server Profile** button in PowerBar1 or choose **Tools | Application Server Profile** from the PowerBuilder menu.

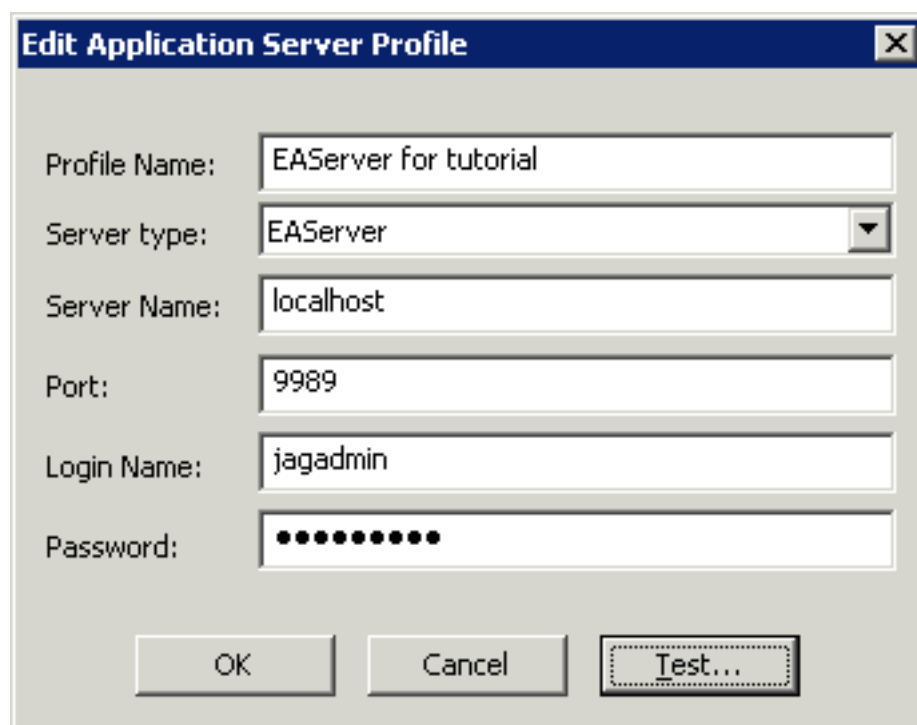
Click **Add** on the Application Server Profile dialog box that is displayed.

Figure 5.2:



Step 3: The Edit Application Server Profile dialog box pops up. Type the information contained in the following table into each field. And then verify the EAServer profile by clicking the *Test* button. Note: EAServer must be running for the test to be successful.

In this field...	You should...
Profile Name	Type <i>EAServer for tutorial</i>
Server Type	Choose EAServer
Server Name	Type <i>localhost</i>
Port Number	Type 9989
Login Name	The default login name is <i>jagadmin</i> . If you have changed the default, enter the correct login name.
Password	There is no password by default. If you have changed the default, enter the correct password.

Figure 5.3:

Step 4: Make sure that connection testing is successful.

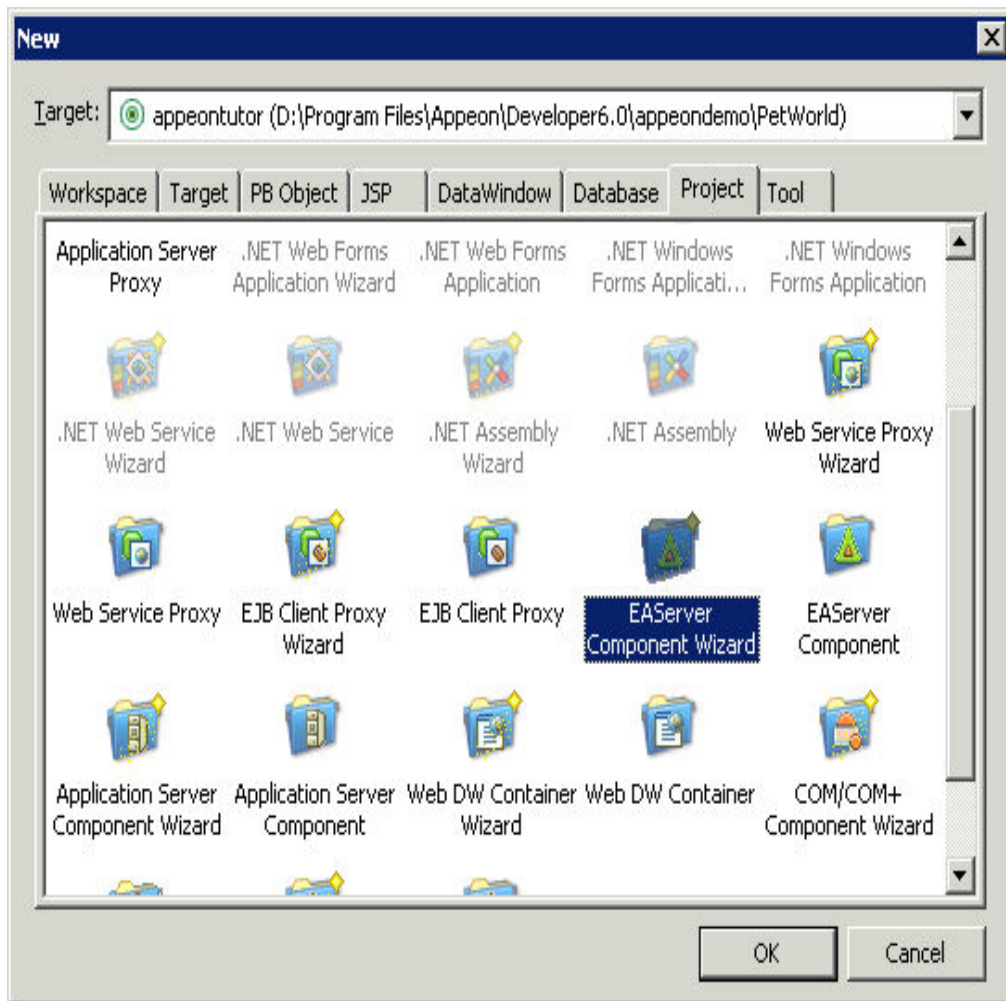
5.2.1 Deploying NVOs to EAServer

Use the EAServer Component Wizard to deploy EAServer components from PowerBuilder.

To deploy the NVO into EAServer:

Step 1: Start EAServer, if it is not already started.

Step 2: Choose **File | New** from the PowerBuilder menu. Then select the **EAServer Component Wizard** icon under the Project tab in the **New** dialog box, and click **OK**.

Figure 5.4:

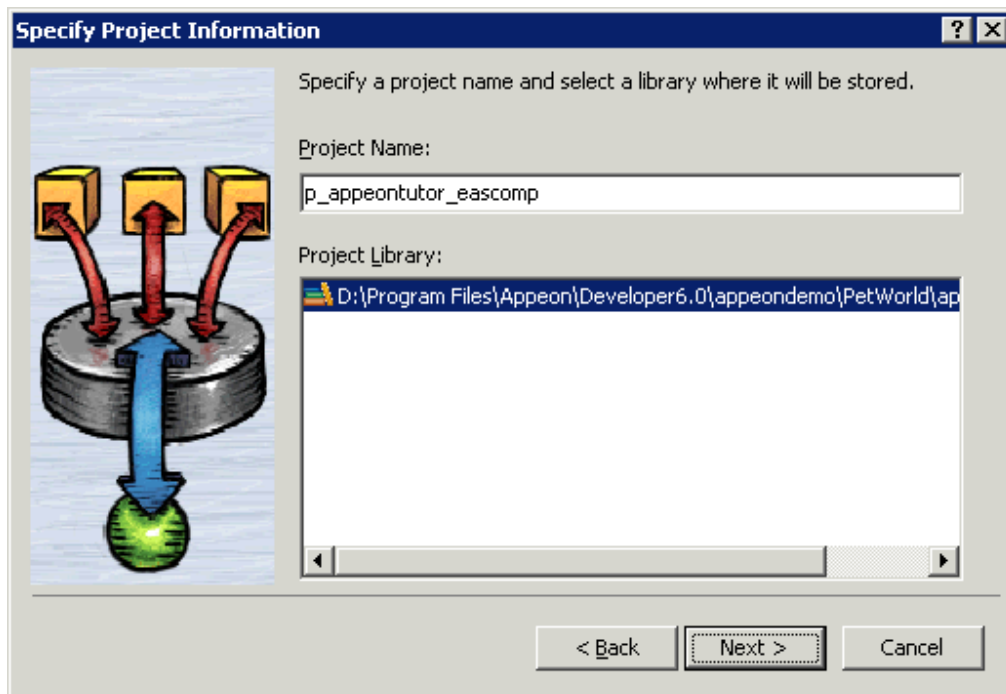
Step 3: The EAServer Component Project Wizard starts. Click **Next**.

Figure 5.5:



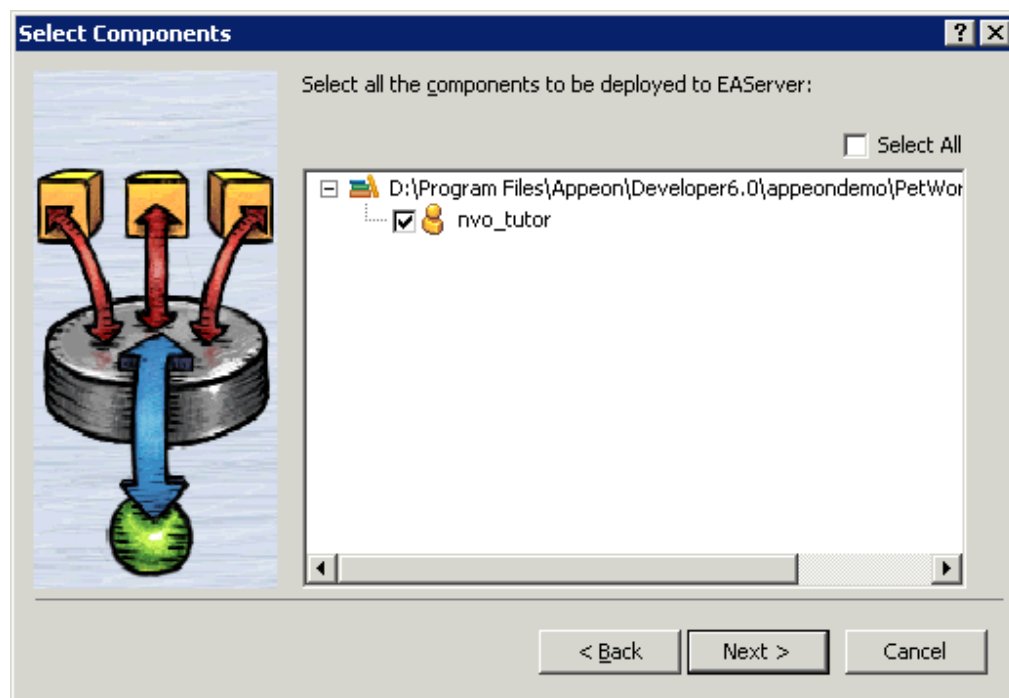
Step 4: Leave the project name as `p_appeontutor_eascomps` and Click **Next**.

Figure 5.6:



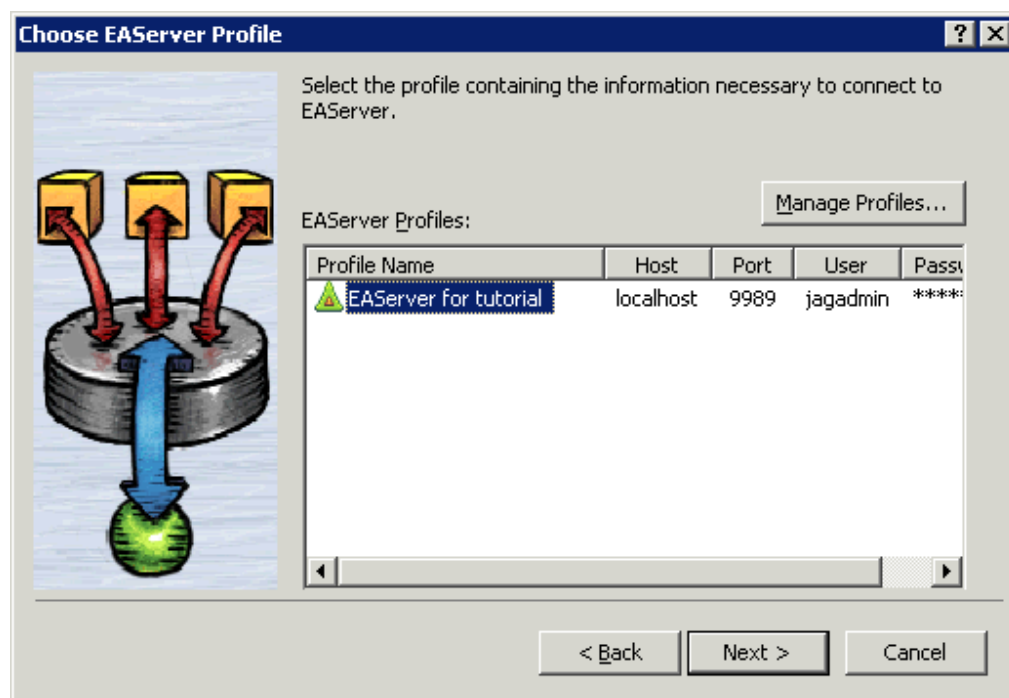
Step 5: Select the `nvo_tutor` NVO that is to be deployed to EAServer. Click **Next**.

Figure 5.7:

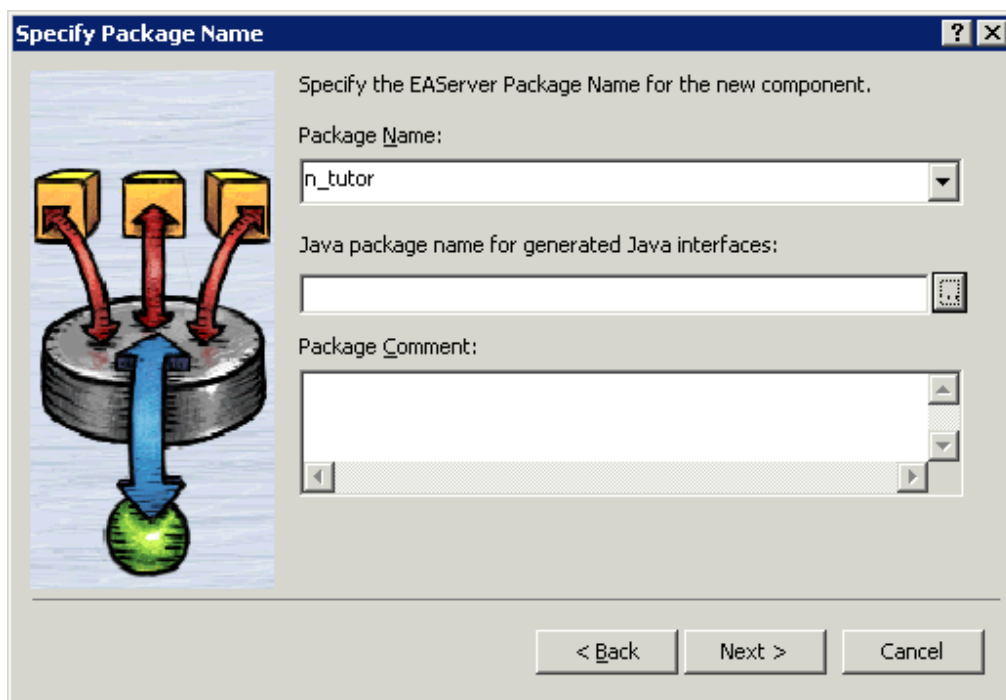


Step 6: Select the **EAServer for tutorial** profile. You created this profile in a previous step. If you have set up profiles outside of the tutorial for other EAServers, there will be more profiles in the list.

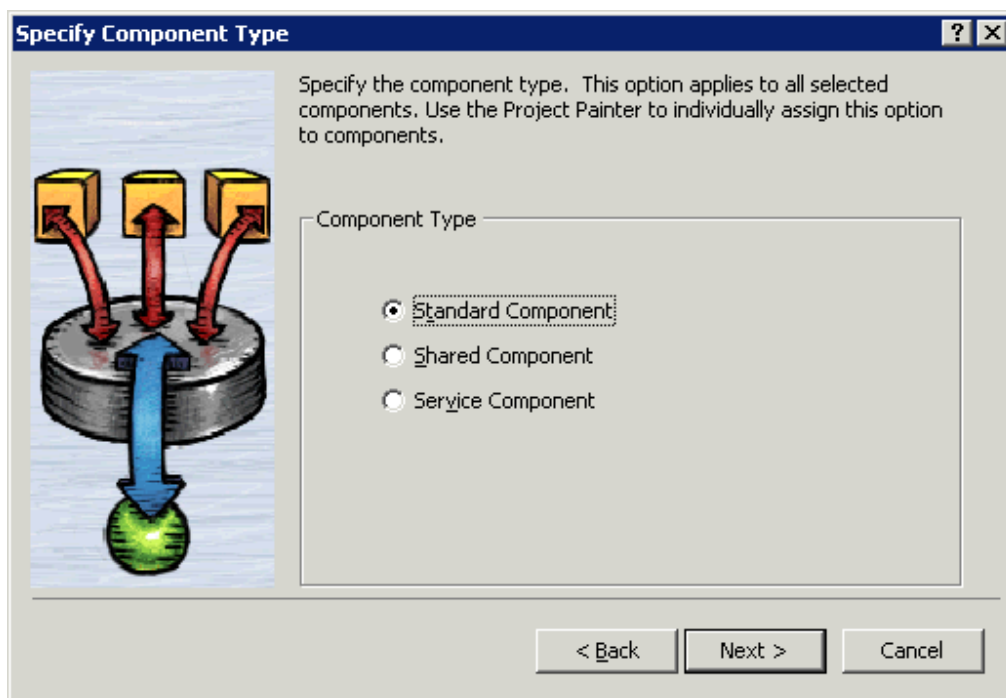
Figure 5.8:



Step 7: Type *n_tutor* into the Package Name dropdown listbox. If necessary type the Java package name according to your own needs. Click **Next**.

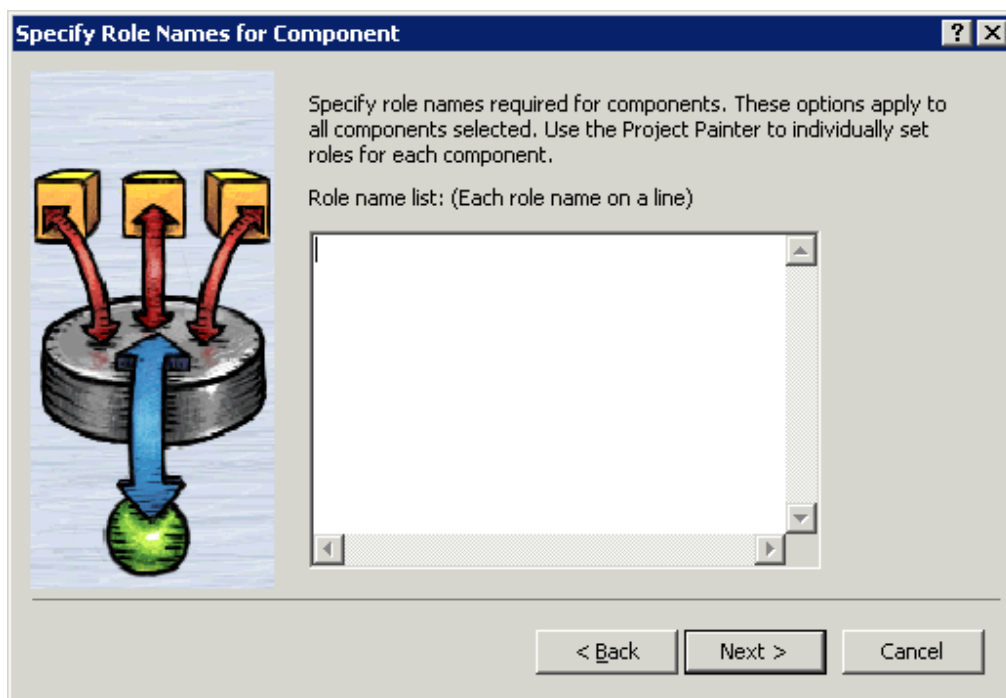
Figure 5.9:

Step 8: Select **Standard Component** on the Specify Component Type dialog box. Click **Next**.

Figure 5.10: PDF

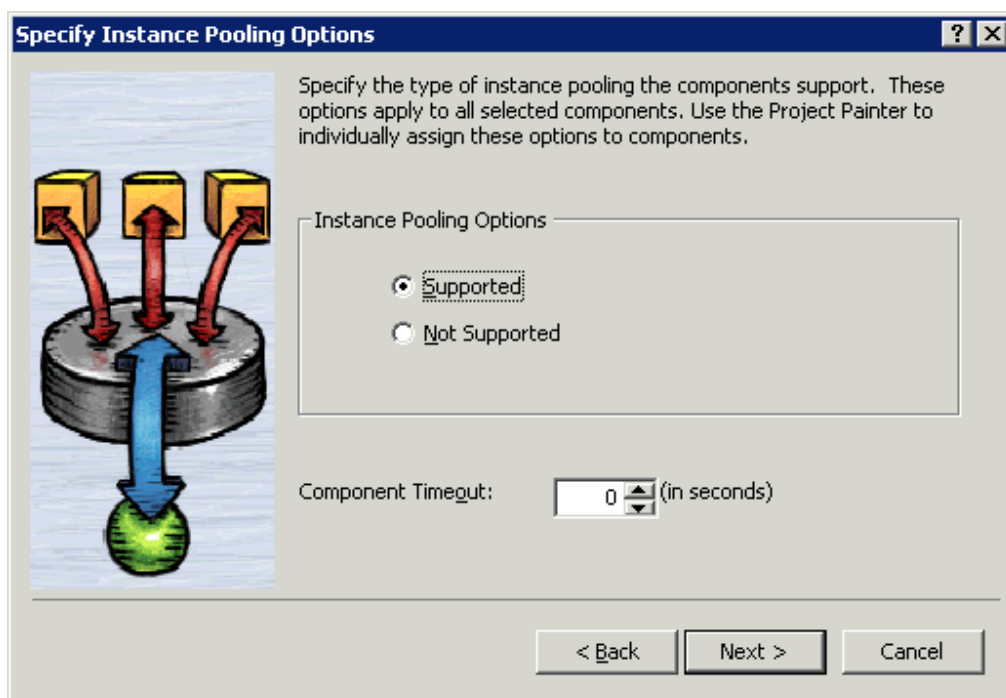
Step 9: Specify the role name if necessary. Click **Next**.

Figure 5.11:



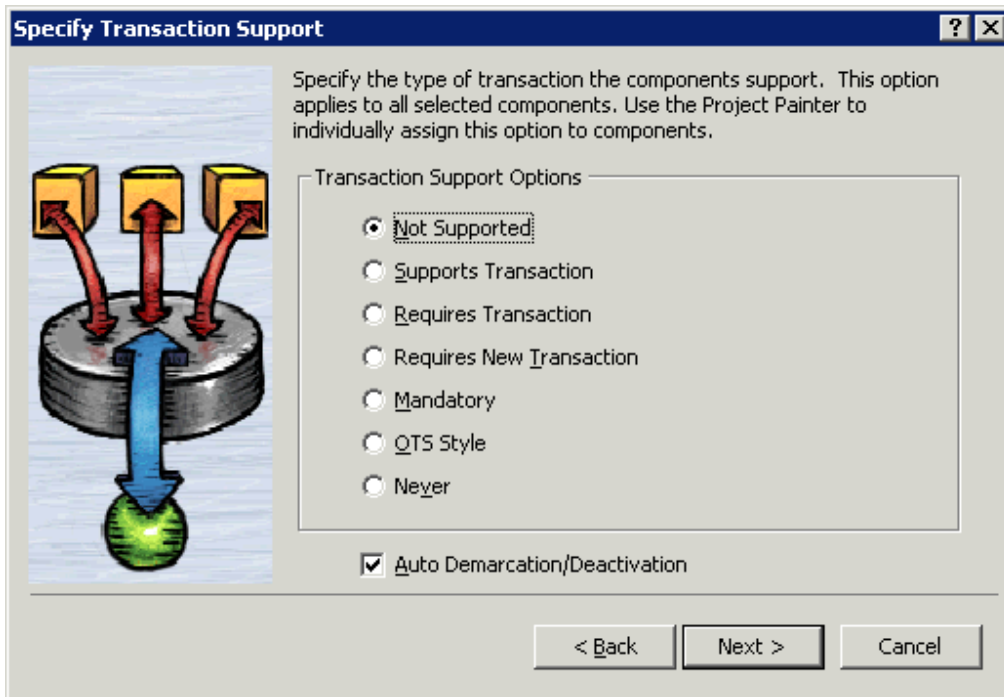
Step 10: Leave the **Supported** option checked for the **Instance Pooling Options** field. Click **Next**.

Figure 5.12:



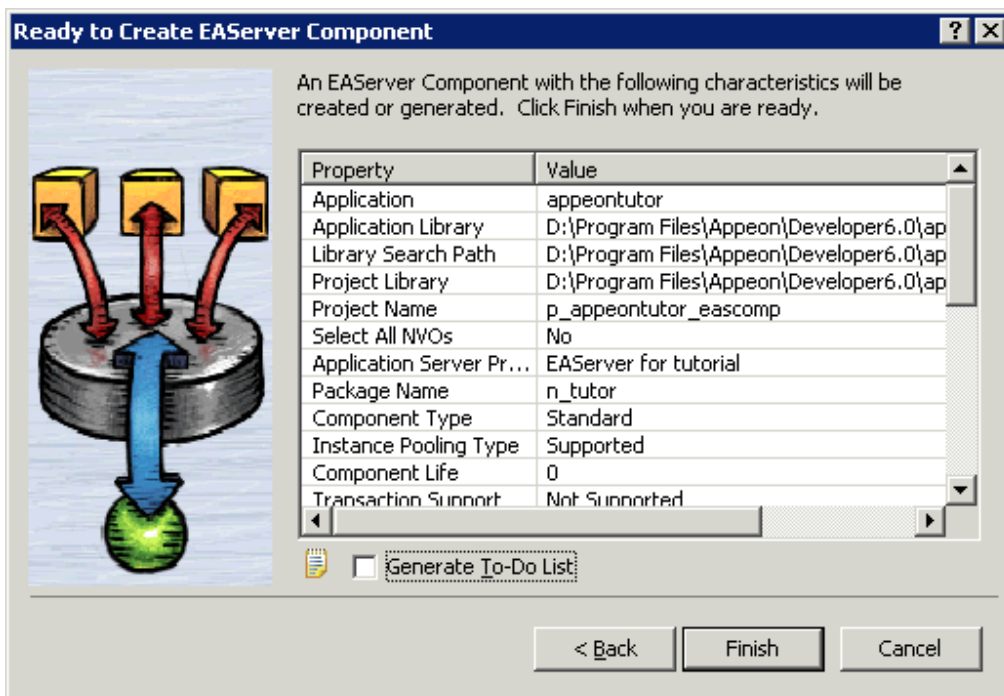
Step 11: In the **Transaction Support Options** field leave the default option as **Not Supported**. Click **Next**.

Figure 5.13:



Step 12: Leave all options at default. Click **Next** until the wizard now comes to its final stage. Click **Generate To-Do List** and **Finish**.

Figure 5.14:

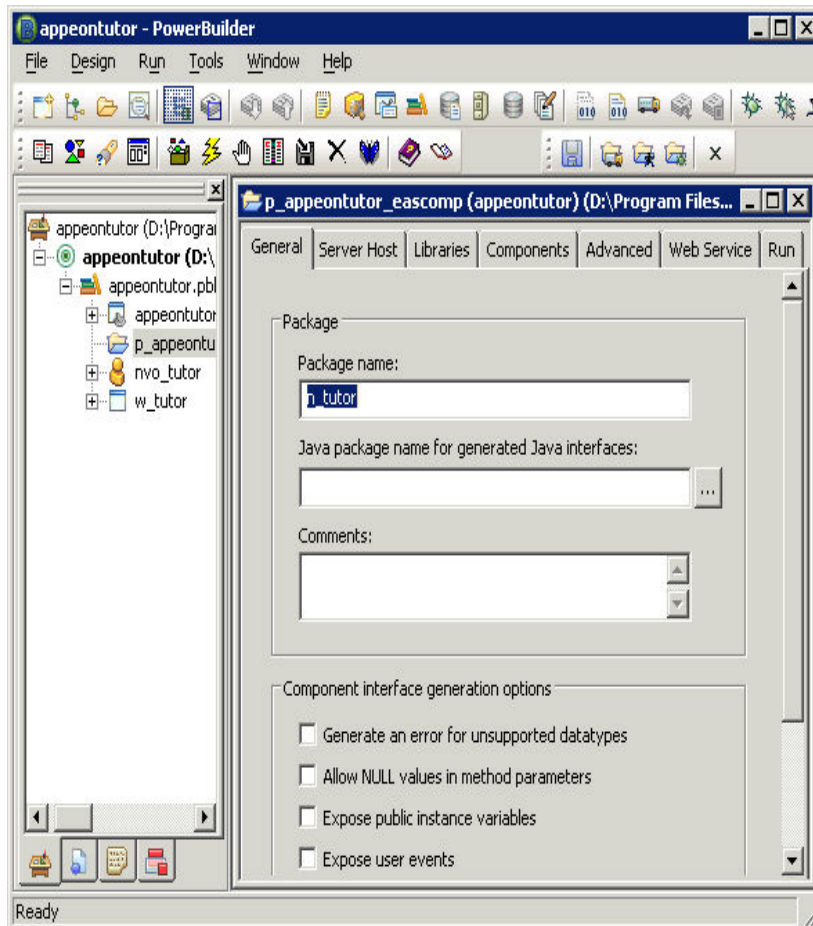


Step 13: The wizard now creates a component project. The Project Painter for the component opens automatically in PowerBuilder, and the project name (p_appeontutor_eascomps) appears in the left system tree list.

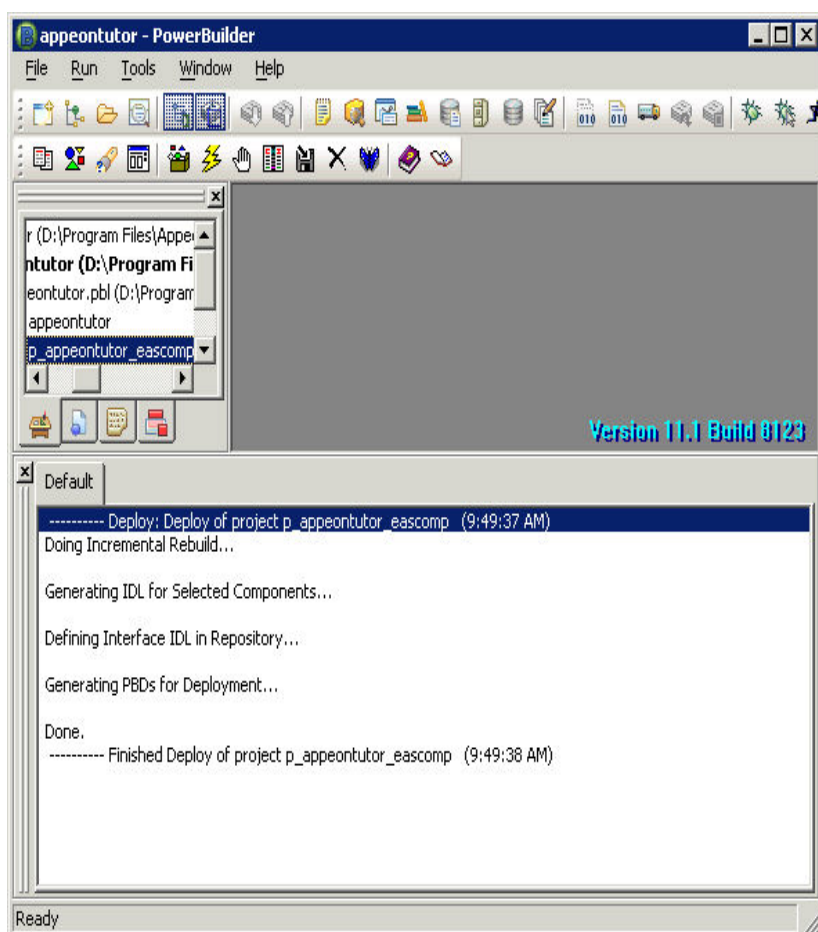
Click the **Deploy** button in PowerBuilder PainterBar1, or choose **Design | Deploy Project** from the PowerBuilder menu to start deployment of the component.

Note: Ensure that EAServer is running.

Figure 5.15:



Step 14: The deployment starts and relevant information is displayed in the Output window. When the process is complete, the Output window will display "Finished Deploy of p_appeontutor_eascomps".

Figure 5.16:

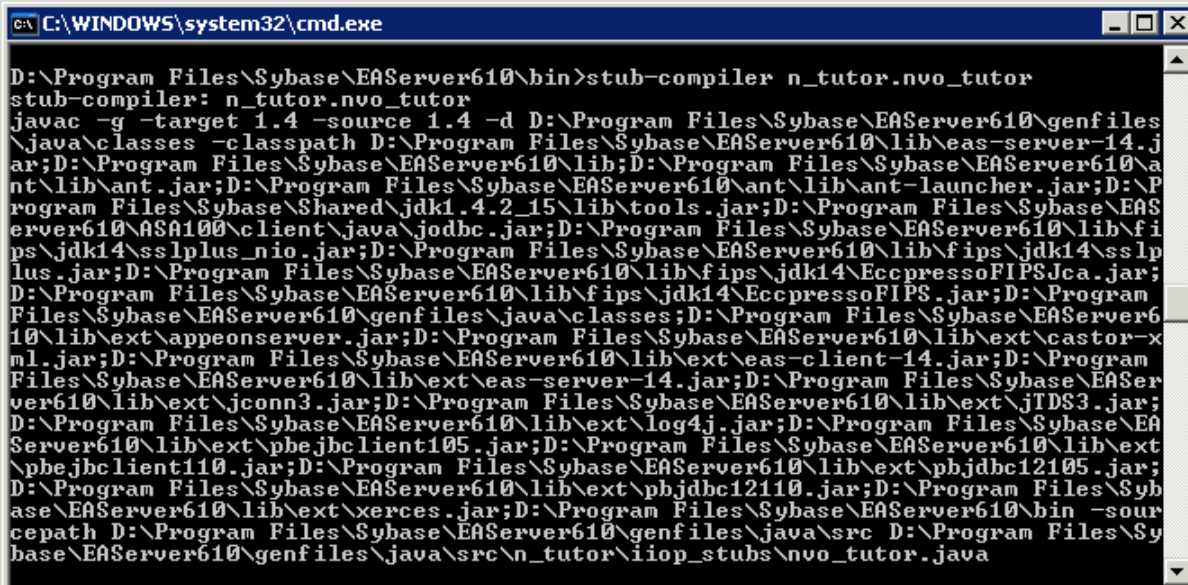
Step 15: Close the *p_appeontutor_eascomps* Project Painter.

Step 16: Check if the Stub is generated in the folder `${EAServer}\genfiles\java\classes`. If the stub is not generated manually compile the Stub by executing the following command at a prompt window:

```
${EAServer}\bin\stub-compiler package_name.component_name.
```

In the deployment mentioned in this section the package name is `n_tutor`, the component name is `nvo_tutor`.

Figure 5.17:



```

C:\WINDOWS\system32\cmd.exe
D:\Program Files\Sybase\EAServer610\bin>stub-compiler n_tutor.nvo_tutor
stub-compiler: n_tutor.nvo_tutor
javac -g -target 1.4 -source 1.4 -d D:\Program Files\Sybase\EAServer610\genfiles
\java\classes -classpath D:\Program Files\Sybase\EAServer610\lib\ea-server-14.j
ar;D:\Program Files\Sybase\EAServer610\lib;D:\Program Files\Sybase\EAServer610\an
t\lib\ant.jar;D:\Program Files\Sybase\EAServer610\ant\lib\ant-launcher.jar;D:\P
rogram Files\Sybase\Shared\jdk1.4.2_15\lib\tools.jar;D:\Program Files\Sybase\EAS
erver610\ASA100\client\java\jodbc.jar;D:\Program Files\Sybase\EAServer610\lib\fi
ps\jdk14\sslplus_nio.jar;D:\Program Files\Sybase\EAServer610\lib\fps\jdk14\sslp
lus.jar;D:\Program Files\Sybase\EAServer610\lib\fps\jdk14\EccpressoFIPSJca.jar;
D:\Program Files\Sybase\EAServer610\lib\fps\jdk14\EccpressoFIPS.jar;D:\Program
Files\Sybase\EAServer610\genfiles\java\classes;D:\Program Files\Sybase\EAServer6
10\lib\ext\appeonserver.jar;D:\Program Files\Sybase\EAServer610\lib\ext\castor-x
ml.jar;D:\Program Files\Sybase\EAServer610\lib\ext\ea-client-14.jar;D:\Program
Files\Sybase\EAServer610\lib\ext\ea-server-14.jar;D:\Program Files\Sybase\EASer
ver610\lib\ext\jconn3.jar;D:\Program Files\Sybase\EAServer610\lib\ext\jTDS3.jar;
D:\Program Files\Sybase\EAServer610\lib\ext\log4j.jar;D:\Program Files\Sybase\EA
Server610\lib\ext\pbejbcclient105.jar;D:\Program Files\Sybase\EAServer610\lib\ext
\pbejbcclient110.jar;D:\Program Files\Sybase\EAServer610\lib\ext\pbjdbc12105.jar;
D:\Program Files\Sybase\EAServer610\lib\ext\pbjdbc12110.jar;D:\Program Files\Syb
ase\EAServer610\lib\ext\xerces.jar;D:\Program Files\Sybase\EAServer610\bin -sour
cepath D:\Program Files\Sybase\EAServer610\genfiles\java\src D:\Program Files\Sy
base\EAServer610\genfiles\java\src\n_tutor\iiop_stubs\nvo_tutor.java

```

5.3 How to deploy NVOs to EAServer without PowerBuilder

Description

You can deploy NVOs to EAServer without PowerBuilder.

Workaround

Step 1: Copy the following files and sub-directories about the n-Tier NVO from %JAGUAR%\Repository\Component\PackageName.

- The component property files (*.props)
- The Component PBD files
- The component sub-directories

Step 2: Paste the files and sub-directories obtained in Step 1 to %JAGUAR%\Repository\Component\PackageName in the new EAServer.

Step 3: Go to **EAServer Manager** and generate stubs and skeletons for the components. For detailed instructions, refer to [How to deploy NVO to EAServer 6.1](#).

5.4 How to use Client resources in Apeon Web applications

Description

Apeon Web applications cannot directly interface with external resources from PowerBuilder code on the Client.

Workaround

Transfer Client resources to the Server, do all manipulations on the Server, and then return the result to the Client.

Example

The following example shows you how to use an n-tier NVO and a JSP page to work around a PowerBuilder application whose main functionality is updating a database using a text file at the Client.

The original PowerBuilder application

The main logic

Step 1: In a DataWindow control, save the specified column values as a text file on the Client.

Step 2: Modify the column values stored in the text file.

Step 3: Update the latest value stored in the text file to the database.

After modification

The main logic

Step 1: Encapsulate the following logic to an NVO.

- Logic to save the specified column values as a text file.
- Logic to update the latest value stored in the text file to the database.

Step 2: Deploy the NVO to EAServer as a Jaguar component.

Step 3: Call Jaguar component to save the column values as a text file on the Server.

Step 4: Read the contents of the text file and save it to the Client using IE browser.


Step 5: Modify the column values in the text file on the Client.

Step 6: Upload the text file to the Server using JSP.

Step 7: Use the functionality encapsulated in the Jaguar component to update the database.

5.5 How to replace Appeon image that displays at the running of applications

Description

When you launch an application in Internet Explorer, an image () shows in the browser for a moment, indicating the start of the loading process. The image is pre-defined in Appeon, but you can change it to any other image.

Workaround

Once you change the Appeon-defined image to your own image, all applications deployed to the server will show the new image at the beginning of application-loading process.

Step 1: Prepare an image that you want to display at the beginning of application-loading process. Name it to "awl_loading.gif".

Step 2: Go to the folder where the Appeon-defined image is stored, and replace the old awl_loading.gif with the new awl_loading.gif. You need to replace awl_loading.gif in the following two folders: %AppeonServer%\weblibrary_ax\debug\image\ and %AppeonServer%\weblibrary_ax\release\image\.

5.6 How to deploy an Appeon application without Appeon Developer

Description

Appeon Developer is needed to deploy the application to at least one Appeon Server and Web Server. Once deployed to an Appeon Server and a Web Server, you can replicate the deployed application to other Appeon Servers and Web Servers without using Appeon Developer.

Workaround

Use the Package Wizard provided in the Appeon Developer toolbar to generate a portable installation package for your Appeon deployed application and install the package to the other Appeon Servers and Web Servers. For step-by-step instructions on how to package and install applications, please refer to Chapter 11, *Packaging Applications* in *Appeon Developer User Guide*.

5.7 How to log in the Web application with single sign-on

Method 1: Use a server component to manage user information.

Step 1: Create an EAServer shared component (either a PowerBuilder or Java component) for storing user information. This shared component is shared between the Appeon application and other applications such as COBOL apps, Web Services, SOA, etc.

Step 2: Add the following logic to the Appeon application: when a user logs in, the user information (user ID and password) is passed to the EAServer shared component.

Step 3: When the user accesses another application (COBOL apps, Web Services, SOA, etc), the other application gets the user information from the shared component and authenticates the user.

The same method can be used for the user to first access a non-Appeon application and then access an Appeon application with single sign-on.

Method 2: Apply command line argument.

Appeon supports the CommandParm function and the command line argument in the Open event of a PowerBuilder application. These features can be applied for implementing single sign-on.

The command line argument can be passed to an Appeon application in the following way:

```
http://192.0.1.94:8080/MyTest/index.htm?user=appeon&password=appeon
```

This attaches the string “*index.htm?arguments*” to the end of the original application URL (“*index.htm*” must be included in the string).

If the user wants to launch an Appeon application after logging on to an LDAP based application, the LDAP based application passes the user information via the URL of the Appeon application, and the user starts the Appeon application without further login procedures.

It is also possible to pass the session ID only. A table is created in the database for keeping the session information of the LDAP based application, with a session ID assigned to each session, and the session information containing user information. When the Appeon

application is launched with the session ID as its command line argument, the application reads from the database table the user information and authenticates the user. The user can start the Apeon application without further login procedures.

5.8 How to add headers & footers to a Web application

Description

Use this solution to add headers and footers to a corporate website with an Apeon application in a frame.

Workaround

Create a new frame page with a header and footer. Set the URL that is used to access the deployed application to the initial page of the content frame.

Step 1: Prepare two HTML files: *header.html* and *footer.html*.

Step 2: In Microsoft FrontPage, create a new frame page that contains header, footer and content frames. To create a new frame page, click **File | New | Page**. In the *Frames Page* tab, select *Header, Footer and Contents Template*.

Step 3: Set *header.html* as the initial page of the header frame, and set *footer.html* as the initial page of the footer frame.

Step 4: Set the URL that is used to access the deployed application as the initial page of the content frame.

Step 5: Put the frame page in the same Web root as the original deployed application and provide the frame page name to the user.

Now the user can directly open the frame page which opens the deployed application. Headers and footers have already been added to the deployed application.

5.9 How to get the user name and password of the operating system

To get the user name and password of the operating system, follow the two steps:

Step 1: Package the `GetUserName` function into a DLL then call the DLL in PowerBuilder.

Step 2: Deploy the DLL to Apeon Server and download the DLL to the client needed. You can get the detail information of the `GetUserName` function from the following Website: [http://msdn.microsoft.com/en-us/library/windows/desktop/ms724432\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms724432(v=vs.85).aspx).

5.10 How to modify the storage location of Web application files in Apeon Developer

Apeon Developer will generate and store Web application files on the local machine before deploying it to Apeon Server. You can configure to store these files to other location rather than the default one. By default the Web files will be stored in `%Apeon_Developer%\Project\application_name%` (e.g., `C:\Program Files\Apeon\Developer2013R2\Project\sales`). `%Apeon_Developer%` indicates the installation path of Apeon Developer.

How to configure the location for Web files

Step 1: Open the ADTConfig.xml file in the directory %Appeon_Developer%\Developer2013R2.

Step 2: Find the following lines:

```
<PathCfg>  
<Project value="%Appeon_Developer%\Developer2013R2\Project\" />  
</PathCfg>
```

Step 3: Replace the value in bold with another location where you want to store the Web files.

5.11 How to integrate Appeon Web applications with JSP/ASP

5.11.1 Applying Appeon CommandParm and Hyperlink features

Description

If your application needs to pass parameters to a JSP/ASP application, use the following method.

Workaround for passing parameters from a JSP/ASP application to an Appeon application

Apply the JSP/ASP programming method to add parameters to the URL of the Appeon application. Based upon the parameters, any functionality can be built in the Appeon Web application, such as opening windows and retrieving data to provide client-side integration.

The Appeon application receives the parameters using the CommandParm function and CommandLine parameter of the Open event.

Workaround for passing parameters from an Appeon application to a JSP/ASP application

Appeon supports PictureHyperLink and StaticHyperLink window controls. For example, you can statically or dynamically assign the URL of the JSP/ASP Web application to *http://www.x.x/index.asp?aid=x&bid=y&cid=z* in the Clicked event of a PictureHyperLink or StaticHyperLink window control and send the parameter from the Appeon application to a JSP/ASP application.

Appeon also supports the HyperLinkToURL PowerScript function. The developers can also apply this function to pass the parameter to a JSP/ASP application through automatic code rather than being user-initiated. The Web application needs to be refreshed while receiving the parameter.

5.11.2 Using Internet Explorer Frame

Description

Appeon applications can be accessed in an Internet Explorer frame. It is possible to set the Appeon application and the ASP/JSP application in two different frames of the same browser.

Workaround

Take the following steps to build up integration using IE frames:

Step 1: Divide the IE browser into two frames; one for running the Appeon application and one for running the ASP/JSP application.

Step 2: Apply the Appeon CommandParm feature for launching the Appeon application from the ASP/JSP application.

Step 3: Apply the Appeon Hyperlink feature for launching the ASP/JSP application from the Appeon application.

5.11.3 Integration through intermediate n-Tier Server-level solutions

Description

It is possible to pass parameters between Appeon applications and JSP/ASP applications by applying server-level integration. The information that is passed can be stored at any of the tiers in the n-Tier environment, including the Client PC, Application Server, or the Database Server.

Workaround

- The information can be stored and read on the Client PC operating system through the signed and secure Appeon ActiveX.

Both Appeon and JSP/ASP application read and write a normal Client PC operating system file. The JSP/ASP application needs the ability to access the Client PC operating system file through ActiveX or a Plug-in, etc.

See the following example for how to store and read the information on the Client PC operating system DLL. Appeon calls to a Client PC operating system DLL file in the same way as in PowerBuilder using the Appeon ActiveX:

```
Var objForm
  set objForm=Server.CreateObject("Scripting.Dictionary")
  set tStream = Server.CreateObject("adodb.stream")
```

- The intermediate information can be stored in a Database Server table.

Both applications can read and write a normal RDBMS database table. Information such as orders, products, customers, or loans can be stored in a database table. After the information is stored in the database table, other applications can trigger a user-initiated event or simple automatic timer event to get updated information.

- The intermediate information can be stored in a file on the Application Server.

If both the Appeon and JSP/ASP applications call to the same DLL, the developer can make use of functionality provided by DLLs for setting up communication between the applications.

5.12 No Workarounds

Not every unsupported feature can be worked around or need to be worked around. Some features are small or trivial or may not necessarily cause functionality loss even if they are not modified or worked around.

If the unsupported feature is flagged as "Have to modify" in the Unsupported Feature Analysis report while you cannot find a workaround for it, please send a test case to <support@appeon.com> for help.

Index

A

AddItem function (EAServer only), [154](#)
 Apeon Bridge, [59](#)
 Apeon Client Functions, [17](#)
 Apeon Commit Label, [53](#)
 Apeon Commit/Rollback Label, [52](#)
 Apeon GetFullState/SetFullState/GetChanges/SetChanges (Windows only), [174](#)
 Apeon Immediate Call Label, [56](#)
 Apeon Labels, [51](#)
 Apeon Queue Labels, [54](#)
 Apeon requirements for EJB development, [67](#)
 Apeon Resize PBL Reference, [149](#)
 Apeon Rollback Label, [53](#)
 Apeon Update Label, [57](#)
 Apeon Workarounds PBL Reference, [4](#)
 Apeon Workspace, [77](#)
 ApeonDataWindow and ApeonDataStore, [48](#)
 ApeonDotNetComponent object, [69](#)
 ApeonExtFuncs object, [5](#)
 ApeonFileService object, [43](#)
 ApeonGetApeonUserName function, [17](#)
 ApeonGetBrowserVersion function, [18](#)
 ApeonGetCacheDir function, [18](#)
 ApeonGetClientID function, [19](#)
 ApeonGetClientIP function, [19](#)
 ApeonGetClientType function, [19](#)
 ApeonGetFullStateEX, [49](#)
 ApeonGetHttpInfo function, [19](#)
 ApeonGetIEHandle function, [20](#)
 ApeonGetIEURL function, [20](#)
 ApeonGetOSType function, [20](#)
 ApeonGetRemainingdays function, [21](#)
 ApeonGetServerType function, [21](#)
 ApeonGetSessionCount function, [22](#)
 apeonisin64browser function, [22](#)
 ApeonLDAPLogon function, [22](#)
 ApeonPopupMenu function, [22](#)
 ApeonPopupMenuOn function, [23](#)
 ApeonPrint2File function, [24](#)
 ApeonPrint2PDF function, [25](#)
 ApeonSetFullStateEX, [49](#)
 ApeonSwitchRealTimeCalc function, [25](#)

apeonwebservicecomponent object, [73](#)
 Application object, [155](#)
 Applying Apeon CommandParm and Hyperlink features, [198](#)

B

Barcode, [91](#)
 Best practices for calling APIs, [4](#)

C

Calling .NET/COM server components (.NET only), [68](#)
 Calling EJB Component, [59](#)
 Calling functions and events, [161](#)
 Calling Web Service, [73](#)
 Camera, [96](#)
 CanUndo function, [167](#)
 Char data type column, [173](#)
 Client Log, [26](#)
 Code Example, [66](#)
 Code Examples, [72](#), [76](#)
 Column control, [173](#)
 Configuring and deploying Apeon File Server, [26](#)
 Configuring the Apeon File Server, [31](#), [32](#), [36](#), [37](#), [39](#), [39](#), [42](#)
 Connection, [100](#)
 ConnectServer function, [61](#)
 ConnectString parameter, [174](#)
 Constructor, [71](#), [75](#)
 Controls in a DataWindow, [172](#)
 CreateRemoteInstance function, [65](#)

D

DataWindow control, [167](#)
 DataWindow Object Properties, [166](#)
 DataWindow operator precedence, [166](#)
 DataWindow operators, [166](#)
 DataWindow Reference, [166](#)
 DBCancel function, [167](#)
 DBParm parameters in Database, [174](#)
 DDE Server functions, [156](#)
 Declarations, [162](#)
 DeleteItem function, [154](#)
 Deploying NVOs to EAServer, [185](#)
 Deploying the Apeon File Server, [34](#), [36](#), [37](#), [39](#), [39](#), [43](#)
 DestroyRemoteInstance function, [66](#)
 Destructor, [71](#), [75](#)

Device, [101](#)
 DisConnection function, [61](#)
 Distributed DataWindows (EAServer only), [48](#)
 Duplicate arguments for a function, [161](#)
 DWMMessageTitle property, [155](#)

E

EJLObject object, [60](#)
 eon_cst_resize object, [149](#), [149](#)
 eon_cst_resizeattrib object, [152](#)
 eon_cst_resizecompose object, [152](#)
 eon_logex, [26](#)
 eon_mobilecameraex object, [96](#)
 eon_mobile_awsex object, [77](#)
 eon_mobile_barcodeex object, [91](#)
 eon_mobile_connectionex object, [101](#)
 eon_mobile_deviceex object, [101](#)
 eon_mobile_mapex object, [116](#)
 eon_mobile_mediaex object, [108](#), [125](#), [131](#)
 eon_mobile_nfcex object, [137](#)
 eon_mobile_notificationex object, [126](#)
 eon_mobile_str_annotation, [125](#)
 eon_mobile_str_cameraoption, [100](#)
 eon_mobile_str_coordinates, [113](#)
 eon_mobile_str_mailcontent, [90](#)
 eon_mobile_str_mapoption, [124](#)
 eon_mobile_str_nfcrecord, [147](#)
 eon_mobile_str_textcheckoption, [136](#)
 eon_offlineex object, [128](#)
 Evaluating DataWindow expressions in scripts, [173](#)
 Events for ApeonDotNetComponent, [71](#)
 Events for apeonwebservicecomponent, [75](#)
 Events for DataWindow control, [170](#)
 External user object, [154](#)

F

FAQ & Workarounds, [182](#)
 File Upload and Download, [26](#)
 FileLength64 function, [156](#)
 FileReadEx function, [157](#)
 FileSeek64 function, [157](#)
 FileWriteEx function, [157](#)
 FreeDBLibraries property, [155](#)
 FromAnsi function, [157](#)
 FromUnicode function, [157](#)
 Functions for ApeonDotNetComponent, [70](#)
 Functions for ApeonFileService object, [44](#)

Functions for apeonwebservicecomponent, [74](#)
 Functions for DataWindow control, [167](#)
 Functions of DataStore, [174](#)

G

Garbage Collection functions, [158](#)
 General API, [4](#)
 GenerateResultSet function, [167](#)
 Geolocation, [108](#)
 GetItemAtPointer function, [155](#)
 GetLibraryList function, [158](#)
 GetTrans function, [168](#)
 GOTO statement, [164](#)

H

Help event, [165](#)
 Help functions, [158](#)
 How to add headers & footers to a Web application, [197](#)
 How to deploy an Apeon application without Apeon Developer, [196](#)
 How to deploy NVO to EAServer 6.1, [183](#)
 How to deploy NVOs to EAServer without PowerBuilder, [194](#)
 How to get the user name and password of the operating system, [197](#)
 How to integrate Apeon Web applications with JSP/ASP, [198](#)
 How to log in the Web application with single sign-on, [196](#)
 How to modify the storage location of Web application files in Apeon Developer, [197](#)
 How to remove the Internet Explorer menu, [182](#)
 How to replace Apeon image that displays at the running of applications, [195](#)
 How to use Client resources in Apeon Web applications, [194](#)
 HyperLinkToURL function, [159](#)

I

il_Id, [48](#)
 InitLocalLanguage function, [62](#)
 Installing the Apeon File Server, [27](#)
 Integration through intermediate n-Tier Server-level solutions, [199](#)
 Introduction to Apeon Workarounds, [4](#)
 Invoking component functions, [63](#)

is_FileServerName, [47](#)
 is_LogOnParams, [48](#)
 ItemActivate event, [156](#)

L

Language basics, [163](#)
 Large Binary/Text database OLE object, [172](#)
 LibraryDirectory function, [159](#)
 ListView control, [155](#)
 LiveScroll property, [172](#)
 LookUpJndi function, [62](#)

M

Map, [115](#)
 Media, [125](#)
 Mobile Device API, [77](#)

N

NFC, [136](#)
 Notification, [126](#)
 Null values, [163](#)

O

Objects & Controls, [154](#)
 oe_annotationclicked, [116](#)
 oe_changed, [117](#)
 oe_error, [109](#), [117](#)
 oe_loadend, [117](#)
 oe_loadstart, [117](#)
 oe_locationchanged, [110](#)
 oe_newtagfound, [137](#)
 Offline, [127](#)
 of_addannotation, [118](#)
 of_addressmessage, [126](#)
 of_addrrecord, [138](#)
 of_adresstocoordinate, [119](#)
 of_appeondownload, [46](#)
 of_appeonupload, [45](#)
 of_applydbupdate, [128](#)
 of_applyupdate, [129](#)
 of_callwebservice, [74](#)
 of_checkappeonserver, [128](#)
 of_checkSizeType, [149](#)
 of_checkupdate, [130](#)
 of_clearlog, [77](#)
 of_clearlog function, [14](#)
 of_close, [110](#), [119](#), [138](#)
 of_completionsforpartialword, [131](#)
 of_coordinatetoaddress, [120](#)

of_create, [91](#)
 of_downloadfile, [47](#)
 of_execinterface, [70](#)
 of_FileExists, [47](#)
 of_getappeonusername function, [7](#)
 of_getappinfo, [84](#)
 of_getapporientation, [80](#)
 of_getapprotationlock, [82](#)
 of_getassistivetouchbtnvisible, [79](#)
 of_getassistivetouchmode, [78](#)
 of_getblocksize, [139](#)
 of_getbrowserversion function, [7](#)
 of_getcachedir function, [8](#)
 of_getclientid function, [8](#)
 of_getclientip function, [9](#)
 of_getclienttype function, [9](#)
 of_getcloseappiconvisible, [88](#)
 of_getComponentName, [149](#)
 of_getconnectioninfo, [101](#)
 of_getcurrentposition, [110](#)
 of_getdeviceid, [101](#)
 of_getdevicetype, [102](#)
 of_getdpi, [103](#)
 of_getdwmousemovemode, [89](#)
 of_getfiledata, [94](#), [96](#)
 of_getfreememory, [104](#)
 of_gethttpinfo function, [9](#)
 of_getiehandle function, [10](#)
 of_getieurl function, [10](#)
 of_getignoredwords, [132](#)
 of_getlanguages, [133](#)
 of_getmaxblockcount, [139](#)
 of_getmaxsectorcount, [139](#)
 of_getmisspelledword, [133](#)
 of_getname, [104](#)
 of_getndefrecord, [140](#)
 of_getndefrecordcount, [140](#)
 of_getorientation, [105](#)
 of_getostype function, [10](#)
 of_getosversion, [106](#)
 of_getplatform, [105](#)
 of_getppi, [103](#)
 of_getresolution, [106](#)
 of_getservertype function, [11](#)
 of_getsessioncount function, [12](#)
 of_getsize, [141](#)
 of_getstatusbarvisible, [107](#)
 of_gettagid, [141](#)

[of_gettimeout](#), [142](#)
[of_gettitlebarvisible](#), [83](#)
[of_gettotalmemory](#), [103](#)
[of_getversion](#), [84](#)
[of_getwindowlisticonvisible](#), [87](#)
[of_guessesforward](#), [134](#)
[of_haslearnedword](#), [134](#)
[of_init](#), [149](#), [152](#)
[of_init \(userobject\)](#), [150](#)
[of_init \(window\)](#), [150](#)
[of_is64browser](#) function, [12](#)
[of_isenabled](#), [111](#)
[of_ldaplogon](#) function, [12](#)
[of_learnword](#), [135](#)
[of_log](#), [78](#)
[of_log](#) function, [13](#)
[of_logofffileservers](#), [45](#)
[of_logonfileservers](#), [44](#)
[of_open](#), [111](#), [121](#), [142](#)
[of_openalbums](#), [97](#)
[of_play](#), [125](#)
[of_popmenu](#) function, [14](#)
[of_popmenuon](#) function, [15](#)
[of_print2file](#) function, [15](#)
[of_Print2PDF](#), [17](#)
[of_read](#), [94](#)
[of_readblock](#), [143](#)
[of_refresh](#), [121](#)
[of_register](#), [112](#), [143](#), [150](#)
[of_register \(a_component, a_obj, as_method\)](#), [152](#)
[of_register \(a_obj, as_method\)](#), [152](#)
[of_removeallannotation](#), [123](#)
[of_removeannotation](#), [122](#)
[of_removemessage](#), [127](#)
[of_runmode](#) function, [11](#)
[of_selectannotation](#), [123](#)
[of_sendmail](#), [85](#)
[of_setapporientation](#), [81](#)
[of_setapprotationlock](#), [82](#)
[of_setassistivetouchbtnvisible](#), [80](#)
[of_setassistivetouchmode](#), [79](#)
[of_setcloseappiconvisible](#), [88](#)
[of_setdwmousemovemode](#), [90](#)
[of_setignoredwords](#), [132](#)
[of_setkey](#), [144](#)
[of_setResizeType](#), [151](#)
[of_setstatusbarvisible](#), [107](#)

[of_settimeout](#), [144](#)
[of_settitlebarvisible](#), [83](#)
[of_setwindowlisticonvisible](#), [87](#)
[of_startblockoperate](#), [145](#)
[of_stopblockoperate](#), [145](#)
[of_switchRealTimeCalc](#) function, [17](#)
[of_takefile](#), [98](#)
[of_triggerevent](#), [146](#)
[of_typeof](#), [151](#)
[of_unlearnword](#), [135](#)
[of_uploadfile](#), [47](#)
[of_writeblock](#), [146](#)
[of_writerecord](#), [147](#)
Other event, [166](#)
Overriding system function, [161](#)

P

Packaging unsupported features in n-Tier NVOs or DLLs, [177](#)
Packaging unsupported objects into n-Tier NVOs, [177](#)
Passing arguments to functions and events, [161](#)
Passing Menu object as a reference parameter, [162](#)
[pfc_resize](#), [153](#)
PipeLine object, [154](#)
Placing unsupported functions into n-Tier NVOs, [179](#)
[PopulateError](#) function, [160](#)
PowerScript Events, [165](#)
PowerScript Reference, [161](#)
PowerScript Statements, [164](#)
PowerScript Topics, [161](#)
Properties for [ApeonDotNetComponent](#), [69](#)
Properties for [ApeonFileService](#) object, [47](#)
Properties for [apeonwebservicecomponent](#), [73](#)
Properties of the DataWindow Control, [172](#)

R

Reference parameter, [162](#)
Registering parameter functions, [64](#)
[Registry](#) function, [160](#)
[RegStruct](#) and [RegStructArray](#) functions, [64](#)
[ReselectRow](#) function, [168](#)
[ResetTransObject](#) function, [168](#)
[Retrieve.AsNeeded](#), [166](#)
[RetrieveRow](#) event, [171](#)

S

SaveAsAscii function, [169](#)
Scroll function, [169](#)
ScrollVertical event, [170](#)
SetDropHighlight function, [156](#)
SetTrans function, [169](#)
Shared Object functions, [160](#)
Shared variables, [162](#)
SignalError function, [160](#)
SQL Statements, [165](#)
Stored procedure with Null output, [165](#)
System Controls, [155](#)
System Functions, [156](#)
System Objects, [154](#)

T

Tab control, [156](#)
TabPostEvent function, [156](#)
Textchecker, [131](#)
TreeView control, [156](#)

U

Uploading and downloading files, [43](#)
UserObject object, [154](#)
Using Internet Explorer Frame, [198](#)

W

Workaround limitations, [50](#)
Workaround steps, [50](#)
Workaround Techniques (for EA Server only), [174](#)
Workarounds for Unsupported Features, [154](#)