

# **Workarounds & APIs Guide**

**Appeon® PowerServer® 2020**  
**FOR WINDOWS & UNIX & LINUX**

DOCUMENT ID: ADC20236-01-2020-01

LAST REVISED: March 25, 2020

Copyright © 2020 Appeon. 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 Inc.

Appeon, the Appeon logo, Appeon PowerBuilder, Appeon PowerServer, PowerServer, PowerServer Toolkit, AEM, and PowerServer Web Component are trademarks of Appeon Inc.

SAP, Sybase, Adaptive Server Anywhere, SQL Anywhere, Adaptive Server Enterprise, iAnywhere, Sybase Central, and Sybase jConnect for JDBC are trademarks or registered trademarks of SAP and SAP affiliate company.

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

# Contents

1 Appeon Workarounds PBL Reference .....	1
1.1 Introduction to Appeon Workarounds .....	1
1.2 Best practices for calling APIs .....	1
1.3 General API .....	1
1.3.1 AppeonExtFuncs Object .....	2
1.3.1.1 of_clearlog function .....	4
1.3.1.2 of_getappeonusername function .....	4
1.3.1.3 of_getbrowserversion function .....	4
1.3.1.4 of_getcachedir function .....	5
1.3.1.5 of_getclientid function .....	5
1.3.1.6 of_getclientip function .....	6
1.3.1.7 of_getclienttype function .....	6
1.3.1.8 of_gethttpinfo function .....	6
1.3.1.9 of_getiehandle function .....	7
1.3.1.10 of_getieurl function .....	7
1.3.1.11 of_getostype function .....	8
1.3.1.12 of_getrunmode function .....	8
1.3.1.13 of_getservertype function .....	8
1.3.1.14 of_getsessioncount function .....	9
1.3.1.15 of_is64browser function .....	9
1.3.1.16 of_Idaplogon function .....	10
1.3.1.17 of_log function .....	10
1.3.1.18 of_networktest function .....	11
1.3.1.19 of_popupmenu function .....	11
1.3.1.20 of_popupmenuon function .....	12
1.3.1.21 of_print2file function .....	13
1.3.1.22 of_Print2PDF function (Obsolete) .....	14
1.3.1.23 of_switchRealTimeCalc function .....	14
1.3.2 Appeon Client Functions .....	15
1.3.2.1 AppeonGetAppeonUserName function .....	15
1.3.2.2 AppeonGetBrowserVersion function .....	15
1.3.2.3 AppeonGetCacheDir function .....	15
1.3.2.4 AppeonGetClientID function .....	16
1.3.2.5 AppeonGetClientIP function .....	16
1.3.2.6 AppeonGetClientType function .....	16
1.3.2.7 AppeonGetClientVersion function .....	17
1.3.2.8 AppeonGetHttpInfo function .....	17
1.3.2.9 AppeonGetIEHandle function .....	17
1.3.2.10 AppeonGetIEURL function .....	18
1.3.2.11 AppeonGetOSType function .....	18
1.3.2.12 AppeonGetRemainingdays function .....	18
1.3.2.13 AppeonGetServerType function .....	19
1.3.2.14 AppeonGetSessionCount function .....	19
1.3.2.15 appeonis64browser function .....	19
1.3.2.16 AppeonLDAPLogon function .....	20
1.3.2.17 AppeonNetworkTest function .....	20

1.3.2.18	AppeonPopupMenu function .....	21
1.3.2.19	AppeonPopupMenuOn function .....	21
1.3.2.20	AppeonPrint2File function .....	22
1.3.2.21	AppeonPrint2PDF function (Obsolete) .....	23
1.3.2.22	AppeonSetHintText function .....	23
1.3.2.23	AppeonSetLongPressInterval function .....	24
1.3.2.24	AppeonSwitchRealTimeCalc function .....	24
1.3.3	Client Logs .....	25
1.3.3.1	eon_logex object .....	25
1.3.4	File Upload and Download .....	25
1.3.4.1	Configuring and deploying Appeon File Server .....	25
1.3.4.2	Uploading and downloading files .....	37
1.3.5	Appeon Labels .....	43
1.3.5.1	Appeon Commit/Rollback Label .....	44
1.3.5.2	Appeon Commit Label .....	45
1.3.5.3	Appeon Rollback Label .....	46
1.3.5.4	Appeon Queue Labels .....	47
1.3.5.5	Appeon Immediate Call Label .....	49
1.3.5.6	Appeon Update Label .....	50
1.3.6	Calling EJB Component .....	52
1.3.6.1	Supported application server .....	52
1.3.6.2	Generate the EJB proxy .....	52
1.3.6.3	Bundle EJB proxy with Appeon Bridge .....	53
1.3.6.4	Deploy to application server .....	54
1.3.6.5	Call EJB component in PowerBuilder application .....	54
1.3.7	Calling .NET/COM server components (.NET only) .....	64
1.3.7.1	AppeonDotNetComponent object .....	65
1.3.8	Calling Web Service .....	68
1.3.8.1	appeonwebservicecomponent object .....	69
1.3.9	Appeon Text Controls .....	73
1.3.9.1	eon_singlelineedit, eon_statictext, & eon_editmask .....	73
1.3.10	Creating Data Source Dynamically .....	74
1.3.10.1	Overview .....	74
1.3.10.2	appeondconfigdatasource .....	74
1.3.11	Accessing JSON-format Data .....	76
1.3.11.1	Overview .....	76
1.3.11.2	eon_cjsonnode .....	77
1.3.11.3	eon_cjsonnodearray .....	83
1.3.12	Appeon Resize Object .....	89
1.3.12.1	eon_appeon_resize object .....	89
1.4	Mobile Device API .....	99
1.4.1	Audio .....	99
1.4.1.1	eon_mobile_audioex .....	99
1.4.2	Appeon Workspace .....	102
1.4.2.1	eon_mobile_awsex object .....	103
1.4.3	Barcode .....	123
1.4.3.1	eon_mobile_barcodeex object .....	123
1.4.4	Camera .....	127

1.4.4.1 eon_mobile_cameraex object .....	127
1.4.5 Connection .....	132
1.4.5.1 eon_mobile_connectionex object .....	132
1.4.6 Device .....	133
1.4.6.1 eon_mobile_deviceex object .....	133
1.4.7 GPS .....	140
1.4.7.1 eon_mobile_geolocationex object .....	140
1.4.8 Image .....	149
1.4.8.1 eon_mobile_imageex object .....	149
1.4.9 Map .....	151
1.4.9.1 eon_mobile_mapex object .....	152
1.4.10 Media .....	162
1.4.10.1 eon_mobile_mediaex object .....	162
1.4.11 NFC .....	162
1.4.11.1 eon_mobile_nfceex object .....	163
1.4.12 Notification .....	178
1.4.12.1 eon_mobile_notificationex object .....	178
1.4.13 Offline .....	180
1.4.13.1 eon_offlineex object .....	180
1.4.14 Online Payment .....	184
1.4.14.1 eon_mobile_paymentex object .....	184
1.4.15 Textchecker .....	190
1.4.15.1 eon_mobile_textcheckex object .....	190
1.4.16 WebView .....	197
1.4.16.1 eon_mobile_webviewex object .....	197
2 Appeon Resize PBL Reference .....	206
3 Integrate with Cordova Plugins .....	207
3.1 Overview .....	207
3.2 Appeon Cordova PBL (solution 1) .....	207
3.2.1 Introduction to Appeon Cordova PBL .....	207
3.2.2 eon_cordova_base object .....	208
3.2.2.1 Functions .....	208
3.2.2.2 Events .....	210
3.2.2.3 Properties .....	211
3.2.3 eon_cordova_bluetooth object (Android only) .....	212
3.2.3.1 Functions .....	212
3.2.3.2 Events .....	214
3.2.3.3 Properties .....	214
3.2.4 eon_cordova_bluetoothle object .....	215
3.2.4.1 Functions .....	215
3.2.4.2 Events .....	221
3.2.4.3 Properties .....	222
3.2.4.4 Structures .....	223
3.2.5 eon_cordova_contact object .....	223
3.2.5.1 Functions .....	223
3.2.5.2 Properties .....	235
3.2.5.3 Structures .....	236
3.2.6 eon_cordova_fingerprint object .....	239

3.2.6.1 Functions .....	239
3.2.6.2 Events .....	240
3.2.6.3 Properties .....	241
3.2.7 eon_cordova_gps object .....	241
3.2.7.1 Functions .....	241
3.2.7.2 Events .....	242
3.2.7.3 Properties .....	243
3.2.7.4 Structures .....	243
3.2.8 eon_cordova_orientation object .....	244
3.2.8.1 Functions .....	244
3.2.8.2 Events .....	246
3.2.8.3 Properties .....	246
3.2.8.4 Structures .....	246
3.2.9 eon_cordova_vibration object .....	247
3.2.9.1 Functions .....	247
3.2.9.2 Properties .....	248
3.3 Cordova plugins (solution 2) .....	248
3.3.1 Connecting with AppeonMobile.CordovaPlugin .....	249
3.3.2 AppeonMobile.CordovaPlugin object .....	250
3.3.2.1 SetTimeout function .....	250
3.3.2.2 AssociateJSwithPB function .....	250
3.3.2.3 ExecJavaScript function .....	251
3.3.2.4 ExecJavaScriptWithReturn function .....	252
3.3.2.5 ExecJavaScriptGetLastFuncCallRetVal function .....	252
3.3.2.6 GetLastErrorExecJS function .....	253
3.3.2.7 oe_execjserror event .....	254
3.3.3 Accessing the JavaScript object properties/methods .....	254
3.3.3.1 Using dot notation .....	254
3.3.3.2 Using a JSON string .....	255
3.3.3.3 Using ExecJavaScript/ExecJavaScriptWithReturn function .....	255
3.3.3.4 Using eon_cjsonnode/eon_cjsonnodearray object .....	255
3.3.3.5 Code example .....	256
3.3.4 Accessing the JavaScript array .....	257
3.3.5 Using callback functions .....	258
3.3.6 Code examples .....	260
4 PowerServer open interfaces .....	263
4.1 PowerServer open interfaces .....	263
4.1.1 Overview .....	263
4.1.2 getAllClients .....	264
4.1.2.1 Syntax for .NET component .....	264
4.1.2.2 Syntax for EJB component .....	265
4.1.2.3 Syntax for Web Service .....	266
4.1.3 getAllSessions .....	266
4.1.3.1 Syntax for .NET component .....	266
4.1.3.2 Syntax for EJB component .....	268
4.1.3.3 Syntax for Web Service .....	270
4.1.4 getSessionByID .....	271

4.1.4.1 Syntax for .NET component .....	271
4.1.4.2 Syntax for EJB component .....	273
4.1.4.3 Syntax for Web Service .....	274
4.1.5 getSessionCount .....	275
4.1.5.1 Syntax for .NET component .....	275
4.1.5.2 Syntax for EJB component .....	276
4.1.5.3 Syntax for Web Service .....	277
4.1.6 getServerVersion .....	278
4.1.6.1 Syntax for .NET component .....	278
4.1.6.2 Syntax for EJB component .....	279
4.1.6.3 Syntax for Web Service .....	279
4.1.7 killAllSessions .....	279
4.1.7.1 Syntax for .NET component .....	280
4.1.7.2 Syntax for EJB component .....	280
4.1.7.3 Syntax for Web Service .....	281
4.1.8 killSessions .....	282
4.1.8.1 Syntax for .NET component .....	282
4.1.8.2 Syntax for EJB component .....	282
4.1.8.3 Syntax for Web Service .....	283
4.1.9 rollbackAllTransactions .....	283
4.1.9.1 Syntax for .NET component .....	284
4.1.9.2 Syntax for EJB component .....	284
4.2 Calling PowerServer open interfaces via EJB component .....	285
4.3 Calling PowerServer open interfaces via .NET component .....	286
4.4 Calling PowerServer open interfaces via Web service .....	287
<b>5 Workarounds for Unsupported Features .....</b>	<b>289</b>
<b>5.1 Objects &amp; Controls .....</b>	<b>289</b>
5.1.1 External user object .....	289
5.1.2 UserObject object .....	289
5.1.2.1 DeleteItem function .....	289
5.1.3 System Objects .....	289
5.1.3.1 Application object .....	289
5.1.4 System Controls .....	290
5.1.4.1 ListView control .....	290
5.1.4.2 TreeView control .....	290
5.1.4.3 Tab control .....	290
<b>5.2 System Functions .....</b>	<b>291</b>
5.2.1 DDE Server functions .....	291
5.2.2 FileLength64 function .....	291
5.2.3 FileSeek64 function .....	291
5.2.4 FileWriteEx function .....	291
5.2.5 FromAnsi function .....	291
5.2.6 FromUnicode function .....	292
5.2.7 Garbage Collection functions .....	292
5.2.8 GetLibraryList function .....	292
5.2.9 Help functions .....	292
5.2.10 HyperLinkToURL function .....	293
5.2.11 LibraryDirectory function .....	293

5.2.12 PopulateError function .....	294
5.2.13 Registry function .....	294
5.2.14 Shared Object functions .....	294
5.2.15 SignalError function .....	295
5.3 PowerScript Reference .....	295
5.3.1 PowerScript Topics .....	295
5.3.1.1 Calling functions and events .....	295
5.3.1.2 Declarations .....	296
5.3.1.3 Language basics .....	297
5.3.2 PowerScript Statements .....	298
5.3.2.1 GOTO statement .....	298
5.3.3 SQL Statements .....	299
5.3.3.1 Stored procedure with Null output .....	299
5.3.4 PowerScript Events .....	299
5.3.4.1 Help event .....	299
5.3.4.2 Other event .....	299
5.4 DataWindow Reference .....	300
5.4.1 DataWindow operators .....	300
5.4.1.1 DataWindow operator precedence .....	300
5.4.2 DataWindow Object Properties .....	300
5.4.2.1 Retrieve.AsNeeded .....	300
5.4.3 DataWindow control .....	301
5.4.3.1 Functions .....	301
5.4.3.2 Events .....	303
5.4.3.3 Properties .....	305
5.4.4 Controls in a DataWindow .....	306
5.4.4.1 Large Binary/Text database OLE object .....	306
5.4.4.2 Column control .....	306
5.4.5 Functions of DataStore object .....	307
5.5 DBParm parameters in Database .....	308
5.5.1 ConnectString parameter .....	308
6 FAQ & Workarounds .....	309
6.1 What files will go to the plugin folder and how to access them .....	309
6.2 How to generate a PDF file and send it via email .....	310
6.3 How to make a phone call in the mobile application .....	311
6.4 How to open Google Map in the mobile application .....	311
6.5 How to download an image to the mobile device and show it in the application .....	313
6.6 How to replace the icon that shows in the IE tab or address bar .....	313
6.7 How to replace Appeon image that displays at the running of applications .....	314
6.8 How to remove the Internet Explorer menu .....	314
6.9 How to deploy a PowerServer application without PowerServer Toolkit .....	315
6.10 How to log in the PowerServer Web application with single sign-on .....	316
6.11 How to add headers & footers to a Web application .....	316
6.12 How to get the user name and password of the operating system .....	317

6.13 How to modify the storage location of Web application files in PowerServer Toolkit .....	317
6.14 How to integrate PowerServer Web applications with JSP/ASP .....	318
6.14.1 Applying Appeon CommandParm and Hyperlink features .....	318
6.14.2 Using Internet Explorer Frame .....	318
6.14.3 Integration through intermediate n-Tier Server-level solutions .....	318
6.15 No Workarounds .....	319
Index .....	320



# 1 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 the PowerServer Web application and the PowerServer mobile application.

## 1.1 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`
- One DLL: `Eonejbclient.dll`
- XML file: `ejb_err_info.xml`

They are located in the `developTempFile\appeon_workarounds` folder under the PowerServer installation directory. For example, `C:\inetpub\wwwroot\appeon\developTempFile\appeon_workarounds`.

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.

## 1.2 Best practices for calling APIs

Not all of the APIs can be used in both the Mobile and Web environment. In [Section 1.3, “General API”](#), the following functions: `of_popupmenu`, `of_popupmenuon`, `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 1.4, “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
```

## 1.3 General API

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

### 1.3.1 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 two 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 PowerServer NVO components. Then, in the PowerServer 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 1.1: Functions of AppeonExtFuncs object**

Function of AppeonExtFuncs object	Appeon client functions	Description
<a href="#">of_getappeonusername</a> <a href="#">function</a>	<a href="#">AppeonGetAppeonUser</a> <a href="#">Name</a> <a href="#">function</a>	Gets the user name that you type into the Appeon Login dialog box.
<a href="#">of_getbrowserversion</a> <a href="#">function</a>	<a href="#">AppeonGetBrowserVersion</a> <a href="#">function</a>	Gets the Internet Explorer version of the client.
<a href="#">of_getcachedir</a> <a href="#">function</a>	<a href="#">AppeonGetCacheDir</a> <a href="#">function</a>	Gets the Cache directory that is used by the current application.
<a href="#">of_getclientid</a> <a href="#">function</a>	<a href="#">AppeonGetClientID</a> <a href="#">function</a>	Gets the unique session identifier for the Web client.
<a href="#">of_getclientip</a> <a href="#">function</a>	<a href="#">AppeonGetClientIP</a> <a href="#">function</a>	Gets the IP address of the Web client.
<a href="#">of_getclienttype</a> <a href="#">function</a>	<a href="#">AppeonGetClientType</a> <a href="#">function</a>	Gets the type of an application.
<a href="#">of_gethttpinfo</a> <a href="#">function</a>	<a href="#">AppeonGetHttpInfo</a> <a href="#">function</a>	Gets the HTTP header information from a particular request.
<a href="#">of_getiehandle</a> <a href="#">function</a>	<a href="#">AppeonGetIEHandle</a> <a href="#">function</a>	Gets the Internet Explorer handle for the current application.

<b>Function of AppeonExtFuncs object</b>	<b>Appeon client functions</b>	<b>Description</b>
<a href="#"><u>of_getieurl function</u></a>	<a href="#"><u>AppeonGetIEURL function</u></a>	Gets the URL of the application.
<a href="#"><u>of_getostype function</u></a>	<a href="#"><u>AppeonGetOSType function</u></a>	Gets the type of OS that runs your application (the PowerServer Web application, the PowerServer mobile application, or the PowerBuilder client application).
NA	<a href="#"><u>AppeonGetRemainingday function</u></a>	Gets the remaining day(s) of license or technical support.
<a href="#"><u>of_getservertype function</u></a>	<a href="#"><u>AppeonGetServerType function</u></a>	Gets the PowerServer type where the application runs.
<a href="#"><u>of_getsessioncount function</u> (Obsolete)</a>	<a href="#"><u>AppeonGetSessionCount function</u> (Obsolete)</a>	This function is obsolete since Appeon PowerServer 2013.  Replace it with the <a href="#"><u>getSessionCount</u></a> function in PowerServer open interfaces.
<a href="#"><u>of_is64browser function</u></a>	<a href="#"><u>appeonisin64browser function</u></a>	Detects if the IE browser where the application runs is 64-bit.
<a href="#"><u>of_ldaplogon function</u></a>	<a href="#"><u>AppeonLDAPLogon function</u></a>	Logs in to the LDAP server with the specified user name and password.
<a href="#"><u>of_networktest function</u></a>	<a href="#"><u>AppeonNetworkTest function</u></a>	Calculates the elapsed time for the client to call the server over the network.
<a href="#"><u>of_popupmenu function</u></a>	<a href="#"><u>AppeonPopupMenu function</u></a>	Pops up Appeon DataWindow menu at a specified position in a specified DataWindow control.
<a href="#"><u>of_popupmenuon function</u></a>	<a href="#"><u>AppeonPopupMenuOn function</u></a>	Pops up Appeon DataWindow menu in a specified window when you right click the mouse button.
<a href="#"><u>of_print2file function</u></a>	<a href="#"><u>AppeonPrint2File function</u></a>	Saves the specified DataWindow as image files of BMP, JPG or GIF format.
<a href="#"><u>of_Print2PDF function</u> (Obsolete)</a>	<a href="#"><u>AppeonPrint2PDF function</u> (Obsolete)</a>	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.
NA	<a href="#"><u>AppeonSetHintText function</u></a>	Sets the HintText property of the SingleLineEdit control. The HintText property is not available in PowerBuilder SingleLineEdit control. It is specially designed for the PowerServer mobile application.

<b>Function of AppeonExtFuncs object</b>	<b>Appeon client functions</b>	<b>Description</b>
<a href="#">of_switchRealTimeCalculation function</a>	<a href="#">AppeonSwitchRealTimeCalculation function</a>	Performs the DataWindow real-time expression calculation in time or performs the calculation for only one time in the whole life-cycle.

### 1.3.1.1 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()
```

<b>Argument</b>	<b>Description</b>
<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.

### 1.3.1.2 of\_getappeonusername function

#### Description

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

#### Syntax

```
appeonextfuncs.of_getappeonusername()
```

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

#### Return value

String.

#### Usage

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

### 1.3.1.3 of\_getbrowserversion function

#### Description

Gets the Internet Explorer version of the Web client.

Supported in the browser-based Web application only.

## Syntax

```
appeonextfuncs.of_getbrowserversversion()
```

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

## Return value

String.

## Usage

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

### 1.3.1.4 of\_getcachedir function

#### Description

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

For the Web application, the cache directory can be customized in AEM. For details, refer to Section 4.4.6.7, “Client Storage Location” in *PowerServer Configuration Guide for .NET* or in PowerServer Configuration Guide for J2EE.

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

## Return value

String.

## Usage

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

### 1.3.1.5 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.

#### 1.3.1.6 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.

#### 1.3.1.7 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 in the Web browser or on the desktop as an IWA (installable Web app).  Returns "PB" if the application runs in PowerBuilder.  Returns "MOBILE" if the application runs on a mobile device.
---------	--

#### 1.3.1.8 of\_gethttpinfo function

### Description

Gets the HTTP header information from a particular request.

## Syntax

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

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

## Return value

String.

## Usage

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

### 1.3.1.9 of\_getiehandle function

#### Description

Gets the Internet Explorer handle for the Web application.

Supported in the browser-based Web application only.

## Syntax

```
appeonextfuncs.of_getiehandle()
```

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

## Return value

Long.

## Usage

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

### 1.3.1.10 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

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

**Return value**

String.

**Usage**

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

**1.3.1.11 of\_getostype function****Description**

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

**Syntax**

```
appeonextfuncs.of_getostype()
```

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

**Return value**

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

**1.3.1.12 of\_getrunmode function****Description**

Gets the running mode of an application.

**Syntax**

```
appeonextfunc.of_getrunmode()
```

Argument	Description
<code>appeonextfunc</code>	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 PowerServer Toolkit, see Section 3.2.1.2.1, “Basic Settings” in <i>PowerServer Toolkit User Guide</i> ) if the application runs on the Web or a mobile device.
--------	---

**1.3.1.13 of\_getservertype function****Description**

Gets the PowerServer type where the application runs.

## Syntax

`appeonextfuncs.of_getservertype()`

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

## Return value

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

## Usage

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

### 1.3.1.14 of\_getsessioncount function

This function is obsolete since Appeon PowerServer 2013. Replace it with the [getSessionCount](#) function in PowerServer open interfaces.

### 1.3.1.15 of\_is64browser function

#### Description

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

Supported in the browser-based Web application only.

#### Syntax

`appeonextfuncs.of_is64browser()`

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

## Return value

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

## Usage

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

### 1.3.1.16 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
<code>appeonextfuncs</code>	A reference to an AppeonExtFuncs object.
<code>as_username</code>	User name for logging into the LDAP server.
<code>as_password</code>	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 PowerServer application, not in the PowerBuilder application. To make this function work in the PowerServer 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 4.3.5.5, “System Security” in *PowerServer Configuration Guide for .NET* or PowerServer Configuration Guide for J2EE.

### 1.3.1.17 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
<code>appeonextfuncs</code>	A reference to an AppeonExtFuncs object.
<code>ai_level</code>	1 - FATAL level. 2 - ERROR level. 3 - WARNING level. 4 - INFO level. (Default value) 5 - DEBUG level.

<code>as_info</code>	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 4.4.6.8, “Client Logs” in *PowerServer Configuration Guide for .NET* or PowerServer Configuration Guide for J2EE.

**1.3.1.18 of\_networktest function****Description**

Calculates the elapsed time for the client to call the server over the network. The returned value can help the developer find out the network state, and then determine whether to do data retrieval or update, especially for a large amount of data. This improves the end user experience, to avoid hanging up the application for a long time when data retrieval or update is performed on a poor network connection.

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

This function works with the .NET IIS server only, not with the J2EE application server.

**Syntax**

```
appeonextfuncs.of_networktest ( long lcount )
```

Argument	Description
<code>appeonextfuncs</code>	A reference to an <code>AppeonExtFuncs</code> object.
<code>lcount</code>	The times to call the server. It should be a number between 1 and 9999.

**Return value**

Returns the elapsed time (in milliseconds) for the call when the function succeeds, and returns -1 when failing to connect to PowerServer, and -2 for timeout (timeout is 1 second and is not configurable).

**1.3.1.19 of\_popupmenu function****Description**

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

Appeon DataWindow Menu is available in PowerServer Web only. For more information about the menu items, refer to Section 7.4, “Appeon DataWindow menu” in *PowerServer Toolkit User Guide*.

Supported on the Web client only.

## Syntax

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

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

### 1.3.1.20 `of_popupmenuon` function

#### Description

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

Appeon DataWindow Menu is available in PowerServer Web only. For more information about the menu items, refer to Section 7.4, “Appeon DataWindow menu” in *PowerServer Toolkit User Guide*.

Supported on the Web client only.

## Syntax

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

Argument	Description
<code>appeonextfuncs</code>	A reference to an <code>AppeonExtFuncs</code> object.
<code>adw_dw</code>	The DataWindow control on which you want to show the Appeon DataWindow menu.
<code>ab_show</code>	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.

### **1.3.1.21 `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
<code>appeonextfuncs</code>	A reference to an <code>AppeonExtFuncs</code> object.
<code>adw</code>	The <code>DataWindow</code> object to be saved as image files.
<code>asoutpath</code>	The path of image files.
<code>asoutname</code>	The specified name of image files.
<code>alouttype</code>	The format type of files: 1-BMP; 2-JPG; 3-GIF

## Return value

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

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.

### 1.3.1.22 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 10.10.3, “Functions of DataWindow control” in *Supported PB Features for PowerServer Web* or *Supported PB Features for PowerServer Mobile*.

Supported on the Web client only.

### 1.3.1.23 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.
----------	-------------------------------

## 1.3.2 Appeon Client Functions

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

### 1.3.2.1 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 PowerServer application, not in the PowerBuilder application.

### 1.3.2.2 AppeonGetBrowserVersion function

#### Description

Gets the Internet Explorer version of the Web client.

Supported in the browser-based Web application only.

#### Syntax

```
AppeonGetBrowserVersion()
```

#### Return value

String.

#### Usage

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

### 1.3.2.3 AppeonGetCacheDir function

#### Description

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

For the Web application, the cache directory can be customized in AEM. For details, refer to Section 4.4.6.7, “Client Storage Location” in *PowerServer Configuration Guide for .NET* or in PowerServer Configuration Guide for J2EE.

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 PowerServer application, not in the PowerBuilder application.

### 1.3.2.4 `AppeonGetClientID` function

## Description

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

## Syntax

`AppeonGetClientID()`

## Return value

String.

### 1.3.2.5 `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 PowerServer application, not in the PowerBuilder application.

### 1.3.2.6 `AppeonGetClientType` function

## Description

Gets the type of an application.

### Syntax

`AppeonGetClientType()`

### Return value

String	<p>Returns "WEB" if the application runs in the Web browser or on the desktop as an IWA (installable Web app).</p> <p>Returns "PB" if the application runs in PowerBuilder.</p> <p>Returns "MOBILE" if the application runs on a mobile device.</p>
--------	---

### 1.3.2.7 `AppeonGetClientVersion` function

#### Description

Gets the version of the Appeon client that runs your application.

Supported in the mobile application only.

#### Syntax

`AppeonGetClientVersion()`

#### Return value

String.

### 1.3.2.8 `AppeonGetHttpInfo` function

#### Description

Gets the HTTP header information from a particular request.

#### Syntax

`AppeonGetHttpInfo( string attribute )`

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

#### Return value

String

### 1.3.2.9 `AppeonGetIHandle` function

#### Description

Gets the Internet Explorer handle for the Web application.

Supported in the browser-based Web application only.

#### Syntax

`AppeonGetIHandle()`

#### Return value

Long.

### Usage

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

#### 1.3.2.10 **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 PowerServer application, not in the PowerBuilder application.

#### 1.3.2.11 **AppeonGetOSType** function

### Description

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

### Syntax

```
AppeonGetOSType()
```

### Return value

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

#### 1.3.2.12 **AppeonGetRemainingdays** function

### Description

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

This function is unsupported in the PowerServer PB Edition.

### Syntax

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

Argument	Description
as_type	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.	Returns a number >0 if the license or technical support has remaining day(s).  Returns 0 if there is no expiration date.  Returns -1 if license or technical support has expired, PowerServer has an exception, or parameter is invalid.
-------	--

**1.3.2.13 AppeonGetServerType function****Description**

Gets the PowerServer type where the application runs.

**Syntax**

```
AppeonGetServerType()
```

**Return value**

Integer.	Returns 1 if the application runs on a PowerServer that is installed to a Java Server (such as JBoss).  Returns 2 if the application runs on a PowerServer that is installed to a .NET IIS server.
----------	--

**Usage**

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

**1.3.2.14 AppeonGetSessionCount function**

This function is obsolete since Appeon PowerServer 2013. Replace it with the [getSessionCount](#) function in PowerServer open interfaces.

**1.3.2.15 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 PowerServer application, not in the PowerBuilder application.

### 1.3.2.16 **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 PowerServer 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 4.3.5.5, “System Security” in *PowerServer Configuration Guide for .NET* or PowerServer Configuration Guide for J2EE.

### 1.3.2.17 **AppeonNetworkTest** function

#### Description

Calculates the elapsed time for the client to call the server over the network. The returned value can help the developer find out the network state, and then determine whether to do data retrieval or update, especially for a large amount of data. This improves the end user experience, to avoid hanging up the application for a long time when data retrieval or update is performed on a poor network connection.

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

This function works with the .NET IIS server only, not with the J2EE application server.

#### Syntax

```
AppeonNetworkTest ( long lcount )
```

Argument	Description
<i>lcount</i>	The times to call the server. It should be a number between 1 and 9999.

#### Return value

Returns the elapsed time (in milliseconds) for the call when the function succeeds, and returns -1 when failing to connect to PowerServer, and -2 for timeout (timeout is 1 second and is not configurable).

### 1.3.2.18 AppeonPopMenu function

#### Description

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

Appeon DataWindow Menu is available in PowerServer Web only. For more information about the menu items, refer to Section 7.4, “Appeon DataWindow menu” in *PowerServer Toolkit User Guide*.

Supported on the Web client only.

#### Syntax

```
AppeonPopMenu ( 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 top 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 AppeonPopMenu function has a higher priority than the AppeonPopMenuOn 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.

### 1.3.2.19 AppeonPopMenuOn function

#### Description

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

Appeon DataWindow Menu is available in PowerServer Web only. For more information about the menu items, refer to Section 7.4, “Appeon DataWindow menu” in *PowerServer Toolkit User Guide*.

Supported on the Web client only.

#### Syntax

```
AppeonPopMenuOn ( 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.  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.

**1.3.2.20 `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 alouttype is an unsupported image format.
Returns -3 if adw 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.

### 1.3.2.21 **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 10.10.3, “Functions of DataWindow control” in *Supported PB Features for PowerServer Web* or *Supported PB Features for PowerServer Mobile*.

Supported on the Web client only.

### 1.3.2.22 **AppeonSetTextHint function**

#### Description

Sets the HintText property of the SingleLineEdit control. The HintText property is not available in PowerBuilder SingleLineEdit control. It is specially designed for the PowerServer mobile application.

**Note:** This function is obsolete since version 2017.

`appeonsethinttext ( singlelineEdit sle_target, string as_hinttext )`

Argument	Description
<i>sle_target</i>	The SingleLineEdit control.
<i>as_hinttext</i>	The text that will be displayed as hint text in the SingleLineEdit control.

## Return value

Integer.	1 - Success. -1 - It is called in PowerBuilder or PowerServer Web, or there is an error. -2 - The <i>sle_target</i> is invalid. -3 - The <i>as_hinttext</i> argument is null.
----------	--

### 1.3.2.23 AppeonSetLongPressInterval function

#### Description

Sets the duration (in milliseconds) for a press before it is recognized as a long press. Once this duration is reached, the keyboard will be brought up to display. By default, the keyboard pops up when the user taps twice in the field of the DataWindow, or when the user presses the DataWindow field for as long as 1500 milliseconds.

Supported on mobile client only.

#### Syntax

`appeonsetlongpressinterval ( long al_interval )`

Argument	Description
<i>al_interval</i>	Sets the duration in milliseconds. If it is set to a value smaller than 500 milliseconds, then 550 milliseconds will be used, because 500 milliseconds is already used as the duration for the DataWindow drag & drop event.

## Return value

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

### 1.3.2.24 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
<code>adw</code>	The DataWindow/DataStore/ DataWindowChild 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.
----------	-----------------------------------

## 1.3.3 Client Logs

### 1.3.3.1 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.

## 1.3.4 File Upload and Download

Appeon PowerServer 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](#)

### 1.3.4.1 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 PowerServer 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 PowerServer installed to the Java server, the WAR package of Appeon File Server (`fileservice.war`) can be found under `%appeon%\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, and JEUS. For more about the deploy instructions, refer to related documents of IIS, WebLogic, WebSphere, JBoss, and JEUS.

#### 1.3.4.1.1 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 PowerServer provides a **setup.exe** file at `%IIS_Web_Root%\appeon\plugin\fileservice`.

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

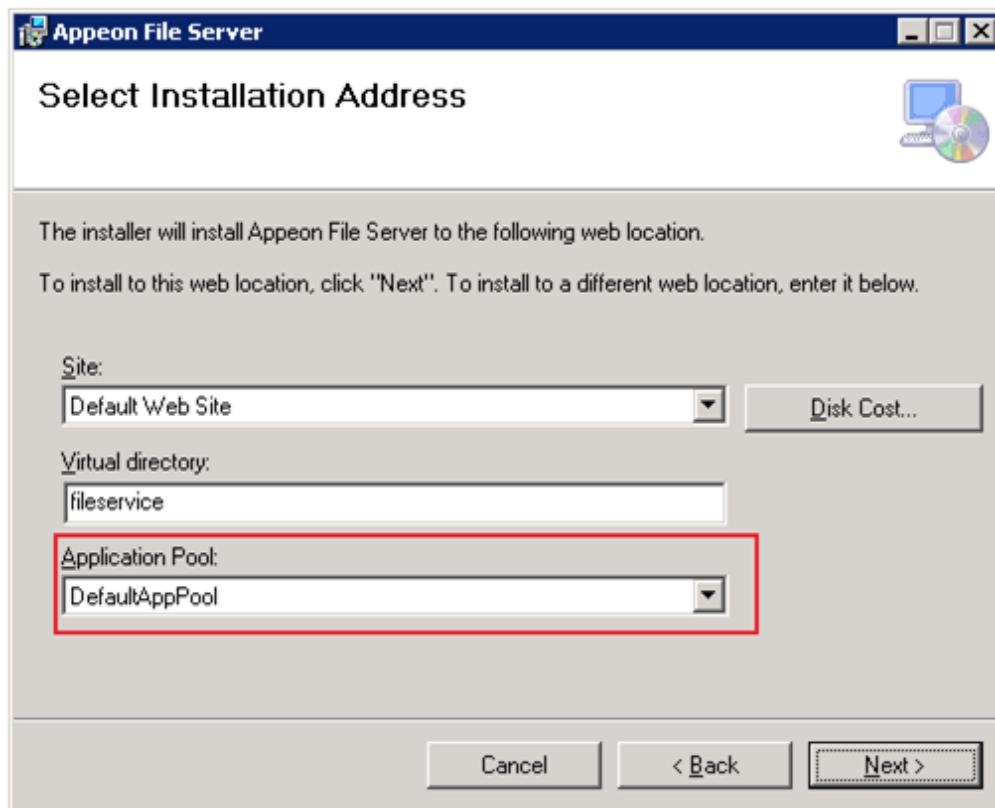
If the "Installation Incomplete" error displays, please try the solution in the Section 1.4.1, "Appeon File Server Installation Incomplete" in *PowerServer Troubleshooting Guide*.

**Figure 1.1: Appeon File Service Setup Wizard**



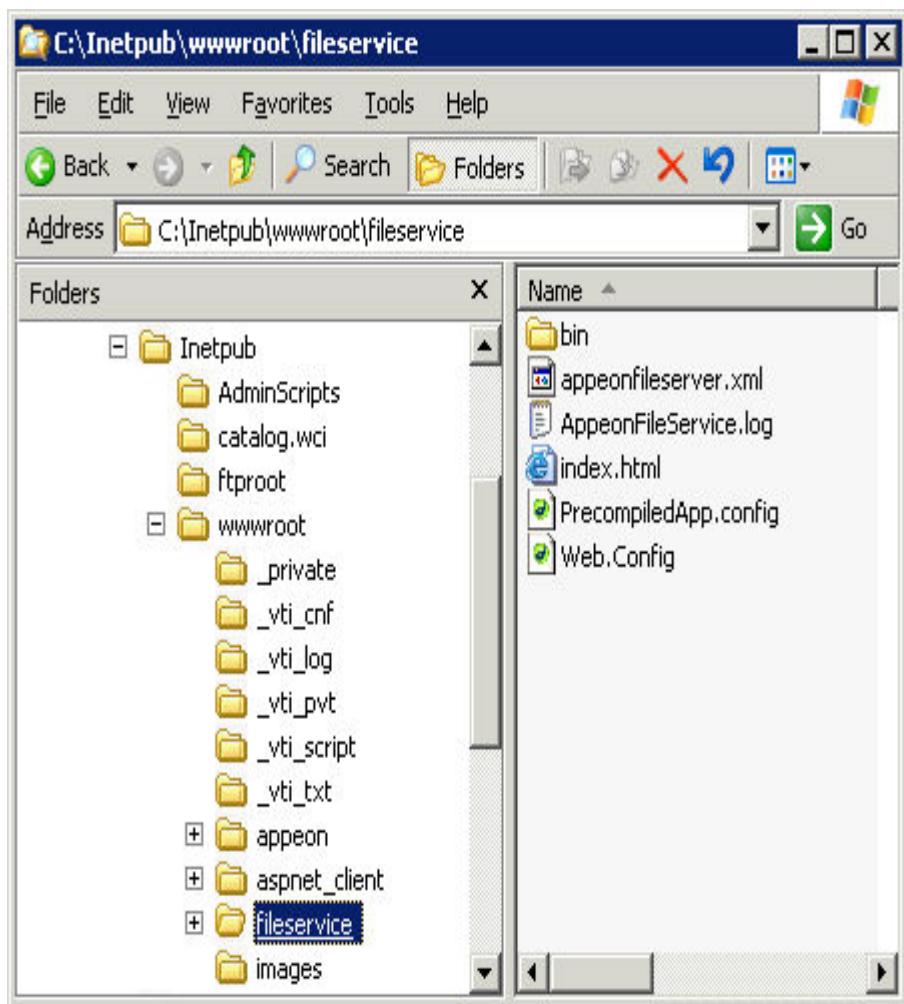
Step 2: Click the **Next** button and then the following window pops up. Select a Web site where the Appeon File Server will be installed.

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

**Figure 1.2: Select Installation Address**

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

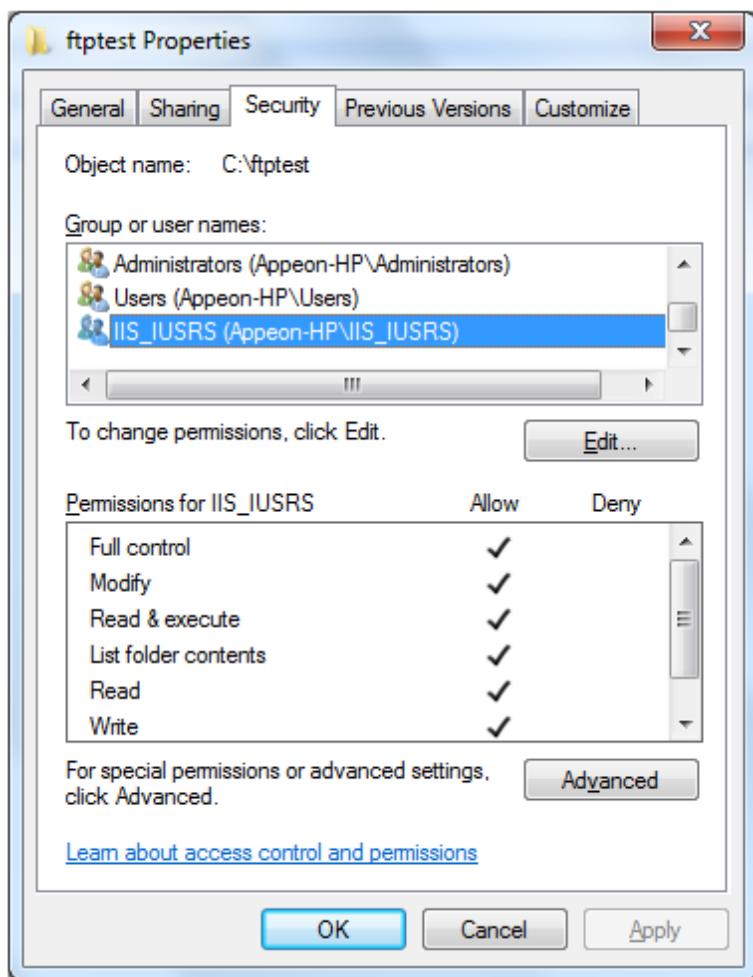
Step 4: Go to directory where the Appeon File Server is installed, for example (*C:\inetpub\wwwroot\fileservice*).

**Figure 1.3: Appeon 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 7.5 or above**, grant the **IIS\_IUSRS** user with **Full Control** permission to this folder.

**Figure 1.4: 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 `appeonfileserver.xml` in `C:\Inetpub\wwwroot\fileservice`.

```
<?xml version="1.0" encoding="UTF-8"?>
<webserver>

<file-path value="D:\appeon\upload" />
<log-level value="3" />           <!-- 0 Error, 1 Info, 2 Func, 3 Debug--&gt;
&lt;session-timeout value="3600" /&gt;  <!-- the unit is second --&gt;
&lt;allowed-file-types value="txt;doc;jpg;mpeg" ignorecase="true" /&gt;
&lt;max-file-size value="20" /&gt;      <!-- the unit is M --&gt;
&lt;users&gt;
  &lt;user name="test" password="password" /&gt;
  &lt;user name="userA" password="userA" /&gt;
  &lt;user name="userB" password="userB" /&gt;
&lt;/users&gt;</pre>

```

```
</webserver>
```

Modify the file according to your own demands.

**Table 1.2: Settings specification value**

Settings	The value of the setting specifies...
<file-path>	Specifies the directory to which files are saved after uploaded to the file server. Files are automatically saved in the "plugin" folder under the application directory when downloaded from the file server.
<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. <ol style="list-style-type: none"> <li>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.</li> <li>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.</li> <li>3. "Ignorecase" - The file type is not case sensitive.</li> </ol>
<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.

### 1.3.4.1.2 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 1.3: The settings specification value**

Settings	The value of the setting specifies...
<file-path>	Specifies the directory to which files are saved after uploaded to the file server. Files are automatically saved in the "plugin" folder under the application directory when downloaded from the file server.
<session-timeout>	Specifies the time, in seconds, that passes after the last request is processed before the session times out.
<allow-file-type>	<p>Specifies the file types that can be uploaded. Use a ";" between two file types.</p> <ol style="list-style-type: none"> <li>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.</li> <li>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.</li> <li>3. "Ignorecase" - The file type is not case sensitive.</li> </ol>
<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 PowerServer is installed, unzip the **fileservice.war** (%powerserver%\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>fileservice</servlet-name>
  <servlet-class>com.appeon.fileservice.WebServiceDispatcher</servlet-class>
  <init-param>
    <param-name>config</param-name>
    <param-value>D:\appeon\config\appeonfileservice.xml</param-value>
  </init-param>

```

```
<load-on-startup>1</load-on-startup>
</servlet>
<servlet>
  <servlet-name>uploadfile</servlet-name>
  <servlet-class>com.appeon.fileservice.UploadFile</servlet-class>
  <init-param>
    <param-name>config</param-name>
    <param-value>D:\appeon\config\appeonfileserv.xml</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>

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

</web-app>
```

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

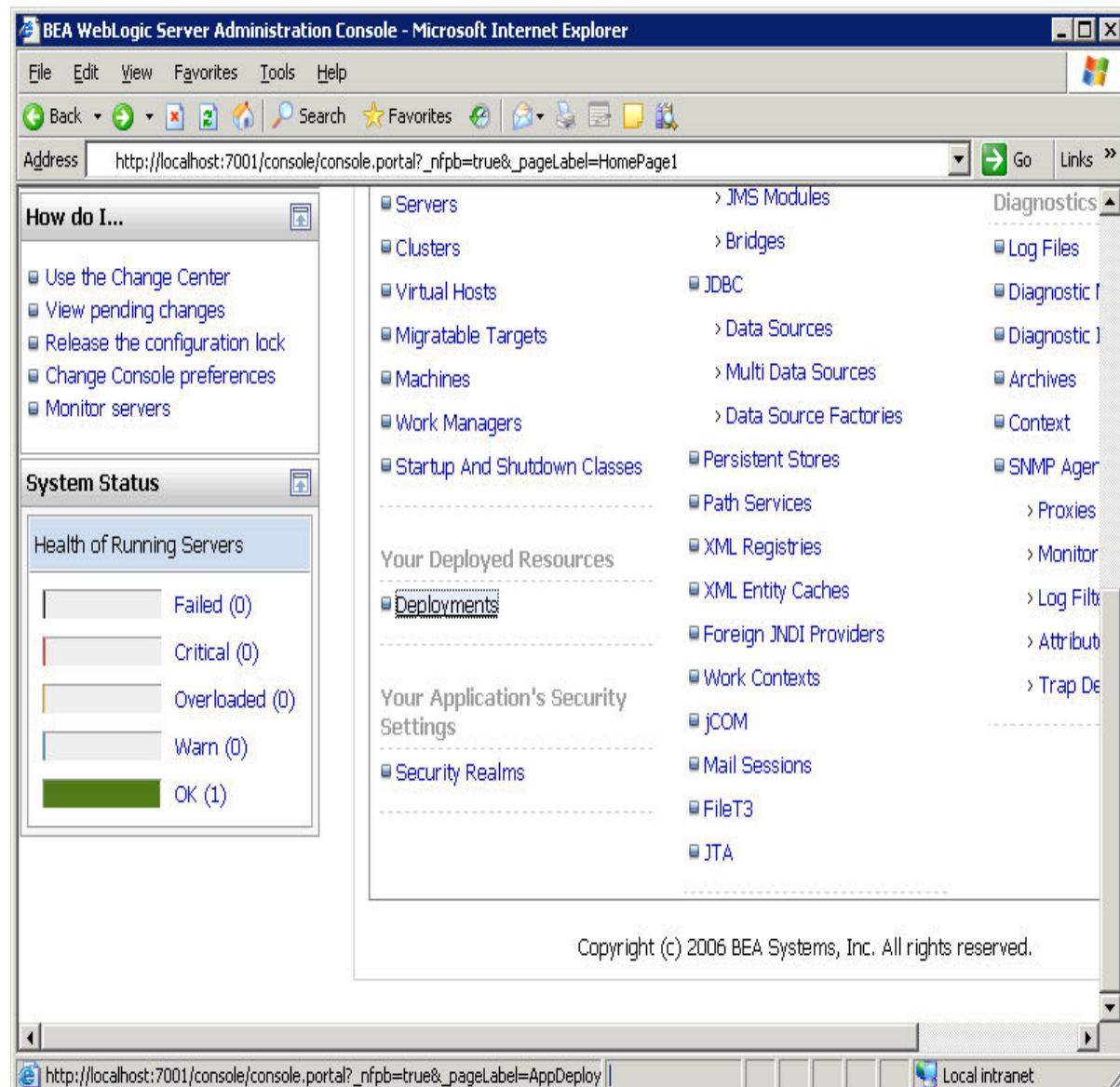
Then follow the next section [Deploying the Appeon File Server](#) to deploy fileservice.war.

### Deploying the Appeon File Server

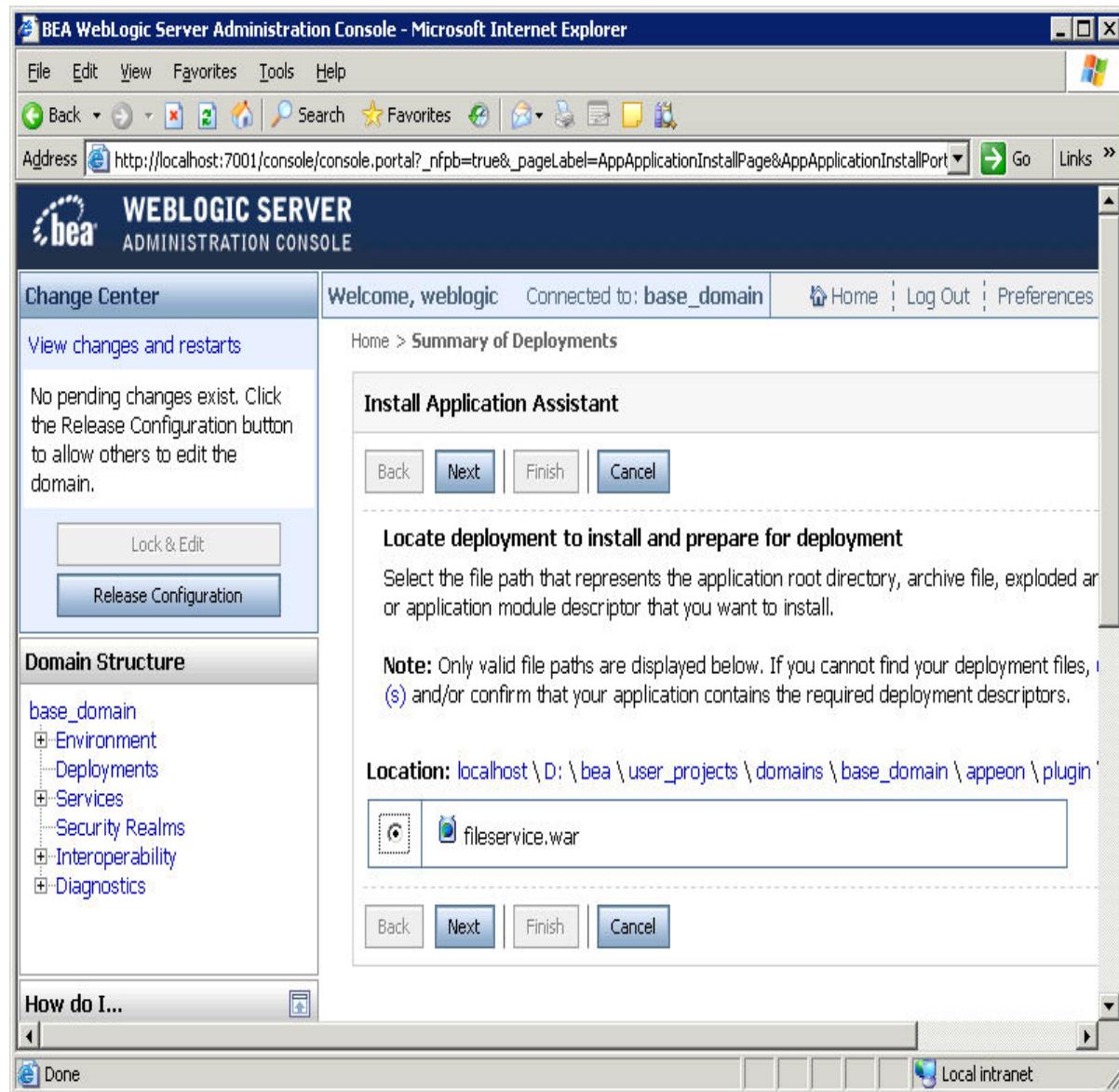
After configuring the Appeon File Server, follow steps below to deploy it:

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

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

**Figure 1.5:**

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

**Figure 1.6:**

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.

#### 1.3.4.1.3 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 etc. is the same. Please refer to [Configuring the Appeon File Server](#) for WebLogic.

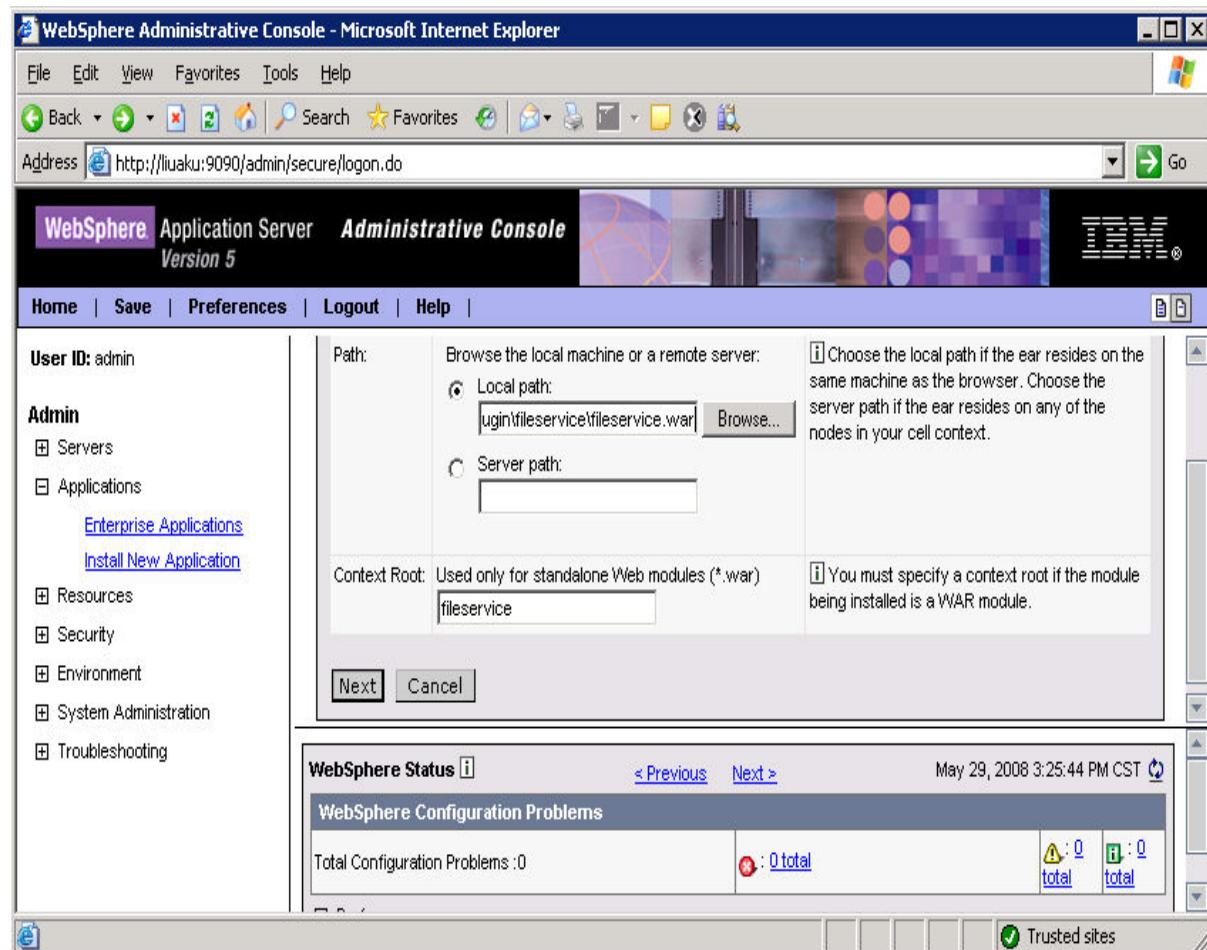
##### Deploying the Appeon File Server

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

Step 1: On the machine where PowerServer 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 `%powerserver%\plugin\fileservice`) and **Context Root** for the WAR file and click the **Next** button.

**Figure 1.7:**



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

#### 1.3.4.1.4 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 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

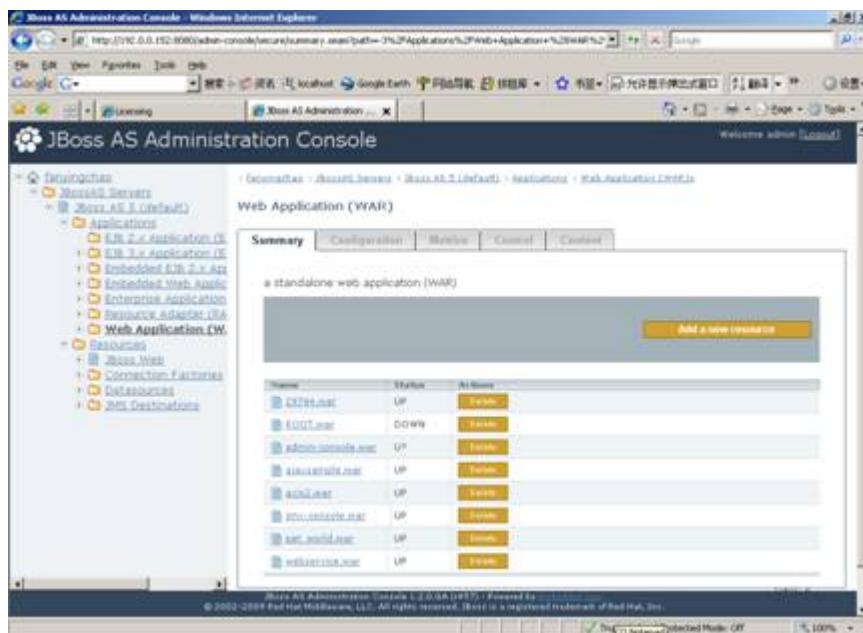
Copy the file **fileservice.war** to the directory `%JBoss installation root directory%\standalone\deployments` and then start JBoss.

The fileservice.war file will be deployed automatically.

- Manual deploy

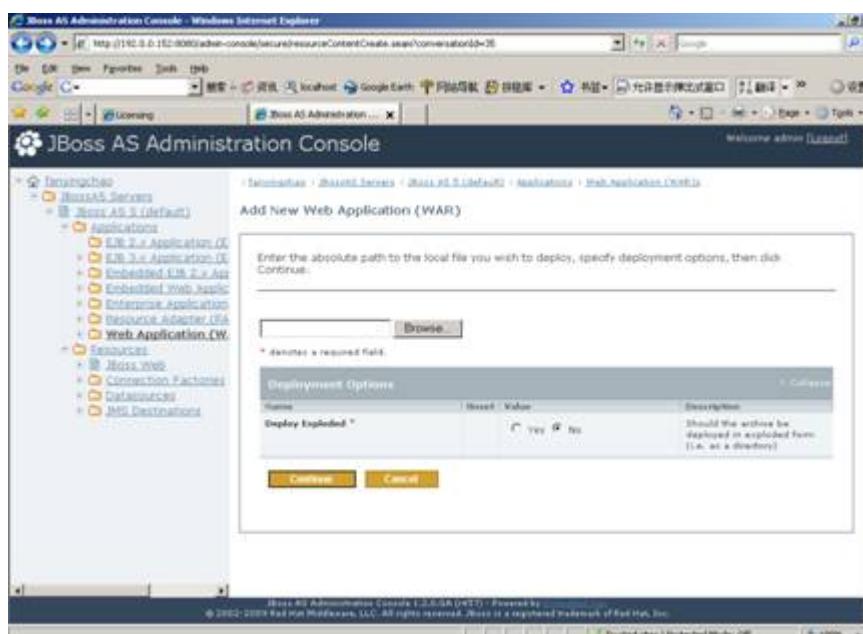
1. Start JBoss and log in to JBoss console.
2. Select **Web Application** in the left tree view and then click **Add a new resource** in the right page.

**Figure 1.8: PDF**



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

**Figure 1.9:**



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

#### 1.3.4.1.5 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 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.

#### 1.3.4.2 Uploading and downloading files

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

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

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

Step 3: Call [of\\_logonfileserver](#) to connect to the file server.

Step 4: Call [of\\_appeonupload](#) or [of\\_appeondownload](#) to upload or download files.

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

##### 1.3.4.2.1 AppeonFileService object

AppeonFileService 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\\_logonfileserver function](#): Connects to the file server.
- [of\\_logofffileserver 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\_logonfileserver****Description**

Connects to the Appeon File Server.

**Syntax**

```
appeonfileservice.of_logonfileserver (string fileserveripaddress, long port, string connectstring)
```

Parameter	Description
appeonfileservice	An instance of an AppeonFileService object
<i>fileserveripaddress</i>	Protocol and IP address or machine name of the Appeon File Server. To use the HTTPS protocol, you should add "https://" before the IP address or machine name, for example, "https://192.168.168.52", otherwise, HTTP protocol (http://) will be used by default.
<i>port</i>	HTTP or HTTPS port of the application where the Appeon File Server is deployed.
<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 those specified in the appeonfileserver.xml configuration file.

**Return value**

Long.	<p>Return values are:</p> <ul style="list-style-type: none"> <li>0 - The logon is successful.</li> <li>-1 - Connection to the file server fails.</li> <li>-2 - The provided user name or password is invalid.</li> </ul>
-------	--

**Code example**

```
long ll_re
string info

ll_re = fileservice.of_logonfileserver ("192.168.168.52", 80,
"username=userA;password=userA")

if ll_re = 0 then
  info = "Logon to file server is successful."
elseif ll_re = 1 then
  info = "Failed to connect to the file server."
else
  info = "Failed to connect to the file server."
```

```
end if

Messagebox( "ll_re="+string(ll_re),info)
```

**of\_logofffileserver****Description**

Disconnects from the Appeon File Server.

**Syntax**

```
appeonfileservice.of_logofffileserver()
```

Parameter	Description
appeonfileservice	An instance of an AppeonFileService object

**Return value**

Long.	<p>Return values are:</p> <ul style="list-style-type: none"> <li>0 - The log-off is successful.</li> <li>-1 - Connection to the Appeon File Server fails.</li> <li>-2 - The session has timed out or the user has not logged off the file server.</li> </ul>
-------	--

**Code example**

```
long ll_re
string info

ll_re = fileservice.of_logofffileserver ( )

if ll_re = 0 then
  info = "Logoff file server is successful."
elseif ll_re = 1 then
  info = "Failed to log off file server."
else
  info = "The session has timed out or the user has not logged off the file server."
end if

Messagebox( "ll_re="+string(ll_re),info)
```

**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)
(Web only)
```

`appeonfileservice.of_appeonupload (boolean isrename, ref long errorcode)` (Web only)

Parameter	Description
<code>appeonfileservice</code>	An instance of an <code>AppeonFileService</code> object.
<code>source</code>	<p>The file name and directory of the source file.</p> <p>For the mobile application, this parameter is <b>required</b>; it must be specified with a valid directory.</p> <p>For the Web application, this parameter is <b>optional</b>. If no directory is specified here, a dialog box will be displayed prompting the user to select a file to upload.</p>
<code>destination</code>	<p>The destination directory on the file server where the file is uploaded. The directory specified here is a relative path to the file directory specified in the <code>appeonfileserver.xml</code> configuration file.</p> <p>This parameter is <b>optional</b> to both the Web application and the mobile application.</p>
<code>isrename</code>	Whether to rename the file if another file with the same name already exists.
<code>errorcode</code>	<p>The error occurred when uploading files. Values are:</p> <ul style="list-style-type: none"> <li>0 - Succeeded.</li> <li>-1 - Connection to the file server fails.</li> <li>-2 - ID error.</li> <li>-3 - The format of the source file is forbidden.</li> <li>-4 - Destination directory is invalid or the file is too large.</li> </ul>

## 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.

## Code example

```
long ll_re
string info,ls_url

ls_url = fileservice.of_appeonupload ("C:\temp\1.pdf", "pdf_files", true, ll_re)

if ll_re = 0 then
  info = "Upload is successful."
elseif ll_re = 1 then
  info = "Failed to upload the file."
```

```

else
    info = "Other errors."
end if

Messagebox("ll_re="+string(ll_re),info+"~r~nurl="+ls_url)

```

## **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 file name and directory of the file which will be downloaded. The directory specified here is a relative path to the file directory specified in the appeonfileserver.xml configuration file.
<i>target</i>	(Optional) The destination directory on the client machine where the file will be saved after downloaded from the file server.  For the mobile application, this parameter is ignored, as the file will be automatically saved to the "plugin" folder under the application directory. You can determine the directory for the current application by calling the <a href="#">AppeonGetCacheDir function</a> .  For the Web application, this parameter is optional. 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.
-------	--

### **Code example**

The following code example is to download a PDF file named "1.pdf" from the "pdf\_files" folder under the specified directory on the Appeon File Server. On the Web client, the file will be saved to "C:\\" as specified by the *target* parameter; while on the mobile device, the file will be automatically saved to the "plugin" folder for the current application, despite that the *target* parameter is set to "C:\\".

```
long ll_re
string info

ll_re = fileservice.of_appeondownload ("pdf_files\1.pdf", "C:\\")

if ll_re = 0 then
  info = "Download is successful."
else
  info = "Other errors."
end if

Messagebox ("ll_re="+string(ll_re),info)
```

The following code example is to view the PDF file (named "1.pdf" located on the local C drive) via Internet Explorer. This is effective for the Web client only.

```
string ls_file

ls_file = "~"C:\Program Files\Internet Explorer\iexplore.exe~" C:\1.pdf"
Messagebox("",ls_file)
run(ls_file)
```

## **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 <code>AppeonFileService</code> 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.

### **Code example**

```
long ll_re
string info

ll_re = fileservice.of_fileexists ("pdf_files\1.pdf")

if ll_re =1 then
```

```
info = "The file exists."
elseif ll_re =0 then
  info = "The file does not exist on the file server."
else
  info = "Other errors."
end if

Messagebox( "ll_re="+string(ll_re),info)
```

### 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.

## **1.3.5 Appeon Labels**

### **Overview**

Appeon Labels provided in the Appeon Workarounds PBL can reduce the interactions between the client and PowerServer, thus boosting the performance of PowerServer 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
<a href="#">Appeon Commit/Rollback Label</a>	<code>of_autocommitrollback</code>	Automatically commits or rolls back the first database operation statement after the label.
<a href="#">Appeon Commit Label</a>	<code>of_autocommit</code>	Automatically commits the first database operation.
<a href="#">Appeon Rollback Label</a>	<code>of_rollback</code>	Automatically rolls back the first database operation statement if the operation fails.
<a href="#">Appeon Queue Labels</a>	<code>of_startqueue</code> <code>of_commitqueue</code>	The Appeon Queue Labels are designed for use when it is necessary to execute numerous database operation statements on PowerServer, 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.
<a href="#">Appeon Immediate Call Label</a>	<code>of_imdcall</code>	Immediately commits a database operation.
<a href="#">Appeon Update Label</a>	<code>of_update</code>	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 5.5, “Technique #4: grouping multiple server calls with Appeon Labels” in *PowerServer Performance Tuning Guide*.

#### 1.3.5.1 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 <code>appeon_nvo_db_update</code> object.

## Return value

None.

## Usage

With the Appeon Commit/Rollback Label, the database operation statement will be sent to the PowerServer. 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 PowerServer. 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 business logic
    COMMIT;
ELSE
    ..... // non-database related business 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.

### 1.3.5.2 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, PowerServer 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 PowerServer , 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.

```
gnv_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.

**1.3.5.3 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\_autorollback

**Syntax**

*objectname*.of\_autorollback( )

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, PowerServer only commits an unsuccessful database operation.

The first Rollback statement after the Appeon Rollback Label will not be submitted to the PowerServer 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

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

#### 1.3.5.4 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 PowerServer, 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 appeon_nvo_db_update 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 of_startqueue function without this argument is preserved for compatibility.

*objectname.of\_commitqueue()*

Argument	Description
objectname	An instance of the appeon_nvo_db_update 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 PowerServer.

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
gnv_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()
gnv_appeonDbLabel.of_autocommitrollback() // the label takes effect
gnv_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()
gnv_appeonDbLabel.of_autocommitrollback() // the label takes effect
gnv_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;
gnv_appeonDbLabel.of_commitqueue()

```

### 1.3.5.5 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

```
gnv_appeonDbLabel.of_startretrievequeue()
dw_1.retrieve()
gnv_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)
gnv_appeonDbLabel.of_endretrievequeue()
```

### 1.3.5.6 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
<i>objectname</i>	An instance of the <code>appeon_nvo_db_update</code> object.
<i>transactionflag</i>	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.

Argument	Description
<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.	<p>Return values are:</p> <ul style="list-style-type: none"> <li>1 - Succeed in update</li> <li>-101 - Fail to update the first DataWindow/DataStore/DataWindowChild</li> <li>-102 - Fail to update the second DataWindow/DataStore/DataWindowChild</li> <li>-103 - Fail to update the third DataWindow/DataStore/DataWindowChild</li> <li>-104 - Fail to update the fourth DataWindow/DataStore/DataWindowChild</li> </ul>
----------	---

## Usage

The update operations of the DataWindows, DataStores, or DataWindowChild will be submitted to the PowerServer together. If the operation of a DataWindow, DataStore or DataWindowChild fails, PowerServer 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

```

### 1.3.6 Calling EJB Component

In order to call EJB components in applications deployed with Appeon PowerServer, Appeon PowerServer provides its own EJB solution which includes a customized object ([EJBObject object](#) in Appeon Workarounds PBL), a DLL (eonejbclient.dll), and a bridge ([Appeon Bridge](#)). With this EJB solution you can call an EJB component in both the PowerBuilder Client/Server application and the deployed PowerServer application.

Compared to the EJB solution of PowerBuilder, Appeon EJB solution can support more complex parameters such as Structure; and there is no need to generate the proxy object in PowerBuilder.

eonejbclient.dll -- Located in the same folder as the Appeon Workarounds PBL, this DLL is used by the PowerBuilder Client/Server application only, not by the deployed PowerServer application. If you want to make the PowerBuilder Client/Server application call the EJB component successfully using the same Appeon EJB solution as the deployed application uses, make sure this DLL is included in the PowerBuilder application directory on the client machine.

#### 1.3.6.1 Supported application server

The Appeon EJB solution supports calling EJB 1.x/2.x/3.x components deployed to the J2EE-compliant application servers only (no .NET IIS application server). The certified supported application server includes WildFly 18, JBoss EAP 7.2, WebSphere 7/8/8.5, & WebLogic 12c.

#### 1.3.6.2 Generate the EJB proxy

JBoss application server requires no EJB proxy generated, but WebLogic and WebSphere application server requires you to generate the EJB proxy and copy the generated EJB proxy to Appeon Bridge.

To generate the EJB proxy in the WebSphere application server:

1. Run the DOS command window as an administrator, and then change the directory to *IBM\WebSphere\AppServer\bin*.
2. Execute this command:

```
createEJBStubs.bat name of the JAR file or EAR file
```

To generate the EJB proxy in the WebLogic application server:

1. Run the DOS command window as an administrator, and then change the directory to *domains\base\_domain\bin*.
2. Execute this command:

```
setDomainEnv.cmd
```

3. Change the directory back to *domains\base\_domain\bin*, and execute this command:

```
java weblogic.appc name of the JAR file or EAR file
```

The proxy functions will be automatically generated into the JAR file and the JAR file that contains the proxy functions is what we call the EJB proxy. Follow the next section to copy/bundle the EJB proxy to the Appeon Bridge.

### 1.3.6.3 Bundle EJB proxy with Appeon Bridge

#### Introduction to Appeon Bridge

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 *%appeon%\plugin\appeonbridge* under the PowerServer installation folder.

#### Bundle EJB proxy with Appeon Bridge

Before deploying Appeon Bridge to the application server, you must bundle the EJB proxy (JAR file) with Appeon Bridge to implement the communication to EJB components.

Three ways of bundling the EJB proxy with Appeon Bridge:

- Method 1: Add the proxy to the **lib** directory in the **appeonbridge.war** file. Since the **appeonbridge.war** file is a ZIP file, you can check the directory in the WAR file with a third-party tool such as WinZip.
  1. Navigate to *%appeon%\plugin\appeonbridge* folder and open **appeonbridge.war** with the third-party tool such as WinZip. *%appeon%* indicates the installation folder of the PowerServer.
  2. Navigate to the folder where the EJB proxy is located.
  3. Copy the entire folder of the EJB proxy to the *appeonbridge.war\WEB-INF\lib* directory (you will need to create the **lib** folder first).

This method does not work for EJB 3.x in JBoss; try the second method if you are using EJB 3.x in JBoss.

- Method 2: Add the classes of the EJB component to the **classes** directory in the **appeonbridge.war** file.
  1. Navigate to *%appeon%\plugin\appeonbridge* folder and open **appeonbridge.war** with the third-party tool such as WinZip. *%appeon%* indicates the installation folder of the PowerServer.

2. Navigate to the folder where the EJB classes are located.
  3. Copy the entire folder of the EJB classes to the *appeonbridge.war\WEB-INF\classes* directory.
- Method 3: Copy the EJB proxy to the **lib** directory in the Java Web server, and add the **lib** directory (where the proxy is stored) to the CLASSPATH environment variable of the machine that hosts the Java Web server.

#### 1.3.6.4 Deploy to application server

Now you are ready to deploy the following files to the application server one by one:

- Appeon Bridge (appeonbridge.war) which includes the EJB proxy. View [Generate the EJB proxy](#) for how to generate the EJB proxy and [Bundle EJB proxy with Appeon Bridge](#) for how to copy the EJB proxy into appeonbridge.war.
- Original JAR file for the EJB component without the EJB proxy functions.
- PowerBuilder application that calls the EJB component. View [Call EJB component in PowerBuilder application](#) for how to write PowerScript to call the EJB component functions.

Deploying any of the above files 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 the above file is the same as deploying any other applications.)

#### 1.3.6.5 Call EJB component in PowerBuilder application

This section guides you through writing PowerScript to call the EJB component. Appeon PowerServer provides an object called EJBObject in the Appeon Workarounds PBL to call the EJB component, and provides Appeon Bridge to communicate with the EJB component.

Step 1: Add the Appeon Workarounds PBL to the Library Search Path of your PowerBuilder application.

Step 2: Declare the EJBObject object.

Step 3: Configure the connection string, call the EJBObject [connectserver](#) function to connect with Appeon Bridge on the application server.

The connection string varies according to the application server. View [connectserver](#) to determine the connection string.

Step 4: Create the instance for EJB component using EJBObject [lookupjndi](#) function or [createremoteinstance](#) function.

Note: For EJB 3.x, **lookupjndi** function works exactly as the **createremoteinstance** function; both functions can directly return the corresponding instance session handle; while for EJB 2.x, **lookupjndi** function will need to obtain the home interface of the

EJB component first, for example, ls\_msg = lo\_ejb.lookupjndi ("TestSBeanLess",ref ll\_homeHandle), before it can return the corresponding instance handle.

The JNDI name has to be accurate, and it varies according to the application server. For WebSphere, you will also need to set the JNDI name first in the WebSphere console. View [JNDI Name](#) for more information.

Step 5: If the function to be called contains parameter(s), call the [EJBObject registering functions](#) to register the parameter.

Step 6: Call the [EJBObject invoking component functions](#) to invoke the EJB component method.

Step 7: Destroy the instance and disconnect from the application server.

Below is a PowerScript code example that calls an EJB 3.x component:

```
ejbobject lo_ejb //Declare EJBobject
long ll_beanc = 0 //For receiving instance's handle
string ls_prop[5]
string ls_serurl,ls_msg,ls_msg1,ls_parm
//appeonbridge URL
ls_serurl = "http://"+ip+":"+port1+"/appeonbridge/Dispatch"
ls_prop[1]= is_appname //the deployed Appeon app name
ls_prop[2] = "javax.naming.Context.INITIAL_CONTEXT_FACTORY=
'org.jboss.as.naming.InitialContextFactory'"
ls_prop[3] = "javax.naming.Context.PROVIDER_URL='remote://"+ip+":"+port2+"'"
ls_prop[4] = "username="
ls_prop[5] = "password="
ls_msg = lo_ejb.connectserver(ls_serurl, ls_prop) //Connect with server
if ls_msg = "" then
    //Connection is successful
else
    MessageBox("Not connected!",ls_msg)
end if
ls_jndi = sle_54.text //JNDI name has to be accurate otherwise instance will fail
to create
//EJB 3.x supports passing any character in the third parameter
ls_msg = lo_ejb.createremoteinstance(ls_jndi,"AllDataType","sayHello", ref ll_beanc)
if ls_msg = "" then
    //Instance is created successfully
    sle_53.text = string(ll_beanc)
else
    MessageBox("createremoteinstance Failed!",ls_msg)
end if
ls_parm = "127.0.0.1"
//Register the parameter that will be called by the EJB functions
ls_msg = lo_ejb.regstring(ls_parm)
if ls_msg = "" then
else
    MessageBox("Regstring Failed !",ls_msg)
end if
//Call the function. Function name is case sensitive
ls_msg = lo_ejb.invokeretstring(ll_beanc,"sayHello1",true, ref ls_msg1)
if ls_msg = "" then
    //Called successfully
    sle_52.text = ls_msg1
else
    MessageBox("invokestring Failed!",ls_msg)
end if
//Disconnect from server
ls_msg = lo_ejb.destroyremoteinstance(ll_beanc)
if ls_msg <> '' then
```

```
MessageBox( "destroyremoteinstance Failed!", ls_msg)
end if
ls_msg = lo_ejb.disconnection( )
if ls_msg <> '' then
  MessageBox( "disconnection Failed!", ls_msg)
end if
```

### 1.3.6.5.1 JNDI name

Different application servers have different JNDI names.

#### JNDI name for JBoss

To call EJB 3.x in JBoss, the JNDI name should be in this format: ejb:" + appName + "/" + moduleName + "/" + distinctName + "/" + beanName + "!" + viewClassName

appName: name of the EAR file. If it is JAR file instead of EAR file, then leave this empty.

moduleName: name of the deployed JAR file; file extension should not be contained here, for example, HelloWorld.

distinctName: if no such name is defined, then leave this empty.

beanName: name of the implementation class.

viewClassName: name of the interface and package, for example, com.ibytecode.business.HelloWorld.

#### JNDI name for WebSphere

To call EJB in WebSphere, you will need to define the JNDI name by yourself:

1. Log into the WebSphere Administration console.
2. In the left pane, expand **Applications > Application Types**, and then select **WebSphere enterprise applications**.
3. In the right pane, click the name of the EJB component, and then click **Bind EJB Business or EJB JNDI Names**.
4. Define the name in the **JNDI Name** text box.

#### JNDI name for WebLogic

To call EJB in WebLogic 12c, you can find out the JNDI name here:

1. Log into the WebLogic Server Administration Console.
2. In the left pane, expand **Environment > Servers**.
3. On the **Server summary** page, click the name of the server.
4. On the server **Settings** page, click **View JNDI Tree**.

The JNDI tree is displayed in a new browser window.

5. In the left pane for the JNDI tree, expand the **java:global** node, find the EJB component, expand the component to find the class, and select the class.

The **Bind Name** displayed in the right page is the JNDI name you are looking for.

### 1.3.6.5.2 EJBObject object

Appeon EJBObject object implements the interaction between client and Appeon Bridge. To use this object, you need to load **appeon\_workaround.pbl** into your application.

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

Function	Description
<a href="#">ConnectServer function</a>	Connecting and disconnecting to Appeon Bridge
<a href="#">DisConnection function</a>	
<a href="#">LookUpJndi function</a>	Obtaining the home interface of an EJB component
<a href="#">CreateRemoteInstance function</a>	Creating and destroying the instance for an EJB component
<a href="#">DestroyRemoteInstance function</a>	
<a href="#">Registering parameter functions</a>	Registering parameters
<a href="#">Invoking component functions</a>	Invoking EJB components
<a href="#">InitLocalLanguage function</a>	Setting the language of the error message

#### 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. Different application servers have different URL. See below for examples.
<i>properties</i> []	Properties of Appeon Bridge.

##### Return value

String.

Returns empty string ("") if it succeeds.

Here are a few examples of the URL for different application servers:

Example for EJB 1.x/2.x in JBoss:

```
String ls_serurl,ls_prop[5]
ls_serurl = "http://127.0.0.1:8080/appeonbridge/Dispatch"
ls_prop[1]= "testappeon" //name of the deployed PowerServer application
```

```

ls_prop[2]=
"javax.naming.Context.INITIAL_CONTEXT_FACTORY='org.jnp.interfaces.NamingContextFactory'"
ls_prop[3]= "javax.naming.Context.PROVIDER_URL='jnp://127.0.0.1:8080'"
ls_prop[4]= "username="
ls_prop[5]= "password="

```

Example for EJB 3.x in JBoss:

```

String ls_serurl,ls_prop[5]
ls_serurl= "http://127.0.0.1:8080/appeonbridge/Dispatch"
ls_prop[1]= "testappeon" //name of the deployed PowerServer application
ls_prop[2]= "javax.naming.Context.INITIAL_CONTEXT_FACTORY=
'org.jboss.as.naming.InitialContextFactory'"
ls_prop[3]= "javax.naming.Context.PROVIDER_URL= 'remote://127.0.0.1:4447'"//default
port is 4447.
//Port number can be configured in ${Home_DIR}\standalone\configuration
\standalone.xml.
ls_prop[4]= "username="
ls_prop[5]= "password="

```

Example for WebSphere:

```

String ls_serurl,ls_prop[5]
ls_serurl = "http://127.0.0.1:9080/appeonbridge/Dispatch"
ls_prop[1]= "testappeon" //name of the deployed PowerServer application
ls_prop[2] = "javax.naming.Context.INITIAL_CONTEXT_FACTORY='com.ibm.websphere.naming.WsnInitialCont
ls_prop[3]= "javax.naming.Context.PROVIDER_URL='iiop://127.0.0.1:2809'"//default
port is 2809.
//Port number can be obtained or changed in WebSphere console > server1 >
BOOTSTRAP_ADDRESS.
ls_prop[4]= "username=admin"
ls_prop[5]= "password=admin"

```

Example for WebLogic:

```

String ls_serurl,ls_prop[5]
ls_serurl = "http://127.0.0.1:7001/appeonbridge/Dispatch"
ls_prop[1]= "testappeon" //name of the deployed PowerServer application
ls_prop[2]=
"javax.naming.Context.INITIAL_CONTEXT_FACTORY='weblogic.jndi.WLInitialContextFactory'"
ls_prop[3]= "javax.naming.Context.PROVIDER_URL='t3://127.0.0.1:7001'"
ls_prop[4]= "username=weblogic"
ls_prop[5]= "password=appeon123"

```

## 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.

## LookUpJndi function

### Description

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

### Syntax

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

Parameter	Description
EJObject	A reference of an EJObject.
<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.

## InitLocalLanguage function

### Description

Sets the language of the error message in PowerBuilder IDE.

### Syntax

EJObject.initlocallanguage (long *nlocalcode*)

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

### Return value

Long.

## Invoking component functions

Invokeretblob	Invokeretblobarray	Invokeretbool	Invokeretboolarray
Invokeretchar	Invokeretchararray	Invokeretdate	Invokeretdatearray

Invokereretdatetime	Invokereretdatetimearray	Invokereretdouble	Invokereretdoublearray
Invokeretint	Invokeretintarray	Invokeretlong	Invokeretlongarray
Invokeretreal	Invokeretrealarray	Invokeretstring	Invokeretstringarray
Invokeretttime	Invokeretttimearray	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 retval argument keeps the consistence with the data type used in the invoking methods, except the Invokeretstru and Invokeretstruarray, the data type of the retval 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.

### **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**

EJBObject.RegBlob(blob *data*)

Parameter	Description
EJBObject	A reference of an EJBObject.
<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**

`EJBObject.regstructarray (any data [], string javaclassname, readonly classdefinition cdef)`

Parameter	Description
<code>EJBObject</code>	A reference of an EJBObject.
<code>data</code>	The parameter to be registered and the name is user-defined.
<code>javaclassname</code>	The name of the corresponding Java class in application server.
<code>cdef</code>	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`.

### CreateRemoteInstance function

#### Description

Creates the instance for an EJB component.

#### Syntax

`EJBObject.CreateRemoteInstance(string jndiname, string homename, string methodname, ref long beanid)`

Parameter	Description
<code>EJBObject</code>	A reference of an EJBObject.
<code>jndiname</code>	The JNDI name of the EJB component.
<code>homename</code>	The name of the home interface of an EJB component.
<code>methodname</code>	The name of the method.
<code>beanid</code>	The handle to the EJB component.

### Return value

String.

Returns empty string ("") if it succeeds.

### **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.

#### **1.3.6.5.3 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[] getPBMap() {
        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;
    }
}
```

### 1.3.7 Calling .NET/COM server components (.NET only)

#### Applies to

PowerServer 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 PowerServer 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 PowerServer 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 PowerServer.

### 1.3.7.1 AppeonDotNetComponent object

#### 1.3.7.1.1 Properties

Properties for AppeonDotNetComponent.

Properties	Type	Description
ComponentType	String	<p>The type of the component to be called.</p> <p>"1" indicates a .NET Assembly to be called.</p> <p>"2" indicates an unmanaged-code COM component to be called.</p> <p>"3" indicates a managed-code COM component to be called.</p> <p>"4" indicates a built-in Appeon Workaround .NET Assembly to be called.</p>
TypeLib	String	The name of the component library. PowerServer uses this name to find the component.

Properties	Type	Description
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.

### 1.3.7.1.2 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)
```

### 1.3.7.1.3 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.

### 1.3.7.1.4 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]
refparam2 = paralist[2]
```

## 1.3.8 Calling Web Service

### Description

To call Web services, Appeon PowerServer 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.

### **1.3.8.1 `appeonwebservicecomponent` object**

#### **1.3.8.1.1 Properties**

Properties for `appeonwebservicecomponent`.

Properties	Type	Description
<code>CallType</code>	<code>String</code>	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. PowerServer for J2EE editions support only <code>CallType="1"</code> .
<code>ProxyDllOrString</code>	<code>String</code>	If <code>CallType="1"</code> , it indicates the URL of the Web service to be called, for example, <code>http://localhost/webservice.asmx</code> (the string "http://" is required); If <code>CallType="2"</code> , it indicates the DLL name of the proxy used by the Web service to be called.
<code>ClassDescriptor</code>	<code>String</code>	If <code>CallType="1"</code> , it can be null; If <code>CallType="2"</code> , it indicates the class name of the nonvisual object deployed as the Web service on the server side.
<code>ReturnValue</code>	<code>Any</code>	Read-only. The return value of functions. The value and value type varies from function to function.
<code>ErrorText</code>	<code>String</code>	Read-only. The error message of functions. The message varies from function to function. Empty string if no error.

#### **1.3.8.1.2 Functions**

##### **`of_callwebservice`**

##### **Description**

Calls the function in the binding Web service.

## Syntax

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

Argument	Description
<i>methodname</i>	The name of the Web service method.
<i>paralist</i> [ ]	Optional. Arrays of Any type. Specifies the parameter arrays for the Web service function.  Passing by reference as well as complexed data types such as array, structure etc. to <i>paralist</i> [ ] is unsupported.

## 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.proxydlorurl="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.proxydlorurl="http://localhost/webservice.asmx"
caller.classdescript=""
paralist[1]="param1"
paralist[2]="param2"
IRet=caller.of_callwebservice( "GetUserName" ,paralist)
```

### 1.3.8.1.3 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 `AppeonWebServiceComponent`, 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.

### 1.3.8.1.4 Code Examples

#### Example 1:

```
long IRet
int iResult
string strError
appeonwebservicecomponent caller

// create appeonwebservicecomponent instance and set properties
```

```

caller=create appeonwebservicecomponent

// if proxy type is DynamicProxy, the value of calltype is 1;
// if proxy type is DllProxy type, the value of calltype is 2.
// PowerServer for J2EE editions support only CallType="1".
// PowerServer 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[]
appeonwebservicecomponent caller

// create appeonwebservice instance and set properties

caller=create appeonwebservicecomponent
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
```

## 1.3.9 Appeon Text Controls

### 1.3.9.1 eon\_singlelineedit, eon\_statictext, & eon\_editmask

#### 1.3.9.1.1 Overview

Appeon PowerServer provides additional properties and events to help developers take advantage of SingleLineEdit control, StaticText control, and EditMask control:

- In order to align text centered vertically, Appeon PowerServer provides a new property (integer `ii_vertical`) and a user event (`ue_setvertical`) for the **SingleLineEdit** control, **StaticText** control, and **EditMask** control.
- To allow developers to take control of the type of the virtual keyboard on iOS and Android, Appeon PowerServer provides a new property (integer `ii_keyboardtype`) for the **EditMask** and **SingleLineEdit** control.

To use the new properties or user event, you will need to call the **eon\_singlelineeditex**, **eon\_statictextex**, or **eon\_editmaskex** objects in the Appeon Workarounds PBL.

Except for the new properties and user event, the objects have the exact same functions, events, and properties as the PowerBuilder SingleLineEdit control, StaticText control, and EditMask control.

#### 1.3.9.1.2 Properties

##### integer `ii_vertical`

Sets the text alignment. When `ii_vertical` is set to 1, text in **eon\_singlelineeditex**, **eon\_statictextex**, or **eon\_editmaskex** is centered vertically; when set to 0, text alignment is consistent with that in PowerBuilder.

##### integer `ii_keyboardtype`

This property is applicable to **eon\_editmaskex** and **eon\_singlelineeditex** on mobile platform only.

Sets the type of the keyboard that displays when **eon\_editmaskex** or **eon\_singlelineeditex** gets focus: 0 - alphabetical keyboard, 1 - numerical keyboard, 2 - URL keyboard.

2 is unsupported on Android devices, thus 0 will be used instead if 2 is set for Android devices.

#### 1.3.9.1.3 Events

##### `ue_setvertical`

###### Description

Sets the `ii_vertical` property for the object.

## Syntax

```
singlelineedit.ue_setvertical( integer ai_vertical )
statictext.ue_setvertical( integer ai_vertical )
editmask.ue_setvertical( integer ai_vertical )
```

Argument Type	Argument	Description
eon_mobile_singlelineeditex	<i>singlelineedit</i>	The name of the eon_singlelineeditex object.
eon_mobile_statictextex	<i>statictext</i>	The name of the eon_statictextex object.
eon_mobile_editmaskex	<i>editmask</i>	The name of the eon_editmaskex object.
integer	<i>ai_vertical</i>	The value of the vertical property ( <a href="#">ii_vertical</a> ) for the object.

## Return value

Integer

1 - the object is centered vertically.

0 - text alignment of the object is consistent with PowerBuilder.

## 1.3.10 Creating Data Source Dynamically

### 1.3.10.1 Overview

Data source can be predefined/modified/deleted in AEM (refer to Section 4.3.3.2, “Data Source” in *PowerServer Configuration Guide for .NET* for details), or dynamically created/modified/deleted via **appeondconfigdatasource** object in Appeon Workarounds PBLs.

But unlike creating data source in AEM, you could only dynamically create data source for the following database types: Microsoft SQL Server, Teradata, MySQL, SAP ASE, and PostgreSQL (data sources for Oracle, IBM Informix, and SAP SQL Anywhere cannot be created via **appeondconfigdatasource** object; they can only be created in AEM).

Note that you can only call **appeondconfigdatasource** object to dynamically create data source on the PowerServer for .NET version (not J2EE version).

### 1.3.10.2 **appeondconfigdatasource**

#### 1.3.10.2.1 **of\_adddatasource** function

##### Description

Create a data source with the specified parameters.

##### Syntax

```
of_adddatasource( eon_str_connection astr_connection )
```

Argument	Description
<i>astr_connection</i>	The parameters for the data source. Refer to <a href="#">eon_str_connection</a> structure for the parameter list.

**Return value**

Integer

0 - Success.

-1 - The parameters are incorrect. Failed to create the data source.

1 - The data source name already exists.

2 - The database type is not supported.

3 - The database connection failed.

**1.3.10.2.2 of\_deletedatasource function****Description**

Delete the specified data source.

**Syntax**`of_deletedatasource ( string as_datasourcename ) returns integer`

Argument	Description
<i>as_datasourcename</i>	The name of the data source to be deleted.

**Return value**

Integer

0 - Success.

-1 - The parameters are incorrect. Failed to delete the data source.

1 - The data source name does not exist. Failed to delete the data source.

**1.3.10.2.3 of\_editdatasource function****Description**

Modify the parameters of the specified data source.

**Syntax**`of_editdatasource ( eon_str_connection astr_connection )`

Argument	Description
<i>astr_connection</i>	The parameters of the specified data source. Refer to <a href="#">eon_str_connection</a> structure for the parameter list.

**Return value**

Integer

0 - Success.

- 1 - The parameters are incorrect. Failed to modify the data source.
- 1 - The data source name does not exist.
- 2 - The database type is not supported.
- 3 - The database connection failed.

#### **1.3.10.2.4 eon\_str\_connection structure**

##### **Description**

Struct.

The parameters of the data source to be created/modified.

Type	Variable Name	Description
string	<i>s_dbtype</i>	The database type of the data source. Supported values are: sqlserver, teradata, mysql, ase, postgresql, db2
string	<i>s_datasource</i>	The name of the data source to be created/modified.
string	<i>s_server</i>	The machine name or IP address of the database server.
string	<i>s_port</i>	The port number of the database server.
string	<i>s_database</i>	The database name.
string	<i>s_username</i>	The user name of the database.
string	<i>s_password</i>	The password of the database.
string	<i>s_maxpool</i>	The maximum number of connections PowerServer opens and pools on startup.
string	<i>s_minpool</i>	The minimum number of connections PowerServer opens and pools on startup.
string	<i>s_pooling</i>	Whether to use the connection pool. Supported values are: true, false.
string	<i>s_connectio</i>	The timeout period (in seconds) for the connection.
string	<i>s_commandti</i>	The timeout period (in seconds) for the commands.
string	<i>s_us</i>	Whether to enable dynamic database connection. Supported values are: true, false.

#### **1.3.11 Accessing JSON-format Data**

##### **1.3.11.1 Overview**

[JSON](#) is a light-weight data format for transmitting data in JavaScript and many other programming languages. Appeon PowerServer provides two objects (`eon_cjsonnode` & `eon_cjsonnodearray`) to help developers easily handle the JSON-format data. These two objects are supported in PowerServer Web and PowerServer Mobile.

- `eon_cjsonnode` is for manipulating the JSON node (a collection of name/value pairs), including parsing names/values from a string, adding/removing name/value pairs,

converting the node to a string etc. A JSON node can contain name/value pairs, JSON nodes, and JSON arrays.

A JSON node begins with left brace ("{" and ends with right brace ("}"), and each name is followed by colon (":") and the name/value pairs are separated by comma (","), for example:

```
{
  "firstname": "Brett",
  "lastname": "Zhang",
  "email": "brett.zhang@appeon.com"
}
```

- `eon_cjsonnodearray` is for manipulating the array values (an ordered list of values) of the JSON node, including parsing values from a string, adding/removing values, converting the array to a string etc. A JSON array can contain values, JSON nodes, and JSON arrays.

A JSON array begins with left bracket ("[") and ends with right bracket ("]"), and values are separated by comma (","), for example:

```
{
  "people": [
    {"firstname": "Brett", "lastname": "Zhang", "email": "brett.zhang@appeon.com"},
    {"firstname": "Jason", "lastname": "Li", "email": "jason.li@appeon.com"},
    {"firstname": "Mary", "lastname": "Wang", "email": "mary.wang@appeon.com"}
  ]
}
```

### 1.3.11.2 `eon_cjsonnode`

#### 1.3.11.2.1 `of_addkey`

##### Description

Adds a key (name/value pair) under the current JSON node.

##### Syntax

```
of_addkey ( string as_keyname, any aa_keyvalue )
```

Argument	Description
<i>as_keyname</i>	Name of the new key. When the name already exists, it will be overwritten.
<i>aa_keyvalue</i>	Value of the new key. Supported data type of the value: long, boolean, string, double, <code>eon_cjsonnode</code> , & <code>eon_cjsonnodearray</code>

##### Return value

None

#### 1.3.11.2.2 `of_clearnode`

##### Description

Clears all of the keys (name/value pairs) under the current JSON node.

##### Syntax

`of_clearnode()`

### Return value

None

#### 1.3.11.2.3 `of_deletekey`

### Description

Removes a key (name/value pair) from the current JSON node.

### Syntax

`of_deletekey( string as_keyname )`

Argument	Description
<i>as_keyname</i>	Name of the key to be deleted.

### Return value

None

#### 1.3.11.2.4 `of_getkeybyindex`

### Description

Gets the name of the key specified by the index.

### Syntax

`of_getkeybyindex( integer ai_index )`

Argument	Description
<i>ai_index</i>	Value of the index pointing to a key.

### Return value

String. The name of the key specified by the index.

Null if the key does not exist.

#### 1.3.11.2.5 `of_gettimebyindex`

### Description

Gets the data type of the value specified by the index.

### Syntax

`of_gettimebyindex( integer ai_index )`

Argument	Description
<i>ai_index</i>	Value of the index pointing to a key.

### Return value

Integer. Data type of the value specified by the index:

- 0 - null
- 1 - string
- 2 - number (including long and double)
- 3 - boolean
- 4 - eon\_cjsonnodearray
- 5 - eon\_cjsonnode
- 1 - invalid value

### **1.3.11.2.6 of\_gettypebykey**

#### **Description**

Gets the data type of the value for the specified key name.

#### **Syntax**

`of_gettypebykey ( string as_keyname )`

Argument	Description
<code>as_keyname</code>	Name of the key.

#### **Return value**

Integer. Data type of the value.

- 0 - null
- 1 - string
- 2 - number (including long and double)
- 3 - boolean
- 4 - eon\_cjsonnodearray
- 5 - eon\_cjsonnode
- 1 - invalid value

### **1.3.11.2.7 of\_load**

#### **Description**

Loads the JSON-format string into the current JSON node.

#### **Syntax**

`of_load ( string as_jsonstring )`

Argument	Description
<code>as_jsonstring</code>	The JSON-format string. Make sure the string complies with the JSON and JavaScript language standards:

Argument	Description
	<ul style="list-style-type: none"> <li>• The key name is case sensitive.</li> <li>• The key name must be included in double quotation marks.</li> <li>• The string value must be included in double quotation marks.</li> <li>• The boolean values (true/false) must be in lower case.</li> <li>• The boolean value, double value, and long value need not be included in double quotation marks.</li> </ul>

**Return value**

Boolean

True - Success.

False - Failed to load.

**1.3.11.2.8 of\_setvalue****Description**

Sets the value for the specified key.

**Syntax**`of_setvalue ( string as_keyname, any aa_keyvalue )`

Argument	Description
<i>as_keyname</i>	Name of the key.
<i>aa_keyvalue</i>	Value of the key. Supported data types include: long, boolean, string, double, eon_cjsonnode, & eon_cjsonnodearray

**Return value**

None

**1.3.11.2.9 of\_size****Description**

Gets the total number of the keys (name/value pairs) under the current JSON node.

**Syntax**`of_size ()`**Return value**

Integer. The total amount of the keys.

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

### 1.3.11.2.10 of\_torawstring

#### Description

Converts the keys (name/value pairs) under the current node to a special JSON-format string. Different from the standard format, the key value in the special format will not be included in double quotation marks automatically. The developer use this function according to the need of excluding double quotes in a key value, for example, to generate the proper date format in a JSON string.

#### Syntax

```
of_torawstring()
```

#### Return value

String. The JSON-format string.

### 1.3.11.2.11 of\_tostring

#### Description

Converts all of the keys (name/value pairs) under the current node to a standard JSON-format string.

#### Syntax

```
of_tostring()
```

#### Return value

String. The JSON-format string.

### 1.3.11.2.12 of\_valuearray

#### Description

Gets the value (array) of the specified key.

#### Syntax

```
of_valuearray( string as_keyname )
```

Argument	Description
as_keyname	Name of the key.

#### Return value

eon\_cjsonnodearray. The array value (eon\_cjsonnodearray object) of the specified key.

Null if the specified key does not exist or the value is not an array.

### 1.3.11.2.13 of\_valueboolean

#### Description

Gets the boolean value of the specified key.

#### Syntax

```
of_valueboolean( string as_keyname )
```

Argument	Description
<code>as_keyname</code>	Name of the key.

**Return value**

Boolean. The value of the specified key.

Null if the specified key does not exist or the value is not a boolean.

**1.3.11.2.14 of\_valuedouble****Description**

Gets the double value of the specified key.

**Syntax**

```
of_valuedouble ( string as_keyname )
```

Argument	Description
<code>as_keyname</code>	Name of the key.

**Return value**

Double. The value of the specified key.

Null if the specified key does not exist or the value is not a double.

**1.3.11.2.15 of\_valuelong****Description**

Gets the long value of the specified key.

**Syntax**

```
of_valuelong ( string as_keyname )
```

Argument	Description
<code>as_keyname</code>	Name of the key.

**Return value**

Long. The value of the specified key.

Null if the specified key does not exist or the value is not a long.

**1.3.11.2.16 of\_valuenode****Description**

Gets the value (eon\_cjsonnode object) of the specified key.

**Syntax**

```
of_valuenode ( string as_keyname )
```

Argument	Description
<code>as_keyname</code>	Name of the key.

### Return value

`eon_cjsonnode`. The value (`eon_cjsonnode` object) of the specified key.

Null if the specified key does not exist or the value is not an `eon_cjsonnode` object.

### 1.3.11.2.17 of\_valuestring

#### Description

Gets the string value of the specified key.

#### Syntax

```
of_valuestring ( string as_keyname )
```

Argument	Description
<code>as_keyname</code>	Name of the key.

### Return value

String. The value of the specified key.

Null if the specified key does not exist or the value is not a string.

### 1.3.11.2.18 Code examples

To add a JSON node with name/value pairs and convert the node to this string:

```
{"type":"work", "value":"212-555-1234", "pref":false}
```

```
eon_cjsonnode node2
node2 = Create eon_cjsonnode
node2.of_addkey ("type", "work")
node2.of_addkey ("value", "212-555-1234")
node2.of_addkey ("pref", false)
node2.of_tostring () //exports this node to a string
```

### 1.3.11.3 eon\_cjsonnodearray

#### 1.3.11.3.1 of\_append

#### Description

Adds a value or an `eon_cjsonnode` object or an `eon_cjsonnodearray` object at the end of the array.

#### Syntax 1

```
of_append ( any aa_keyvalue )
```

Argument	Description
<code>aa_keyvalue</code>	Value to be added.

Argument	Description
	Supported data types include long, boolean, string, double, eon_cjsonnode, & eon_cjsonnodearray.

**Syntax 2**

```
of_append ( eon_cjsonnode anvo_node )
```

Argument	Description
<i>anvo_node</i>	The eon_cjsonnode object to be added.

**Syntax 3**

```
of_append ( eon_cjsonnodearray anvo_nodearray )
```

Argument	Description
<i>anvo_nodearray</i>	The eon_cjsonnodearray object to be added.

**Return value**

None

**1.3.11.3.2 of\_cleararray****Description**

Clears all of the values of the current array.

**Syntax**

```
of_cleararray ()
```

**Return value**

None

**1.3.11.3.3 of\_getarraynodebyindex****Description**

Gets the value (eon\_cjsonnodearray object) specified by the index.

**Syntax**

```
of_getarraynodebyindex ( integer ai_index )
```

Argument	Description
<i>ai_index</i>	Value of the index.

**Return value**

eon\_cjsonnodearray.

Null if the value is not an eon\_cjsonnodearray object.

### 1.3.11.3.4 of\_getboolbyindex

#### Description

Gets the boolean value specified by the index.

#### Syntax

```
of_getboolbyindex( integer ai_index )
```

Argument	Description
<i>ai_index</i>	Value of the index.

#### Return value

Boolean.

Null if the value is not a boolean.

### 1.3.11.3.5 of\_getdoublebyindex

#### Description

Gets the double value specified by the index.

#### Syntax

```
of_getdoublebyindex( integer ai_index )
```

Argument	Description
<i>ai_index</i>	Value of the index.

#### Return value

Double

Null if the value is not a double.

### 1.3.11.3.6 of\_getlongbyindex

#### Description

Gets the long value specified by the index.

#### Syntax

```
of_getlongbyindex( integer ai_index )
```

Argument	Description
<i>astr_connection</i>	Value of the index.

#### Return value

Long

Null if the value is not a long.

### 1.3.11.3.7 of\_getnodebyindex

#### Description

Gets the value (eon\_cjsonnode object) specified by the index.

#### Syntax

```
of_getnodebyindex( integer ai_index )
```

Argument	Description
<i>ai_index</i>	Value of the index.

#### Return value

eon\_cjsonnode.

Null if the value is not an eon\_cjsonnode object.

### 1.3.11.3.8 of\_getstringbyindex

#### Description

Gets the string value specified by the index.

#### Syntax

```
of_getstringbyindex( integer ai_index )
```

Argument	Description
<i>ai_index</i>	Value of the index.

#### Return value

String

Null if the value is not a string.

### 1.3.11.3.9 of\_gettimebyindex

#### Description

Gets the data type of the value specified by the index.

#### Syntax

```
of_gettimebyindex( integer ai_index )
```

Argument	Description
<i>ai_index</i>	Value of the index.

#### Return value

Integer. Data type of the value:

0 - null

1 - string

2 - number

3 - boolean (including long and double)

4 - eon\_cjsonnodearray

5 - eon\_cjsonnode

-1 - Invalid value

### 1.3.11.3.10 of\_insertatindex

#### Description

Inserts a value or an eon\_cjsonnode object or an eon\_cjsonnodearray object at the position specified by the index.

#### Syntax 1

`of_insertatindex ( integer ai_index, any aa_keyvalue )`

Argument	Description
<i>ai_index</i>	Value of the index.
<i>aa_keyvalue</i>	The key value to be inserted. Support data types include long, boolean, string, double, eon_cjsonnode, & eon_cjsonnodearray.

#### Syntax 2

`of_insertatindex ( integer ai_index, eon_cjsonnode anvo_node )`

Argument	Description
<i>ai_index</i>	Value of the index.
<i>anvo_node</i>	The eon_cjsonnode object to be inserted.

#### Syntax 3

`of_insertatindex ( integer ai_index, eon_cjsonnodearray anvo_nodearray )`

Argument	Description
<i>ai_index</i>	Value of the index.
<i>anvo_nodearray</i>	The eon_cjsonnodearray object to be inserted.

#### Return value

None

### 1.3.11.3.11 of\_load

#### Description

Loads the JSON-format string into the current array.

#### Syntax

`of_load( string as_jsonstring )`

Argument	Description
<i>as_jsonstring</i>	The JSON-format string. Make sure the string complies with the JSON and JavaScript language standards.

### Return value

Boolean

True - Success.

False - Failed to load.

### 1.3.11.3.12 of\_removebyindex

#### Description

Removes the value specified by the index.

#### Syntax

`of_removebyindex( integer ai_index )`

Argument	Description
<i>ai_index</i>	Value of the index.

### Return value

None

### 1.3.11.3.13 of\_size

#### Description

Gets the total amount of the values in the array.

#### Syntax

`of_size()`

### Return value

Integer. The total amount of values in the array.

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

### 1.3.11.3.14 of\_tostring

#### Description

Converts the array to the standard JSON-format string.

#### Syntax

`of_tostring()`

### Return value

String. The standard JSON-format string.

### 1.3.11.3.15 Code examples

To add a JSON array with string values and convert the array to this string:

```
["displayName","organizations"]
```

```
eon_cjsonnodearray  array1
array1 = create  eon_cjsonnodearray
array1.of_append ( "displayName" )
array1.of_append ( "organizations" )
array1.of_tostring () //exports this array to a string
```

To add a JSON array with values of various types and convert the array to this string: [{"type":"work", "value":"212-555-1234", "pref":false}, {"type":"mobile", "value":"917-555-5432", "pref":true}, {"type":"home", "value":"203-555-7890", "pref":false}]

```
eon_cjsonnodearray  array2
array2 = create eon_cjsonnodearray
eon_cjsonnode node2
node2 = Create eon_cjsonnode
node2.of_addkey ( "type", "work" )
node2.of_addkey ( "value", "212-555-1234" )
node2.of_addkey ( "pref", false )
array2.of_append ( node2 )
node2.of_setvalue ( "type", "mobile" )
node2.of_setvalue ( "value", "917-555-5432" )
node2.of_setvalue ( "pref", true )
array2.of_append ( node2 )
node2.of_setvalue ( "type", "home" )
node2.of_setvalue ( "value", "203-555-7890" )
node2.of_setvalue ( "pref", false )
array2.of_append ( node2 )
array2.of_toString() //exports this array to a string
```

## 1.3.12 Appeon Resize Object

The Appeon resize object helps developers easily resize UI elements to fit with multiple screens, especially for the mobile apps.

### 1.3.12.1 eon\_appeon\_resize object

The eon\_appeon\_resize object provides the most capacity and flexibility because it can not only resize the window and the user object container, but also the controls and the font, at the same time, it also allows you to dynamically change the resize behavior for a particular window/user object/control at runtime.

#### 1.3.12.1.1 Functions

##### **of\_fontresize**

##### **Description**

Determines whether the font is resized proportionally with the control.

##### **Syntax**

```
of_fontresize ( boolean ab_resize, integer ai_type )
```

##### **Parameter**

*ab\_resize* - When true, the font is automatically resized; when false, the font is not resized.

*ai\_type* - 1 - Resizes with the height; 2 - Resizes with the width; 3 - Resizes with the height or width whichever has the smaller change in proportion; 4 - Resizes with the height or width whichever has the bigger change in proportion.

### Return value

None

## **of\_getscale (internal function)**

### Description

This is an internal function that will be automatically called when resize script is executed.

Gets the default ratio of each control.

### Syntax

```
of_getscale ( string as_input, ref string as_flag, ref string as_xscale, ref string  
as_yscale, ref string as_wscale, ref string as_hscale )  
  
of_getscale ( string as_input, ref string as_flag, ref string as_xscale, ref string  
as_yscale, ref string as_wscale, ref string as_hscale, ref string as_width, ref  
string as_height )  
  
of_getscale ( string as_input, ref string as_flag, ref string as_xscale, ref string  
as_yscale, ref string as_wscale, ref string as_hscale, ref string as_width, ref  
string as_height, ref string as_fontsize )
```

### Return value

None

## **of\_gettag**

### Description

Gets the original tag of a control or object. Original tag includes the information that is not altered by the of\_init function.

### Syntax

```
of_gettag ( windowobject awo )
```

### Parameter

*awo* - The name of the object or control.

### Return value

String. The original tag information of the control or object.

## **of\_init**

### Description

Initializes the user object container or the window that needs to be resized. You can initialize the user object container or the window only; you can also initialize the controls contained in

the user object container or the window by setting the `ab_recursion` parameter to true. Only the initialized user object/window/control can be resized, therefore, this function must be the very first one to be called.

By initializing the user object/window/control, it actually registers the following information to the Tag property of the user object/window/control, separated by "/":

`register flag/resize flag/x ratio/y ratio/width ratio/height  
ratio/width/height/x/y/parentwidth/parentheight/{textsize}/original  
tag`

`flag`: used by the developer to manually specify the resize behavior of a control or object, such as scaletobottom, fixtright etc. By default, the value is "1111". It can be set by using the [of\\_setflag](#) function.

`textsize`: only effective when the control or object has the textsize property.

**Syntax 1: Initializes the user object. If `ab_recursion` is true, all of the controls contained in the user object will be initialized as well.**

`of_init ( userobject au_uo, boolean ab_recursion )`

**Syntax 2: Initializes the user object. If `ab_recursion` is true, all of the controls contained in the user object will be initialized as well.**

`of_init ( userobject au_uo, integer ai_width, integer ai_height, boolean ab_recursion )`

**Syntax 3: Initializes the window and the controls in it. If `ab_recursion` is true, all of the user objects and tab controls contained in the window will be initialized as well.**

`of_init ( window aw_window, boolean ab_recursion )`

## Parameter

`au_uo` - The name of the user object container to be initialized.

`aw_window` - The name of the window container to be initialized, and the controls in the window will be automatically initialized as well.

`ai_width` - The new width of the user object container. The original proportion of each control will be calculated based on it.

`ai_height` - The new height of the user object container. The original proportion of each control will be calculated based on it.

`ab_recursion` - When true, the controls contained in the user object container will be initialized as well; or the user objects and tab controls in the window container will be initialized as well; when false, only the user object container or the window (and controls in the window) will be initialized.

## Return value

None

## **of\_init\_object**

### Description

Changes the ratio of a single object or control that has been registered using `of_init`.

## Syntax

```
of_init_object ( windowobject awo, decimal ad_xscale, decimal ad_yscale,  
decimal ad_wscale, decimal ad_hscale )  
of_init_object ( windowobject awo, integer ai_parentwidth, integer  
ai_parentheight, boolean ab_recursion )  
of_init_object ( windowobject awo, boolean ab_recursion )
```

Comparatively speaking, the second syntax is easier to use and produce the desired layout; the first syntax would require more testing to determine an accurate x/y ratio.

The third syntax is for refreshing the object or control easily without needing a lot arguments.

## Parameter

*awo* - The name of the object or control to be initialized.

*ad\_xscale* - The new x ratio.

*ad\_yscale* - The new y ratio.

*ad\_wscale* - The new width ratio.

*ad\_hscale* - The new height ratio.

*ai\_parentwidth* - The new width of the parent container.

*ai\_parentheight* - The new height of the parent container.

*ab\_recursion* - When true, the controls contained in the object or the control will be initialized as well; when false, only the object or the control will be initialized.

## Return value

None

## of\_internal\_resize (internal function)

### Description

This is an internal function that will be automatically called when resize script is executed.

Resizes the DragObject by utilizing its flag.

## Syntax

```
of_internal_resize ( ref dragobject awo, string as_flag, integer ai_newwidth,  
integer ai_newheight, string as_xbl, string as_ybl, string as_wbl, string as_hbl )
```

## Parameter

*awo* - The name of the DragObject.

*as\_flag* - The flag of the DragObject.

*ai\_newwidth* - The new width of the DragObject.

*ai\_newheight* - The new height of the DragObject.

*as\_xbl* - The x ratio of the DragObject.

*as\_ybl* - The y ratio of the DragObject.

*as\_wbl* - The width ratio of the DragObject.

*as\_hbl* - The height ratio of the DragObject.

### Return value

None.

## **of\_orginterval\_h (internal function)**

### Description

This is an internal function that will be automatically called when resize script is executed.

Gets the default height of a control.

### Syntax

```
of_orginterval_h ( windowobject awo )
```

### Parameter

*awo* - The name of the object or control.

### Return value

Long. The default height of the control.

## **of\_orginterval\_w (internal function)**

### Description

This is an internal function that will be automatically called when resize script is executed.

Gets the default width of a control.

### Syntax

```
of_orginterval_w ( windowobject awo )
```

### Parameter

*awo* - The name of the object or control.

### Return value

Long. The default width of the control.

## **of\_orginterval\_x (internal function)**

### Description

This is an internal function that will be automatically called when resize script is executed.

Gets the default x coordinate of a control.

### Syntax

```
of_orginterval_x ( windowobject awo )
```

### Parameter

*awo* - The name of the object or control.

### Return value

Long. The default x coordinate of the control.

**of\_orginterval\_y (internal function)****Description**

This is an internal function that will be automatically called when resize script is executed.

Gets the default y coordinate of a control.

**Syntax**

```
of_orginterval_y( windowobject awo )
```

**Parameter**

*awo* - The name of the object or control.

**Return value**

Long. The default y coordinate of the control.

**of\_pause****Description**

Pauses the auto-resizing service for all controls. It is used in pair with of\_recover.

**Syntax**

```
of_pause()
```

**Return value**

None.

**of\_recover****Description**

Resumes the auto-resizing service for all controls. It is used in pair with of\_pause.

**Syntax**

```
of_recover()
```

**Return value**

None.

**of\_refresh\_object****Description**

Refreshes the properties of a single object or control that has been registered using of\_init. The object or control's parent will not be refreshed by this function.

**Syntax**

```
of_refresh_object( dragobject ado )
```

**Parameter**

*ado* - The name of the object or control whose size has been changed.

**Return value**

None.

**of\_registered (internal function)****Description**

This is an internal function that will be automatically called when resize script is executed.

Checks whether a control has been registered-initialized.

**Syntax**

```
of_registered( windowobject awo )
```

**Parameter**

*awo* - The name of the object or control.

**Return value**

Boolean.

**of\_resizable (internal function)****Description**

This is an internal function that will be automatically called when resize script is executed.

Checks whether a control is resizable.

**Syntax**

```
of_resizable( windowobject awo )
```

**Parameter**

*awo* - The name of the object or control.

**Return value**

Boolean.

**of\_resize****Description**

Resizes a control according to the registered information.

**Syntax 1: Resizes the user object.**

```
of_resize( userobject au_uo, integer ai_newwidth, integer ai_newheight,  
boolean ab_recursion )
```

**Syntax 2: Resizes the window and the controls in the window. If *ab\_recursion* is true, objects in the user object and tab controls will be resized as well.**

```
of_resize( window aw_window, integer ai_newwidth, integer ai_newheight,  
boolean ab_recursion )
```

**Syntax 3: Resizes the DataWindow control. Normally the DataWindow control will be resized with its parent container, therefore, it is not recommended to call this function to resize it.**

```
of_resize( datawindow adw, integer ai_newwidth, integer ai_newheight,  
boolean ab_zoom, integer ai_type, decimal ad_maxrate )
```

**Syntax 4: Resizes the DataWindow control. Normally the DataWindow control will be resized with its parent container, therefore, it is not recommended to call this function to resize it.**

```
of_resize ( datawindow adw, integer ai_newwidth, integer ai_newheight,  
decimal ad_maxrate )
```

### Parameter

*au\_uo* - The name of the user object container to be resized.

*aw\_window* - The name of the window to be resized.

*adw* - The name of the DataWindow to be resized.

*ai\_newwidth* - The new width of the user object container, the window, or the DataWindow.

*ai\_newheight* - The new height of the user object container, the window, or the DataWindow.

*ab\_zoom* - Sets whether the DataWindow control is resizable.

*ai\_type* - The type of resizing. See *of\_zoom* for more.

*ad\_maxrate* - The maximum resize ratio.

*ab\_recursion* - When true, the controls contained in the user object container or the window will be resized as well; when false, only the user object container or the window will be resized.

### Return value

None.

## of\_setflag

### Description

Sets the flag of a control or object, so that the control or object will be resized differently.

### Syntax

```
of_setflag ( windowobject awo, string as_flag )
```

### Parameter

*awo* - The name of the object or control.

*as\_flag* - The flag of the object or control. The value of flag is composed of 4 digits, by default it is "1111".

The first digit indicates the resize behavior of the x coordinate: 0 - Keeps unchanged; 1 - Changes proportionally; 2 - Moves to right horizontally.

The second digit indicates the resize behavior of the y coordinate: 0 - Keeps unchanged; 1 - Changes proportionally; 2 - Moves down vertically.

The third digit indicates the resize behavior of the width: 0 - Keeps unchanged; 1 - Changes proportionally; 2 - Extends to right horizontally; 3 - Extends to left horizontally; 4 - Extends to right edge; 5 - Extends to left edge.

The fourth digit indicates the resize behavior of the height: 0 - Keeps unchanged; 1 - Changes proportionally; 2 - Extends down vertically; 3 - Extends up vertically; 4 - Extends to the bottom; 5 - Extends to the top.

### **Return value**

Boolean.

#### **of\_start**

##### **Description**

Restarts the auto-resizing service for a control or object. It is used in pair with of\_stop.

##### **Syntax**

```
of_start ( windowobject awo )
```

##### **Parameter**

*awo* - The name of the object or control.

##### **Return value**

None.

#### **of\_stop**

##### **Description**

Stops the auto-resizing service for a control or object. It is used in pair with of\_start.

##### **Syntax**

```
of_stop ( windowobject awo )
```

##### **Parameter**

*awo* - The name of the object or control.

##### **Return value**

None.

#### **of\_stringtoarray (internal function)**

##### **Description**

This is an internal function that will be automatically called when resize script is executed.

Splits a string into an array.

##### **Syntax**

```
of_stringtoarray ( string as_input, ref string as_array[ ], string as_flag )
```

##### **Return value**

None.

#### **of\_typeof (internal function)**

##### **Description**

This is an internal function that will be automatically called when resize script is executed.

Checks the type of a control or an object.

### Syntax

```
of_typeof ( windowobject awo_control )
```

### Return value

String. The type of the control or the object.

### of\_zoom

#### Description

Determines whether to turn on or off the resizing feature for the DataWindow control.

### Syntax

```
of_zoom ( boolean ab_zoom, integer ai_type )
```

```
of_zoom ( boolean ab_zoom, integer ai_type, decimal adec_rate )
```

### Parameter

*ab\_zoom* - When true, resizing feature for the DataWindow control is enabled; when false, resizing feature is disabled.

*ai\_type* - 1 - Resizes proportionally with height; 2 - Resizes proportionally with width; 3 - Resizes with height or width whichever has the smaller change in proportion; 4: Resizes with height or width whichever has the bigger change in proportion.

*adec\_rate* - The maximum resizing rate of the DataWindow control. By default, it is 3, which means the DataWindow control can be resized by no more than 300%.

### Return value

None.

#### 1.3.12.1.2 Code example

Below is the sample code for resizing a user object. The code for resizing a window is similar.

```
//declears the instance variable for the user object
eon_appeon_resize  ieon_resize

//initializes in the constructor event of the user object
//(if it is window, initializes in the open event of the window)
ieon_resize = create eon_appeon_resize
// registers all of the controls contained in this user object
ieon_resize.of_init( this, true )
//enables font resizing and resizes with the width
ieon_resize.of_fontresize( true, 2 )
//enables datawindow resizing and resizes with the width
ieon_resize.of_zoom( true, 2 )
//changes the datawindow resize behavior by keeping x and y coordinates unchanged,
//and scaling to the right edge and to the bottom of the user object container.
ieon_resize.of_setflag( dw_1, '0044' )
//keeps the cb_1 size unchanged, and also keeps its relative position unchanged
//which means the distances from the right edge and from the bottom remain the
//same.
ieon_resize.of_setflag( cb_1, '2222' )
```

```
//defines the resize event for the user object, event ID is set to pbm_size
//Resizes all of the controls contained in the user object
ieon_resize.of_resize( this, newwidth, newheight, true)

//destroys the user object in the destructor event
Destroy ieon_resize
```

## 1.4 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.

### 1.4.1 Audio

Plays the audio file using the device's audio player.

If more than one audio object instance is created and their of\_play function is called, then more than one audio file is being played at the same time. This could happen if the window is closed without destroying the audio instance (so the audio is still being played), and then when the window is opened again, the audio instance is created again.

Supported on iOS device only.

#### 1.4.1.1 eon\_mobile\_audioex

##### 1.4.1.1.1 Properties

###### **integer ii\_errorcode**

Error code which is returned when the [oe\\_error event](#) is triggered.

###### **powerobject ipo\_bindevent**

The object to bind with the [oe\\_error event](#).

###### **string is\_bindwitherrorevent**

The name of the event bound with the [oe\\_error event](#) of the powerobject ipo\_bindevent.

###### **string is\_errortext**

Error message which is returned when the [oe\\_error event](#) is triggered.

#### 1.4.1.1.2 Events

##### **oe\_error**

###### **Description**

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

Supported on iOS device only.

Note: This event will update the value of [is\\_errortext](#) and [ii\\_errorcode](#) properties.

### Syntax

None.

### Return value

None.

#### 1.4.1.1.3 Functions

##### **of\_getstatus**

###### Description

Gets the current status of the audio file.

Supported on iOS device only.

### Syntax

```
audio.of_getstatus()
```

Argument Type	Argument	Description
eon_mobile_audioex	<i>audio</i>	The name of the eon_mobile_audioex object.

### Return value

Integer.

0 - The audio file is stopped.

1 - The audio file is being played.

2 - The audio file is paused.

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

##### **of\_pauseorresume**

###### Description

Pauses the audio file that is being played or resumes playing it from where it stopped.

Supported on iOS device only.

### Syntax

```
audio.of_pauseorresume()
```

Argument Type	Argument	Description
eon_mobile_audioex	<i>audio</i>	The name of the eon_mobile_audioex object.

### Return value

Integer.

1 - The audio file is paused successfully.

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

### **of\_play**

#### **Description**

Uses the default audio player application to play the audio file.

The supported audio file formats are .aac, .mp3, .aiff, and .wav. Not all the audio files with these suffixes can be played; if some media files cannot be played, they should be transcoded first via iTunes.

Supported on iOS device only.

#### **Syntax**

`audio.of_play( string as_filepath )`

Argument Type	Argument	Description
eon_mobile_audioex	<i>audio</i>	The name of the eon_mobile_audioex object.
string	<i>as_filepath</i>	The full path of the audio file to be played.

#### **Return value**

Integer.

1 - The audio file is played successfully.

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

### **of\_register**

#### **Description**

Registers the object and the event to be bound with the oe\_error event.

Supported on iOS device only.

#### **Syntax**

`audio.of_register( powerobject apb_bind, string as_error )`

Argument Type	Argument	Description
eon_mobile_audioex	<i>audio</i>	The name of the eon_mobile_audioex object.
powerobject	<i>apb_bind</i>	The object to be bound with the <a href="#">oe_error event</a> .
string	<i>as_error</i>	The event to be bound with the <a href="#">oe_error event</a> .

#### **Return value**

Integer.

1 - Success.

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

### **of\_stop**

#### **Description**

Stops the audio file that is being played.

Supported on iOS device only.

#### **Syntax**

*audio.of\_stop()*

Argument Type	Argument	Description
eon_mobile_audioex	<i>audio</i>	The name of the eon_mobile_audioex object.

#### **Return value**

Integer.

1 - The audio file is stopped successfully.

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

### **of\_triggerevent**

#### **Description**

Triggers the object event bound with the oe\_error event.

Supported on iOS device only.

#### **Syntax**

*audio.of\_triggerevent ( string as\_event )*

Argument Type	Argument	Description
eon_mobile_audioex	<i>audio</i>	The name of the eon_mobile_audioex object.
string	<i>as_event</i>	The event name bound with <a href="#">oe_error event</a> .

#### **Return value**

None

## **1.4.2 Appeon Workspace**

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

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

### **1.4.2.1 eon\_mobile\_awsex object**

#### **1.4.2.1.1 Properties**

*powerobject ipo\_bindevent*

PowerBuilder object bound with the [oe\\_urlschemesucceed](#) event.

*string is\_bindwithsucceedevent*

The name of the event bound with the [oe\\_urlschemesucceed](#) event.

#### **1.4.2.1.2 Events**

##### **oe\_urlschemesucceed**

###### **Description**

This event is triggered when Appeon Workspace is launched by other PowerServer mobile apps or third-party apps via the URL scheme.

This event can be bound to an object and event using the [of\\_register](#) and [of\\_triggerevent](#) function.

Supported on mobile client only.

###### **Syntax**

`aws.oe_urlschemesucceed ( string as_urlscheme )`

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.
string	<i>as_urlscheme</i>	Receives the value of the URL scheme parameter passed by the other PowerServer mobile app or third-party app.

###### **Return value**

None.

#### **1.4.2.1.3 Functions**

##### **of\_checkpermissionforinet**

###### **Description**

Checks if the app in Appeon Workspace has permissions to access the .NET IIS server.

Supported on mobile client only.

## Syntax

```
aws.of_checkpermissionforinet ( string as_workspaceid, string appname )
```

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.
string	<i>as_workspaceid</i>	ID of Appeon Workspace.
string	<i>appname</i>	Name of the application installed in Appeon Workspace.

## Return value

Integer.

- 1 - It is called in PowerBuilder or PowerServer Web, or there is an error.
- 2 - Server is not .NET IIS server.
- 3 - Failed to get the application name from Appeon Workspace.
- 6 - Failed to execute AppeonDotNetComponent.of\_ExecInterface.
- 1 - The app has permissions to access the server.
- 20 - Appeon Workspace ID does not exist.
- 10 - Appeon Workspace ID exists, but is disabled.
- 11 - Appeon Workspace ID and the app is not associated.
- 30 - Other reasons.

## of\_clearlog

### Description

Clears the mobile app 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 PowerServer Web, or there is an error.

**of\_getappinfo****Description**

Gets the information (such as app name, app URL) of the current PowerServer mobile application.

Supported on mobile client only.

**Syntax**

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

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

**Return value**

Integer.

1 - Success.

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

**of\_getapporientation****Description**

Detects if the screen orientation of the PowerServer 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
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.

**Return value**

Integer.

0 - Unknown.

1 - Portrait.

2 - Landscape.

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

**of\_getapprotationlock****Description**

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

Supported on mobile client only.

**Syntax**

```
aws.of_getapprotationlock ()
```

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 PowerServer 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 PowerServer Web, or there is an error.

## **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 PowerServer 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 PowerServer Web, or there is an error.

### **of\_getdwinputmode**

#### **Description**

Obtains the display mode of the virtual keyboard on the mobile device when the user taps into the DataWindow text field. If [of\\_setdwinputmode](#) is called, then `of_getdwinputmode` will obtain the display mode set via [of\\_setdwinputmode](#). If [of\\_setdwinputmode](#) is not called, then `of_getdwinputmode` will obtain the display mode setting in Appeon Workspace.

Supported on mobile client only.

#### **Syntax**

```
aws.of_getdwinputmode ()
```

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 - Keyboard displays when the user taps once in the DataWindow field.

2 - Keyboard displays when the user taps twice in the DataWindow field.

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

#### **See also**

[of\\_setdwinputmode](#)

### **of\_getdwmousemovemode**

#### **Description**

Detects if the `pdm_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 PowerServer Web, or there is an error.

### **of\_gettitlebarvisible**

#### **Description**

Detects if the application title bar is visible or invisible. For more about the titlebar, refer to the section called “Window title bar, menu, & toolbar” in *Mobile UI Design & Development Guide*.

Supported on mobile client only.

#### **Syntax**

```
aws.of_gettitlebarvisible()
```

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

#### **Return value**

Integer.

1 - The title bar is visible.

0 - The title bar is invisible but the normal view icon is visible.

2 - Both the title bar and the normal view icon are invisible.

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

### **of\_geturlschemeparm**

#### **Description**

Gets the value of the URL scheme parameter passed by another PowerServer mobile app or a third-party app.

For more about the URL scheme, refer to URL Scheme Name in *PowerServer Toolkit User Guide*.

Supported on mobile client only.

#### **Syntax**

```
aws.of_geturlschemeparm( ref string as_urlscheme )
```

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

Argument Type	Argument	Description
ref string	<i>as_urlscheme</i>	Receives the value of the URL scheme parameter passed by the third-party app.

**Return value**

Integer.

1 - Success.

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

**of\_getversion****Description**

Gets the version number of Appeon Workspace.

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 Workspace version number. Returns empty string if it is called in PowerBuilder or PowerServer Web, or if there is any error.

**Return value**

Integer.

1 - Success.

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

**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	<i>aws</i>	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 PowerServer Web, or there is an error.

### **of\_getworkspaceheight**

#### Description

Gets the height (in PBU) of the Appeon Workspace screen. If the workspace title bar is visible, then its height is not included.

Supported on mobile client only.

#### Syntax

```
aws.of_getworkspaceheight ()
```

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

### Return value

Integer

The height of the Appeon Workspace screen.

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

### **of\_getworkspaceidappname**

#### Description

Gets the Appeon Workspace ID and the name of the application currently running in Appeon Workspace. The Appeon Workspace ID is based on the device's unique identifier which depends on the device manufacturer, product type, and operating system.

Supported on mobile client only.

#### Syntax

```
aws.of_getworkspaceidappname ()
```

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.
string	<i>as_workspaceid</i>	The Appeon Workspace ID.

Argument Type	Argument	Description
string	<i>as_appname</i>	The name of the application currently running in Appeon Workspace.

**Return value**

Integer

1 - Success.

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

**of\_getworkspacewidth****Description**

Gets the width (in PBU) of the Appeon Workspace screen.

Supported on mobile client only.

**Syntax**`aws.of_getworkspacewidth ()`

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

**Return value**

Integer

The width of the Appeon Workspace screen.

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

**of\_log****Description**

Writes the log information 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.

Argument Type	Argument	Description
		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 PowerServer Web, or there is an error.

### Usage

For the log path of mobile applications, please refer to Section 4.4.6.8, “Client Logs” in *PowerServer Configuration Guide for .NET* or PowerServer Configuration Guide for J2EE.

### **of\_register**

#### Description

Registers the object and the event to be bound with the [oe\\_urlschemesucceed](#) event.

Supported on mobile client only.

#### Syntax

```
aws.of_register ( powerobject apb_bind, string as_schemeparam )
```

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.
powerobject	<i>apb_bind</i>	The object to be bound with the <a href="#">oe_urlschemesucceed</a> event.
string	<i>as_schemeparam</i>	The event to be bound with the <a href="#">oe_urlschemesucceed</a> event.

### Return value

Integer.

1 - Success.

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

### **of\_sendmail**

#### Description

Sends an email from the mobile application.

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 <a href="#">eon_mobile_str_mailcontent</a> .
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. The following return values are effective for the iOS device only. On the Android device, 0 will be returned under all circumstances, because the corresponding mail API from the Android SDK returns no standardized value.

1 - Send the email successfully.

0 - Cancel sending the email.

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

-2 - The email account is not configured.

-100 - System error.

## Code example

Refer to [How to generate a PDF file and send it via email](#) for the code example of generating and sending a PDF file via email.

## of\_setapporientation

### Description

Sets the screen orientation of the PowerServer mobile application.

The screen orientation is set regardless of the current screen rotation.

After this function is called successfully to set the screen orientation, be sure to call [of\\_setapprotationlock](#) to lock the screen orientation. And do not call [of\\_setapprotationlock](#)

before this function is called. If [of\\_setapporientationlock](#) is called before this function is called, this function will have no effect and return -1.

Supported on mobile client only.

### Syntax

```
aws.of_setapporientation ( 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 screen orientation to Portrait. 2 - Sets screen orientation to Landscape.

### Return value

Integer.

1 - Success.

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

### [of\\_setapporientationlock](#)

#### Description

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

This function must be called after [of\\_setapporientation](#) is called successfully to set the screen orientation.

Supported on mobile client only.

### Syntax

```
aws.of_setapporientationlock ( 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 PowerServer mobile application, so it will not rotate when the screen orientation of the device changes. 0 - Unlocks the screen orientation of the PowerServer mobile application.

## Return value

Integer.

1 - Success.

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

### **of\_setassistivetouchmode**

#### Description

Sets the 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.

Argument Type	Argument	Description
		2 - Sets to the Drag mode.

### Return value

Integer.

1 - Success.

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

### **of\_setdwinputmode**

#### **Description**

Sets the display mode of the virtual keyboard on the mobile device when the user taps into the DataWindow text field. You can also set the display mode for all applications installed in the same Appeon Workspace, by going to **Appeon Workspace | Settings | General**, and setting the **Show Keyboard on a Single Tap** option. For more information, refer to Section 4.5, “Showing keyboard on single tap or double taps” in *Appeon Workspace User Guide*. The **of\_setdwinputmode** function takes precedence over the setting in Appeon Workspace.

By default, the keyboard pops up when the user taps twice in the field of the DataWindow, or when the user presses the DataWindow field for as long as 1500 milliseconds (you can change this value using [of\\_setlongpressinterval](#)).

Supported on mobile client only.

#### **Syntax**

```
aws.of_setdwinputmode ( 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 – Keyboard displays when the user taps once on a DataWindow field. 2 – Keyboard displays when the user taps twice on a DataWindow field.

#### **Return value**

Integer.

1 - Success.

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

#### **Limitation**

When the user is dragging in the DataWindow control, the virtual keyboard may display unexpectedly.

When the virtual keyboard displays after the single tap, the DataWindow text field will not respond to the scroll gesture any more.

#### **See also**

[of\\_getdwinputmode](#)

**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
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.
integer	<i>ai_mode</i>	1 - Supported. 0 - Unsupported.

**Return value**

Integer.

1 - Success.

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

**of\_sethinttext****Description**

Sets the hint text for the SingleLineEdit control. The hint text property is not available in PowerBuilder SingleLineEdit control. It is specially designed for the PowerServer mobile application.

**Syntax**

```
aws.of_sethinttext ( singlelineedit sle_target, string as_hinttext )
```

Argument	Description
<i>aws</i>	The name of the eon_mobile_awsex object.
<i>sle_target</i>	The name of the eon_singlelineeditex object.
<i>as_hinttext</i>	The text that will be displayed as hint text in the SingleLineEdit control.

**Return value**

Integer.	1 - Success.  -1 - It is called in PowerBuilder or PowerServer Web, or there is an error.  -2 - The <i>as_hinttext</i> argument is null.
----------	--

**of\_sethinttextcolor****Description**

Sets the color of the hint text for the SingleLineEdit control. The hint text color property is not available in PowerBuilder SingleLineEdit control. It is specially designed for the PowerServer mobile application.

### Syntax

```
aws.of_sethinttextcolor( singlelineEdit sle_target, long al_newcolor )
```

Argument	Description
<i>aws</i>	The name of the eon_mobile_awsex object.
<i>sle_target</i>	The name of the eon_singlelineeditex object.
<i>al_newcolor</i>	The color of the hint text that will be displayed in the SingleLineEdit control.

### Return value

Integer.	1 - Success. -1 - It is called in PowerBuilder or PowerServer Web, or there is an error. -2 - The <i>al_newcolor</i> argument is null.
----------	--

### of\_setlongpressinterval

#### Description

Sets the duration (in milliseconds) for a press before it is recognized as a long press. Once this duration is reached, the keyboard will be brought up to display. By default, the keyboard pops up when the user taps twice in the field of the DataWindow, or when the user presses the DataWindow field for as long as 1500 milliseconds.

Supported on mobile client only.

### Syntax

```
aws.of_setlongpressinterval( long al_interval )
```

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.
long	<i>al_interval</i>	Sets the duration in milliseconds. If it is set to a value smaller than 500 milliseconds, then 550 milliseconds will be used, because 500 milliseconds is already used as the duration for the DataWindow drag & drop event.

### Return value

Long.

1 - Success.

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

### **of\_settitlebarvisible**

#### **Description**

Sets whether the application title bar is visible or invisible. When the title bar is set to invisible, the application will be displayed in full screen view.

Supported on mobile client only.

#### **Syntax**

```
aws.of_settitlebarvisible ( 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>	<p>1 - Sets the title bar to be visible.</p> <p>0 - Sets the title bar to be invisible, and displays the normal view icon on the top right corner of the window, when it is tapped, the app will return to the normal view (with the title bar visible again).</p> <p>2 - Sets the title bar to be invisible, and displays no icon for showing the title bar again.</p>

#### **Return value**

Integer.

1 - Success.

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

**of\_triggerevent****Description**

Triggers the object event bound with the [oe\\_urlschemesucceed](#) event.

Supported on mobile client only.

**Syntax**

```
aws.of_triggerevent ( string as_event )
```

Argument Type	Argument	Description
eon_mobile_awsex	<i>aws</i>	The name of the eon_mobile_awsex object.
string	<i>as_event</i>	The event name bound with <a href="#">oe_urlschemesucceed</a> event.

**Return value**

None.

**1.4.2.1.4 Structures****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.

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

### 1.4.3 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.

#### 1.4.3.1 eon\_mobile\_barcodeex object

##### 1.4.3.1.1 Functions

###### 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 in the application sandbox's "plugin" directory.
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	
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.
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.
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

Data	Format	Specification
		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	Supported on iOS only.  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 PowerServer 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

Blob.

**of\_read****Description**

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

The camera will be opened automatically by PowerServer, 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	
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.
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.
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.

Data	Format	Specification
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	Supported on iOS only. 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 PowerServer Web, or there is an error.

## 1.4.4 Camera

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

### 1.4.4.1 eon\_mobile\_cameraex object

#### 1.4.4.1.1 Functions

##### 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 ab1b_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>	In iOS, it returns the full path pointing to the selected file in the sandbox plugin directory. The selected video or photo file will be copied from the album to the plugin directory of the current application's sandbox. The files in this plugin directory will be removed when the application exits.  In Android, it returns the full path pointing to the selected file.
blob	<i>ab1b_data</i>	Returns the file data of the photo file or video file.

Argument Type	Argument	Description
		The file data could be saved into the database via the UpdateBlob SQL statement.

## Return value

Integer.

1 - Select a photo file successfully, and *as\_filepath* will return the full path pointing to the selected photo file.

2 - Select a video file successfully, and *as\_filepath* will return the full path pointing to the selected video file.

0 - Cancel selecting a file from the album.

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

## Code example 1

```
eon_mobile_cameraaex 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_cameraaex
li_return = ln_camera.of_openalbums (ls_filepath)
```

## Code example 2

```
eon_mobile_cameraaex 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_cameraaex
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_allowededit,
ref string as_filepath )
```

```
camera.of_takefile ( value integer ai_filetype, value boolean ab_allowededit,
ref string as_filepath, ref blob abl_data )
```

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

`camera.of_takemode ( value eon_mobile_str_cameraoption astr_option, 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 <a href="#">eon_mobile_str_cameraoption</a> .
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 or any other integer - 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 "plugin" directory.  In iOS, the video and photo files will be saved to the iOS album, as well as to the "plugin" directory of the current application's sandbox. As there is restricted access to the directory of iOS album, only the "plugin" directory can be obtained and returned. The files in this "plugin" directory will be removed when the application exits.  In Android, the video and photo files will be only saved to the "plugin" directory of the current application's sandbox. The files in the

Argument Type	Argument	Description
		"plugin" directory will remain when the application exits and will be displayed when the Android album is opened by of_openalbums.
blob	<i>ab1b_data</i>	Returns the file data of the photo file or video file. The file data could be saved into the database via the UpdateBlob SQL statement.

## 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 video-recording is cancelled, or if there is any error.

0 - Cancel taking a photo or making a video.

-1 - It is called in PowerBuilder or PowerServer 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_cameraex
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_cameraex
li_return = ln_camera.of_takefile (lstr_option, ls_filepath, lb_data)
```

### 1.4.4.1.2 Structures

#### **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_allowededit	Sets if the file is editable after the picture is taken or the video is recorded.

## 1.4.5 Connection

Obtains the network state and connection type.

### 1.4.5.1 eon\_mobile\_connectionex object

#### 1.4.5.1.1 Functions

##### 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 PowerServer Web, or there is an error.

## 1.4.6 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.

### 1.4.6.1 eon\_mobile\_deviceex object

#### 1.4.6.1.1 Functions

##### **of\_getdeviceid**

###### Description

Gets the device's unique identifier. For Android, the device ID is determined by the device manufacturer, product type, and operating system. For iOS, it is not allowed to get the device information, therefore, a random device ID will be generated and stored in the app by PowerServer when the app is running. If two Appeon Workspace are installed, they will have different device IDs. If an app is completely erased from the device (because of system formatting etc.), the app will have a different device ID when it is installed again.

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

###### Return value

Integer.

1 - Success.

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

###### Usage

Use the `of_getdeviceid()` function to 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 will not 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
eon_mobile_deviceex	<i>device</i>	The name of the eon_mobile_deviceex 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 PowerServer Web or if there is any error.

**Return value**

Integer.

1 - Success.

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

### **of\_getfreememory**

#### **Description**

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

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 PowerServer 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 PowerServer Web, or if there is any error.

### Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or PowerServer 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 PowerServer 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 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 PowerServer Web, or if there is any error.

### Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or PowerServer 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_width, ref integer ai_height )
```

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

**Return value**

Integer.

1 - Success.

-1 - It is called in PowerBuilder or PowerServer 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 PowerServer Web, or if there is any error.

## Return value

Integer.

1 - Success.

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

### 1.4.7 GPS

Opens the GPS functionality and gets the current position. You would need to make sure GPS is enabled on the mobile device, and there is good network connection.

#### 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 GPS 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.

#### 1.4.7.1 eon\_mobile\_geolocationex object

##### 1.4.7.1.1 Properties

###### **integer ii\_errorcode**

Error code which is returned when the [oe\\_error event](#) is triggered.

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 cancelled.
- 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.

**powerobject ipo\_bindevent**

The object to bind with the [oe\\_locationchanged event](#) and [oe\\_error event](#).

**string is\_bindwitherrorevent**

The name of the event bound with the [oe\\_error event](#) of the powerobject ipo\_bindevent.

**string is\_bindwithlocationchangeevent**

The name of the event bound with the [oe\\_locationchanged event](#) of the powerobject ipo\_bindevent.

**string is\_errortext**

Error message which is returned when the [oe\\_error event](#) is triggered.

#### 1.4.7.1.2 Events

**oe\_error****Description**

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

Supported on mobile client only.

Note: This event will update the value of [is\\_errortext](#) and [ii\\_errorcode](#) properties.

**Syntax**

None.

**Return value**

None.

**oe\_locationchanged****Description**

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

When the messages for this event in the message queue reaches 500, the new messages for this event will not be stored, they will be directly discarded.

## Syntax

```
oe_locationchanged ( decimal dec_longitude, decimal dec_latitude, decimal
dec_altitude, decimal dec_accuracy, decimal dec_altitudeaccuracy,
decimal dec_heading, decimal dec_speed, datetime dt_timestamp )
```

Parameter Name	Parameter Type	Description
dec_longitude	Decimal	The longitude value of the current position.
dec_latitude	Decimal	The latitude value of the current position.
dec_altitude	Decimal	The altitude value of the current position.
dec_accuracy	Decimal	The latitude and longitude positioning accuracy.
dec_altitudeaccuracy	Decimal	The altitude positioning accuracy.
dec_heading	Decimal	The degrees clockwise from true north (0 to 359.99 degrees).
dec_speed	Decimal	The displacement velocity (m/sec).
dt_timestamp	DateTime	The time stamp to get the current position information.

## Return value

None.

### 1.4.7.1.3 Functions

#### of\_close

##### Description

Closes the GPS functionality.

Supported on mobile client only.

## Syntax

```
gps.of_close()
```

Argument Type	Argument	Description
eon_mobile_geolocationex	gps	The name of the eon_mobile_geolocationex object.

## Return value

Integer.

1 - Success.

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

### **of\_getcurrentposition**

#### **Description**

Gets the current position information. Be sure to call this function separately from the [of\\_open](#) function, otherwise, this function will not work as expected. The best practice is to call this function in a user event which is bound with the oe\_locationchanged event, because the oe\_locationchanged event will be automatically triggered once the of\_open function is finished. The recommended steps are:

1. Call the [of\\_register](#) function to bind the user event with the [oe\\_locationchanged](#) event.
2. Call the [of\\_open](#) function.
3. In the user event that is bound in Step 1, call the of\_getcurrentposition function.

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 <a href="#">eon_mobile_str_coordinates</a> .

#### **Return value**

Integer.

1 - Success.

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

#### **Code example**

The following code example shows you how to call of\_getcurrentposition and of\_open functions separately (in different events), in order for of\_getcurrentposition to work normally.

```
//instance variables of w_1 window  
eon_mobile_geolocationex  ignv_aws  
  
//in the constructor event of w_1 window
```

```
.....
if appeongetclienttype()="MOBILE" then
    ignv_aws = CREATE eon_mobile_geolocationex
        If ignv_aws.of_isenabled() = 1 Then
            ignv_aws.of_open (0, 1)
        Else
            destroy ignv_aws
            messagebox('', 'GPS Disabled')
        End If
end if
.....
// in clicked event of cb_1 of w_1 window
.....
if isvalid( ignv_aws) then
    ignv_aws.of_getcurrentposition (astr_coordinates)
    ignv_aws.of_close()
    messagebox('Long',astr_coordinates.dec_longitude)
    messagebox('Long',astr_coordinates.dec_latitude)
end if
.....
```

The following code example shows you how to call of\_open and of\_getcurrentposition in the same event. But this code example only works in the iOS application, not in the Android application.

```
if appeongetclienttype()="MOBILE" then
    eon_mobile_geolocationex lgnv_aws
        lgnv_aws = CREATE eon_mobile_geolocationex
        If lgnv_aws.of_isenabled() = 1 Then
            lgnv_aws.of_open (0, 1)
            sleep(1)
            lgnv_aws.of_getcurrentposition (astr_coordinates)
            lgnv_aws.of_close()
            messagebox('Long',astr_coordinates.dec_longitude)
            messagebox('Long',astr_coordinates.dec_latitude)
        Else
            messagebox('', 'GPS Disabled')
        End If
        destroy lgnv_aws
end if
```

### **of\_isenabled**

#### **Description**

Detects if the GPS service 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 PowerServer Web, or there is an error.

## **of\_open**

### Description

Opens the GPS functionality. The execution time of this function relies on the network connection between the mobile device and the GPS satellites. Only after this function is finished, can [of\\_getcurrentposition](#) work normally.

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>	<p>Location accuracy update, recommended to use 0 or 1 for the automatic selection.</p> <p>0 - Use the highest-level of accuracy.</p> <p>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.</p> <p>&gt;1 - User-defined accuracy (in meters).</p>
integer	<i>ai_distancefilter</i>	<p>Location filter, used to control the location update message frequency (in meters).</p> <p>0 - Notifies by every update.</p>

Argument Type	Argument	Description
		>0 - Updates only when the location change exceeds this value.

## Return value

Integer.

1 - Success.

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

## of\_register

### Description

Binds user-defined events with the [oe\\_error event](#) and the [oe\\_locationchanged event](#).

After the [oe\\_error event](#) and the [oe\\_locationchanged event](#) are triggered, the bound events will be triggered automatically. The bound events can be utilized to extend the events of the GPS object (which is an NVO object), as NVO object cannot be extended in the PB IDE. If you do not want to extend the events, then you do not need to call the of\_register function, as it will be called internally and automatically.

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 <a href="#">oe_error event</a> and <a href="#">oe_locationchanged event</a> .
string	<i>as_changedevent</i>	The event to be bound with the <a href="#">oe_locationchanged event</a> . If this parameter is set to null, oe_locationchanged event of the bound object will be triggered; in such case, make sure the bound object has the oe_locationchanged event and the same parameter amount and type.
string	<i>as_errorevent</i>	The event to be bound with the <a href="#">oe_error event</a> .

## Return value

Integer.

1 - Success.

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

## of\_triggerevent

### Description

This is an internal function.

### 1.4.7.1.4 Structures

#### 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).
decimal{2}	dec_speed	The displacement velocity (m/sec).
datetime	dt_timestamp	The time stamp to get the current position information.

### 1.4.7.1.5 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_cameraaex

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

// take a photo
li_return = inv_camera.of_takemode (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 with 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_gpserror" event of the Window to
// the "oe_locationchanged" event and "oe_error" event of Geolocation object.
inv_gps.of_register (this, "ue_gps", "ue_gpserror")

```

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)

```

## 1.4.8 Image

Sets the quality or the gray level of an image, or generates a thumbnail of the image with specified options.

### 1.4.8.1 eon\_mobile\_imageex object

#### 1.4.8.1.1 Functions

##### **of\_createimagewithoptions**

###### Description

Generates an image according to the specified options.

Supported on mobile client only.

###### Syntax

```
image.of_createimagewithoptions ( blob ablb_sourcefile, string
as_targetfile, eon_mobile_str_imageoption astr_imageoption, ref
eon_mobile_str_imageinfo astr_imageinfo )
```

```
image.of_createimagewithoptions ( string as_sourcefile, string
as_targetfile, eon_mobile_str_imageoption astr_imageoption, ref
eon_mobile_str_imageinfo astr_imageinfo )
```

Argument Type	Argument	Description
eon_mobile_imageex	<i>image</i>	The name of the eon_mobile_imageex object.
blob	<i>ablb_sourcefile</i>	The source file of the image (Blob data).
string	<i>as_sourcefile</i>	The location and file name of the source image.
string	<i>as_targetfile</i>	The location and file name of the generated image. When file name is empty, no image will be generated, only the blob data will be generated, and if <i>b_returnblob</i> of <b>eon_mobile_str_imageoption</b> is false, the function will return an argument error.
eon_mobile_str_imageoption	<i>imageoption</i>	The arguments for generating the target image. See variable list of <a href="#">eon_mobile_str_imageoption</a> .
eon_mobile_str_imageinfo	<i>imageinfo</i>	The information of the generated image.

Argument Type	Argument	Description
		See variable list of <a href="#">eon_mobile_str_imageinfo</a> .

**Return value**

Integer

- 1 - Successful.  
 0 - Unknown error.  
 -1 - Invalid file location.  
 -2 - Argument error.  
 -3 - Failed to load the source image file.  
 -4 - Failed to save the generated image file.  
 -5 - Failed to compress the image to the specified size.  
 -6 - Insufficient memory.  
 -7 - File already exists.

**1.4.8.1.2 Structures****eon\_mobile\_str\_imageoption****Description**

The options for generating the target image.

**Property**

Type	Variable Name	Description
long	l_width	The width of the target image. If 0, then the original width will be used to generate the image.
long	l_height	The height of the target image. If 0, then the original height will be used to generate the image.
integer	i_quality	The quality of the target image: [1, 10]. 10 indicates the original quality. If it is a different number other than 10, it indicates the quality of the target image and only JPG format will be supported and i_imagetype will be ignored.
double	dbl_filesize	When dbl_filesize (same as Blob, unit is BYTE) is greater than zero, i_quality will be ignored, and the image quality will be automatically adjusted according to the value of i_loopstep, and therefore only JPG image type will be supported.
integer	i_loopstep	The percentage of the quality loss during compression: [1-9]. The number indicates the percentage level (1 is 10%, 9 is 90% etc.) that the quality will be reduced by.

Type	Variable Name	Description
boolean	b_aspectfit	Keep the aspect ratio during compression. When it is true, keep the aspect ratio. If the width aspect ratio (the target image width divided by the source image width) is less than the height aspect ratio (the target image height divided by the source image height), then the width aspect ratio will be used as the compression ratio, otherwise, the height aspect ratio will be used as the compression ratio.
boolean	b_returnblob	Whether to return the indicator of the blob data.
boolean	b_grayimage	Whether to gray out the target image.
boolean	b_existstover	Whether to overwrite the existing file.
integer	i_imagetype	The supported image file type: 0 - JPG, & 1 - PNG
boolean	b_scalefit	When true, the image will be generated at the size of the content area (the image will be scaled to fill up the content area, possibly anamorphically); when false, the image will be generated at the size specified by l_width and l_height and will not be scaled.

### **eon\_mobile\_str\_imageinfo**

#### **Description**

The information of the generated image.

#### **Property**

Type	Variable Name	Description
long	l_width	The width of the generated image.
long	l_height	The height of the generated image.
integer	i_quality	The quality of the generated image: [1, 10]. 10 indicates the original quality.
double	dbl_filesize	The size of the generated image.
blob	ablb_filedata	The data of the generated image, only when b_returnblob of <b>eon_mobile_str_imageoption</b> is true, there is value of ablb_filedata.

#### **1.4.9 Map**

Accesses the information about the longitude, latitude, and address as well as adds/removes annotations.

**Note:** In order to load the map on the Android device, you would need to make sure the Google Play service is installed beforehand.

#### **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 GPS 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 GPS API to receive repeated location updates of the user's position.

### **1.4.9.1 eon\_mobile\_mapex object**

#### **1.4.9.1.1 Properties**

##### **eon\_mobile\_str\_annotation istr\_clickedannotation**

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

##### **integer ii\_errorcode**

Error code which is returned when the oe\_error event is triggered.

1 - MKErrorUnknown

2 - MKErrorServerFailure. The map service is turned off.

3 - MKErrorLoadingThrottled

4 - MKErrorPlacemarkNotFound. Address is not found.

-1 - NSURLErrorUnknown

-999 - NSURLErrorCancelled

-1001 - NSURLErrorTimedOut

-1002 - NSURLErrorCannotFindHost

-1004 - NSURLErrorCannotConnectToHost

-1005 - NSURLErrorNetworkConnectionLost

-1009 - NSURLErrorNotConnectedToInternet

**string is\_errortext**

Error message which will be returned when the oe\_error event is triggered.

#### 1.4.9.1.2 Events

**oe\_annotationclicked****Description**

This 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**

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

Supported on mobile client only.

Note: This event is automatically triggered after calling the [of\\_open function](#).

**Syntax**

None.

**Return value**

None.

**oe\_loadend****Description**

This event is automatically triggered upon the successful completion of loading the map. LPARAM and WPARAM are empty.

Supported on mobile client only.

Note: This event is automatically triggered after calling the [of\\_open function](#).

**Syntax**

None.

**Return value**

None.

#### **oe\_error**

##### **Description**

This event is automatically triggered upon failure in loading the map. WPARAM is the error number.

Supported on mobile client only.

Note: This event is automatically triggered after calling the [of\\_open function](#).

This event will update the value of [is\\_errortext](#) and [ii\\_errorcode](#) properties.

##### **Syntax**

None.

##### **Return value**

None.

#### **oe\_changed**

##### **Description**

This event is automatically triggered when the map display area has changed.

Note: This event is automatically triggered after calling the [of\\_open function](#).

##### **Syntax**

None.

##### **Return value**

None.

#### **1.4.9.1.3 Functions**

##### **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

Argument Type	Argument	Description
		<a href="#">eon_mobile_str_annotation</a> structure.) When the annotation is added successfully, this function will generate and store the unique identifier of the annotation to the l_id property in the structure. And the l_id 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 PowerServer 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="PowerServer Mobile"
istr_annotation_appeon.s_title="Appeon"
uo_map.of_addannotation(istr_annotation_appeon, true)
```

## of\_addressstocoordinate

### Description

Obtains the longitude and latitude according to the address. **Note:** before this function is called, be sure to add an eon\_mobile\_mapex control and call the of\_open function of this control to open the map successfully.

Supported on mobile client only.

### Syntax

```
map.of_addressstocoordinate ( 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.

Argument Type	Argument	Description
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 PowerServer 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 PowerServer Web, or there is an error.

### **of\_coordinateaddress**

#### Description

Converts the longitude and latitude into the corresponding address. **Note:** before this function is called, be sure to add an eon\_mobile\_mapex control and call the of\_open function of this control to open the map successfully.

Supported on mobile client only.

## Syntax

*map.of\_coordinate\_toaddress* ( 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 PowerServer 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

Argument Type	Argument	Description
		<a href="#">eon_mobile_str_mapoption</a> 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 <a href="#">eon_mobile_str_annotation</a> structure.)

## Return value

Integer.

1 - Success.

-1 - Failure or it is called in PowerBuilder or PowerServer 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 <a href="#">eon_mobile_mapex</a> object.
eon_mobile_str_mapoption	<i>astr_mapoption</i>	Relevant information about the map. (For the relevant information, refer to the property of the <a href="#">eon_mobile_str_mapoption</a> structure.)

## Return value

Integer.

1 - Success.

-1 - Failure or it is called in PowerBuilder or PowerServer 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>	<p>The annotation to be removed. (For the relevant information, refer to the property of the <a href="#">eon_mobile_str_annotation</a> structure.)</p> <p>This function will remove the annotation according to the unique identifier of the annotation stored in the <i>l_id</i> property of the structure.</p>
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 PowerServer 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 PowerServer 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>	<p>The annotation to be selected. (For the relevant information, refer to the property of the <a href="#">eon_mobile_str_annotation</a> structure.)</p> <p>This function will select the annotation according to the unique identifier of the annotation stored in the <i>l_id</i> property of the structure.</p>
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 PowerServer Web, or there is an error.

### 1.4.9.1.4 Structures

#### **eon\_mobile\_str\_mapoption**

#### Description

Struct.

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

## Property

Type	Variable Name	Description
boolean	b_locatetocurrentlocation	Moves to the current location.
decimal{6}	dec_longitude	The longitude of the map central location (only takes effect when the value of b_locatetocurrentlocation is False).
decimal{6}	dec_latitude	The latitude of the map central location.
boolean	b_allowmove	Allows the user to move or not.
boolean	b_allowzoom	Allows the user to zoom or not.
integer	i_mapaccuracy	The map display accuracy or amplification factor (unit: meter).
integer	i_maptype	The map type (0: Map (Default), 1: Satellite, 2: Mixed).

## 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

Type	Variable Name	Description
		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.

## 1.4.10 Media

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

### 1.4.10.1 eon\_mobile\_mediaex object

#### 1.4.10.1.1 Functions

##### 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 PowerServer Web, or there is an error.

## 1.4.11 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 NFC tag. 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\\_addrecord](#)  
[of\\_writerecords](#)

- 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 the flowchart and code examples for method 1 and 2 in [Code example](#).

#### 1.4.11.1 eon\_mobile\_nfce object

##### 1.4.11.1.1 Properties

###### **integer ii\_errorcode**

Error code which is returned when the [oe\\_error event](#) is triggered.

0 - Exception. Exception occurs during operation. The error message will be returned by the Android OS.

- 1 - ServiceDisable. NFC service is not enabled or the current device does not support NFC.  
2 - DomainInvalid. Domain is invalid or contains illegal characters such as colon.  
3 - MIMETYPEINVALID. MIME type cannot be empty or contain multi-byte characters.

###### **integer ii\_nfccardtype**

NFC card type.

- 0 - MifareClassic. Requires the encryption key, or uses the default key.
- 1 - MifareUltralight. Requires no encryption key.
- 2 - NfcV. Requires no encryption key.
- 3 - NDEFOnly. Supports read/write NDEF records but does not support read/write data (in bytes) from the block.
- 4 - UNKNOWN. Unrecognizable tag types. Cannot read/write from the tag, but can get tag identifier.

#### **powerobject ipo\_bindevent**

The object to bind with the [oe\\_newtagfound event](#) and [oe\\_error event](#).

#### **string is\_bindwitherrorevent**

The name of the event bound with the [oe\\_error event](#) of the powerobject ipo\_bindevent.

#### **string is\_bindwithnewtagfounevent**

The name of the event bound with the [oe\\_newtagfound event](#) of the powerobject ipo\_bindevent.

#### **string is\_errortext**

Error message which is returned when the [oe\\_error event](#) is triggered.

### **1.4.11.1.2 Events**

#### **oe\_newtagfound**

##### **Description**

Automatically occurs when the NFC tag is detected by the reader. **Note:** This event is triggered internally and automatically.

##### **Syntax**

*nfc.oe\_newtagfound ( integer ai\_type )*

<b>Argument Type</b>	<b>Argument</b>	<b>Description</b>
eon_mobile_nfc		The name of the eon_mobile_nfcex object.
integer	ai_type	<p>0 - MifareClassic. Requires the encryption key, or uses the default key.</p> <p>1 - MifareUltralight. Requires no encryption key.</p> <p>2 - NfcV. Requires no encryption key.</p> <p>3 - NDEFOnly. Supports read/write NDEF records but does not support read/write data (in bytes) from the block.</p> <p>4 - UNKNOWN. Unrecognizable tag types. Cannot read/write from the tag, but can get tag identifier.</p>

**Return value**

None

**oe\_error****Description**

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

Supported on mobile client only.

Note: This event will update the value of [is\\_errortext](#) and [ii\\_errorcode](#) properties.

**Syntax**

None.

**Return value**

None.

**1.4.11.1.3 Functions****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_nfcrex	<i>nfc</i>	The name of the eon_mobile_nfcrex object.
eon_mobile_str_nfcrecord	<i>astr_nfcrecord</i>	The record to be added. See variable list of <a href="#">eon_mobile_str_nfcrecord</a> .

**Return value**

Integer.

1 - Success.

-1 - It is called in PowerBuilder or PowerServer 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_nfce	<code>nfc</code>	The name of the eon_mobile_nfce object.

## Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or PowerServer 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_nfce	<code>nfc</code>	The name of the eon_mobile_nfce object.

## Return value

Integer.

>0 - Returns the size (in bytes) of the block.

-1 - It is called in PowerBuilder or PowerServer 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_nfce	<code>nfc</code>	The name of the eon_mobile_nfce object.

## Return value

Integer.

>0 - Returns the total number of blocks.

-1 - It is called in PowerBuilder or PowerServer 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_nfcecx	<i>nfc</i>	The name of the eon_mobile_nfcecx object.

## Return value

Integer.

>0 - Returns the total number of sectors.

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

## **of\_getndefrecord**

### Description

Gets the NDEF record. The records cannot be read right after they are written to the tag; the tag will need to be detected by the reader again, and then the records can be read by using this function.

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_nfcecx	<i>nfc</i>	The name of the eon_mobile_nfcecx object.
integer	<i>ai_index</i>	<p>The record index which is obtained through <a href="#">of_getndefrecordcount</a>.</p> <p>Record indexes are sequential numbers starting from 1.</p>

Argument Type	Argument	Description
eon_mobile_str_nfcrecord	<i>astr_nfcrecord</i>	The record to be read. See variable list of <a href="#">eon_mobile_str_nfcrecord</a> .

**Return value**

Integer.

1 - Success.

-1 - It is called in PowerBuilder or PowerServer 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_nfcecx	<i>nfc</i>	The name of the <a href="#">eon_mobile_nfcecx</a> object.

**Return value**

Integer.

>0 - Returns the total number of the NDEF records.

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

**of\_getsize****Description**

Gets the total capacity (in bytes) of the tag.

Supported on mobile client only.

**Syntax**

*nfc.of\_getsize ()*

Argument Type	Argument	Description
eon_mobile_nfcecx	<i>nfc</i>	The name of the <a href="#">eon_mobile_nfcecx</a> object.

**Return value**

Integer.

>0 - Returns the total capacity (in bytes) of the tag.

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

### **of\_gettagid**

#### **Description**

Gets the unique identifier of the tag.

Supported on mobile client only.

#### **Syntax**

*nfc.of\_gettagid ( ref string as\_tagid )*

Argument Type	Argument	Description
eon_mobile_nfcecx	<i>nfc</i>	The name of the eon_mobile_nfcecx object.
string	<i>as_tagid</i>	Returns the unique identifier of the tag; or returns empty string if it is called in PowerBuilder or PowerServer Web or if there is any error.

#### **Return value**

Integer.

1 - Success.

-1 - It is called in PowerBuilder or PowerServer 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_nfcecx	<i>nfc</i>	The name of the eon_mobile_nfcecx object.

#### **Return value**

Integer.

>0 - Returns the timeout value for reading/writing data.

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

**of\_open****Description**

Starts the NFC communication. Make sure this function is called before the tag is detected, otherwise, the other functions (such as the read/write functions) will not take effect. If the tag is detected before this function is called, the tag will need to be detected again after this function is called.

Supported on mobile client only.

**Syntax**

```
nfc.of_open ()
```

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

**Return value**

Integer.

1 - Success.

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

**of\_readblock****Description**

Reads data (in bytes) from the block. Make sure [of\\_startblockoperate](#) is called before the tag is detected and this function is called.

The records cannot be read right after they are written to the tag; the tag will need to be detected by the reader again, and then the records can be read by using the [of\\_readblock](#) function.

Supported on mobile client only.

**Syntax**

```
nfc.of_readblock ( integer ai_block, ref blob abl_data )
```

Argument Type	Argument	Description
eon_mobile_nfcecx	<i>nfc</i>	The name of the eon_mobile_nfcecx object.
integer	<i>ai_block</i>	The block index which is obtained through <a href="#">of_getMaxBlockCount</a> . 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 PowerServer Web, or there is an error.

## of\_register

### Description

Binds user-defined events with the [oe\\_newtagfound](#) event and the [oe\\_error](#) event.

After oe\_newtagfound and oe\_error are triggered, the bound events will be triggered automatically. The bound events can be utilized to extend the events of the NFC object, as the NFC object is an NVO object which cannot be extended in the PB IDE. If you do not want to extend the events, then you do not need to call the of\_register function, as it will be called internally and automatically.

Supported on mobile client only.

### Syntax 1

```
nfc.of_register ( powerobject apb_bind, string as_newtagfound )
```

### Syntax 2

```
nfc.of_register ( powerobject apb_bind, string as_newtagfound, string as_error )
```

Argument Type	Argument	Description
eon_mobile_nfce	<i>nfc</i>	The name of the eon_mobile_nfce object.
powerobject	<i>apb_bind</i>	The object to be bound with the <a href="#">oe_newtagfound</a> event and the <a href="#">oe_error</a> event.
string	<i>as_newtagfound</i>	The event to be bound with the <a href="#">oe_newtagfound</a> event.
string	<i>as_error</i>	The event to be bound with the <a href="#">oe_error</a> event.

## Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or PowerServer 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_nfcecx	<i>nfc</i>	The name of the eon_mobile_nfcecx object.
string	<i>as_key</i>	Encryption key for reading/writing data. Or use the system default key. MIFARE Classic tags require the key. Tag types can be obtained through <a href="#">oe_newtagfound</a> event.

## Return value

Integer.

1 - Success.

-1 - It is called in PowerBuilder or PowerServer 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_nfcecx	<i>nfc</i>	The name of the eon_mobile_nfcecx 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 operation will time out and disconnect automatically.

## Return value

Integer.

1 - Success.

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

### **of\_startblockoperate**

#### **Description**

Starts reading/writing data from the block. This function must be called before [of\\_writeblock](#) or [of\\_readblock](#) is called.

Supported on mobile client only.

#### **Syntax**

```
nfc.of_startblockoperate ()
```

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

#### **Return value**

Integer.

1 - Success.

-1 - It is called in PowerBuilder or PowerServer 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_nfcecx	<i>nfc</i>	The name of the eon_mobile_nfcecx object.

#### **Return value**

Integer.

1 - Success.

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

### **of\_triggeerevent**

#### **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
eon_mobile_nfcecx	<i>nfc</i>	The name of the eon_mobile_nfcecx object.
string	<i>as_event</i>	The event name bound with <a href="#">oe_newtagfound</a> event.

## Return value

Integer.

1 - Success.

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

## [of\\_writeblock](#)

### Description

Writes data (in bytes) to the block. Before this function is called, make sure [of\\_startblockoperate](#) is called first, otherwise this function will not take effect.

After the records are successfully written to the tag, they cannot be read immediately, the tag will need to be detected by the reader again, and then the records can be read successfully by using the [of\\_readblock](#) function.

Supported on mobile client only.

## Syntax

```
nfc.of_writeblock ( integer ai_block, blob abl_data )
```

Argument Type	Argument	Description
eon_mobile_nfcecx	<i>nfc</i>	The name of the eon_mobile_nfcecx object.
integer	<i>ai_block</i>	<p>The block index which is obtained through <a href="#">of_getMaxBlockCount</a>.</p> <p>Block indexes are sequential numbers starting from 1.</p>
blob	<i>abl_data</i>	<p>Data to be written to the block.</p> <p>Block size should be obtained through <a href="#">of_getBlockSize</a> first. Block size can be 4 bytes or 16 bytes. Data exceeding the block size will be automatically truncated.</p>

## Return value

Integer.

1 - Success.

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

## **of\_writerecords**

### Description

Writes the NDEF record. [of\\_addrecord](#) should be called first to add one or more records, then of\_writerecords should be called to write all records at one time. After the records are successfully written to the tag, they cannot be read immediately, the tag will need to be detected by the reader again, and then the records can be read successfully by using the [of\\_getndefrecord](#) function.

Supported on mobile client only.

### Syntax

*nfc.of\_writerecords ()*

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

## Return value

Integer.

1 - Success.

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

### 1.4.11.1.4 Structures

#### **eon\_mobile\_str\_nfcrecord**

##### Description

Structure.

The record to be added.

##### Property

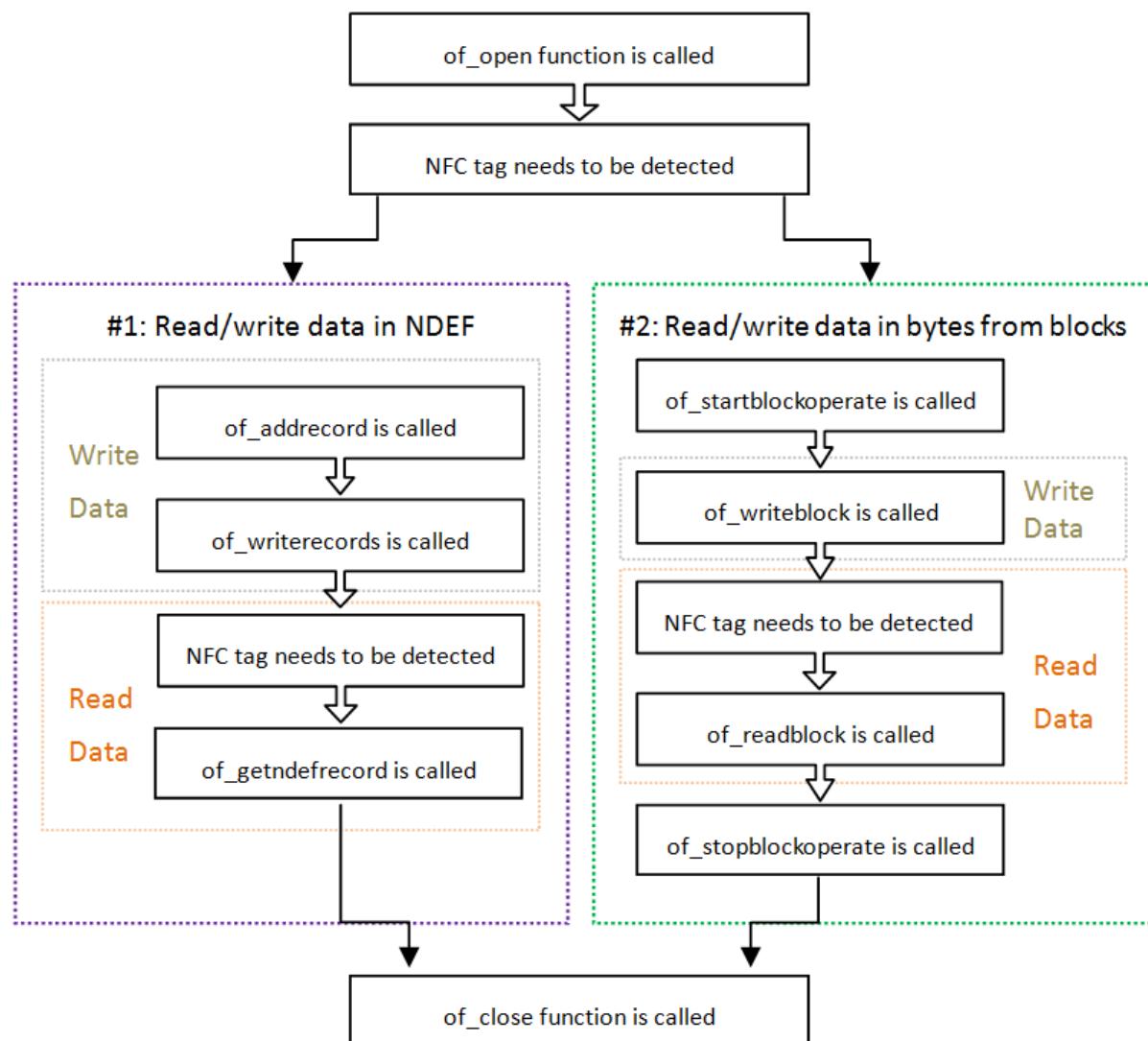
Type	Variable Name	Description
int	i_messagestype	0 - URL message 1 - Mime message 2 - App message 3 - Text message 4 - Poster message

Type	Variable Name	Description
		5 - Alternative carrier message 6 - Handover carrier message 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.

#### 1.4.11.1.5 Code example

Below flowchart illustrates how NFC functionality works.

**Figure 1.10: NFC flowchart**



- [of\\_open](#): it must be called before the NFC tag is detected, otherwise, the other functions (such as the read/write functions) will not take effect. If the NFC tag is detected before this function is called, the tag will need to be detected again after this function is called.
- [of\\_addrecord](#) and [of\\_writerecords](#): of\_addrecord should be called first to add one or more records, then of\_writerecords should be called to write all records at one time.
- [of\\_getndefrecord](#): After the records are successfully written to the tag, they cannot be read immediately, the tag will need to be detected by the reader again, and then the records can be read successfully by the of\_getndefrecord function.
- [of\\_startblockoperate](#): it must be called before of\_writeblock or of\_readblock is called, otherwise of\_writeblock and of\_readblock will not take effect.
- [of\\_readblock](#): After the records are successfully written to the tag, they cannot be read immediately, the tag will need to be detected by the reader again, and then the records can be read successfully by the of\_readblock function.

### **Code example for method 1:** Read/write data in NDEF (NFC Data Exchange Format)

```
eon_mobile_nfce   Inv_mobile_nfc //create nfc object
eon_mobile_str_nfcrecord      istr_nfcrecord

//open nfc object
Inv_mobile_nfc.of_open()

//read nfc tag 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 tag 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_nfce   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 tag record
Inv_mobile_nfc.of_readBlock(li_block, ibl_data)

//write nfc tag record
Inv_mobile_nfc.of_writeBlock(li_block, ibl_data)

Inv_mobile_nfc.of_stopBlockOperate()

//close nfc object
Inv_mobile_nfc.of_close()
```

## 1.4.12 Notification

Sends a notification to the notification bar when the mobile application is running in the foreground. If the application is closed or running in the background, no notification can be sent.

### Usage example

- Airlines:

Because of the weather effects and other force majeures, 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.

### 1.4.12.1 `eon_mobile_notificationex` object

#### 1.4.12.1.1 Functions

##### `of_addmessage`

###### Description

Sends a notification to the notification bar when the mobile application is running in the foreground. If the application is closed or running in the background, no notification can be sent.

Supported on mobile client only.

###### Syntax 1

```
notification.of_addmessage ( value string as_message )
```

###### Syntax 2

```
notification.of_addmessage ( value string as_title, value string as_message )
```

Argument Type	Argument	Description
<code>eon_mobile_notificationex</code>	<code>notification</code>	The name of the <code>eon_mobile_notificationex</code> object.

Argument Type	Argument	Description
string	<i>as_title</i>	The title to be displayed on the notification bar.  This parameter takes effect on the Android device only. On the iOS device, this parameter will be ignored, and the title is always the application name, which means if your app runs in Appeon Workspace, the title will be "AppeonMobile"; if your app is a native app packaged via the PowerServer Toolkit Package tool, the title will be whatever specified in the App Name field when packaged.
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 PowerServer Web, or there is an error.

## Code example

```
eon_mobile_notificationex ieon_notification
integer ii_notification

ieon_notification = CREATE eon_mobile_notificationex

string ls_now
ls_now = string (now(), 'HH:MM:SS')
ii_notification = ieon_notification.of_addmessage ("Time", ls_now)
```

Note that the of\_addmessage function will send a notification to the notification bar; and the notification will show on top of the screen when the end user swipe down from the top of the screen to trigger the Notification Center.

## [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 returned by the <a href="#">of_addmessage() function</a> .

## Return value

Integer.

1 - Success.

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

## Code example

```
eon_mobile_notificationex ieon_notification
integer ii_notification

ieon_notification = CREATE eon_mobile_notificationex

string ls_now
ls_now = string (now(), 'HH:MM:SS')
ii_notification = ieon_notification.of_addmessage ("Time", ls_now) // add
ieon_notification.of_removemessage (ii_notification) // remove
```

Do not use the of\_removemessage function to remove the notification from the notification bar as soon as it is added, otherwise it will not show in the Notification Center. Instead, you should remove the notification from the Notification Center as needed.

## 1.4.13 Offline

Checks and applies the updates of the current application.

### 1.4.13.1 eon\_offlineex object

#### 1.4.13.1.1 Functions

##### 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 PowerServer Web and this operation is not supported.

## **of\_checkappeonserver**

### Description

Detects if PowerServer 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 - PowerServer exists.

-1 - PowerServer does not exist or there is an error in the checking of the PowerServer, 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

Argument Type	Argument	Description
		<p>prompts the end user to restart the app after updated and confirm to install the update. 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 <b>OK</b> to update the app, or choose <b>Cancel</b> to not to update the app.</p> <p>True - To display the dialog box.</p> <p>False - Not to display the dialog box.</p>

## Return value 1

Integer.

1 - Success.

2 - User has clicked **Cancel** in the dialog box to not to update the app.

0 - Latest app and no need to update.

-1 - Failure or it is called in PowerBuilder.

-2 - It is called in PowerServer 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
<i>eon_offlineex</i>	<i>offline</i>	The name of the <i>eon_offlineex</i> 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 PowerServer 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 > PB Features >INI Files
  2. Application > PB Features > Decimal Precision
  3. Application > Client Features > [codeexamples] > Application Title
  4. Application > Client Features > [codeexamples] > Retina Display (Mobile Only)
  5. Application > Data Transfer > Encoding
  6. Mobile UI Resizing
- Application configuration
- Data source
- Embedded SQL
- DW SQL

The following application configuration information will be synchronized when running the offline mobile application for the first time or when calling the of\_applydbupdate function.

- AEM configuration
  1. Application > Local Database
- Local database files

### **of\_checkupdate**

#### **Description**

Detects if there is any update for 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 for the application.
- 2 - There is an update only for the database file.
- 3 - There are updates for both the database file and the application.
- 1 - Failure in the update check or it is called in PowerBuilder.
- 2 - It is called in PowerServer Web and this operation is not supported.

### 1.4.14 Online Payment

Makes online payment with the third-party payment service providers such as PayPal, AliPay, Western Union, Google etc.

Currently PayPal is the only supported provider and Android is the only supported OS for PayPal (iOS support is discontinued since version 2019).

#### 1.4.14.1 eon\_mobile\_paymentex object

##### 1.4.14.1.1 Properties

###### **integer ii\_appplatformtype**

Platform type. Currently PayPal is the only provider supported by PowerServer.

0 - Unknown.

1 - PayPal.

###### **integer ii\_errorcode**

Error code which is returned when the [oe\\_error event](#) is triggered. It is designed to hold the error code returned from the payment platform, but currently PayPal returns no error code, so this property is reserved for future use.

###### **powerobject ipo\_bindevent**

The object to bind with the [oe\\_error event](#).

###### **string is\_bindwitherrorevent**

The name of the event bound with the [oe\\_error event](#) of the powerobject ipo\_bindevent.

###### **string is\_errortext**

Error message which is returned when the [oe\\_error event](#) is triggered.

###### **string is\_paymentcancel**

The name of the event bound with the [oe\\_paymentcancel event](#).

###### **string is\_paymentfailed**

The name of the event bound with the [oe\\_paymentfailed event](#).

**string is\_paymentok**

The name of the event bound with the [oe\\_paymentok event](#).

#### 1.4.14.1.2 Events

**oe\_error****Description**

Occurs when there is any error in this functionality.

Supported on Android client only (Discontinued on iOS since version 2019).

**Syntax**

None.

**Return value**

None.

**oe\_paymentcancel****Description**

Occurs when the payment is cancelled or terminated.

Supported on Android client only (Discontinued on iOS since version 2019).

**Syntax**

None.

**Return value**

None.

**oe\_paymentfailed****Description**

Occurs when the payment failed.

Supported on Android client only (Discontinued on iOS since version 2019).

**Syntax**

None.

**Return value**

None.

**oe\_paymentok****Description**

Occurs when the payment is successful.

Supported on Android client only (Discontinued on iOS since version 2019).

**Syntax**

None.

### **Return value**

None.

#### **1.4.14.1.3 Functions**

##### **of\_getappplatform**

###### **Description**

Detects which payment platform the application is connecting now to make payments. Currently PayPal is the only supported platform.

Supported on Android client only (Discontinued on iOS since version 2019).

###### **Syntax**

*payment.of\_getappplatform ()*

Argument Type	Argument	Description
eon_mobile_paymentex	<i>payment</i>	The name of the eon_mobile_paymentex object.

##### **Return value**

Integer.

0 - Unknown.

1 - PayPal.

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

##### **of\_init**

###### **Description**

Initializes the payment record.

Supported on Android client only (Discontinued on iOS since version 2019).

###### **Syntax**

*payment.of\_init ( eon\_mobile\_str\_paymentinit *astr\_paymentinit* )*

Argument Type	Argument	Description
eon_mobile_paymentex	<i>payment</i>	The name of the eon_mobile_paymentex object.
eon_mobile_str_paymentinit	<i>astr_paymentinit</i>	The payment record to be initialized. (For details, refer to the property of the

Argument Type	Argument	Description
		<a href="#">eon_mobile_str_paymentinit</a> structure.)

## Return value

Integer.

1 - Success.

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

## of\_register

### Description

Registers the object and the event to be bound with the oe\_paymentok, oe\_paymentcancel, oe\_paymentfailed, and oe\_error events.

Supported on Android client only (Discontinued on iOS since version 2019).

### Syntax

```
payment.of_register ( powerobject apb_bind, string as_paymentok, string
as_paymentcancel, string as_paymentfailed, string as_error )
```

Argument Type	Argument	Description
eon_mobile_paymentex	<i>payment</i>	The name of the eon_mobile_paymentex object.
powerobject	<i>apb_bind</i>	The object to be bound with the oe_paymentok, oe_paymentcancel, oe_paymentfailed, and oe_error events.
string	<i>as_paymentok</i>	The event to be bound with the oe_paymentok event.
string	<i>as_paymentcancel</i>	The event to be bound with the oe_paymentcancel event.
string	<i>as_paymentfailed</i>	The event to be bound with the oe_paymentfailed event.
string	<i>as_error</i>	The event to be bound with the oe_error event.

## Return value

Integer.

1 - Success.

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

**of\_setappplatform****Description**

Sets the payment platform which the application will connect to make payments. Currently PayPal is the only supported platform.

Supported on Android client only (Discontinued on iOS since version 2019).

**Syntax**

```
payment.of_setappplatform( integer ai_appplatform )
```

Argument Type	Argument	Description
eon_mobile_paymentex	<i>payment</i>	The name of the eon_mobile_paymentex object.
integer	<i>ai_appplatform</i>	Type of the payment platform. Currently PayPal is the only supported platform. 0 - Unknown. 1 - PayPal.

**Return value**

Integer.

1 - Success.

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

**of\_submit****Description**

Submits the payment record.

Supported on Android client only (Discontinued on iOS since version 2019).

**Syntax**

```
payment.of_submit( eon_mobile_str_paymentsubmit astr_paymentsubmit )
```

Argument Type	Argument	Description
eon_mobile_paymentex	<i>payment</i>	The name of the eon_mobile_paymentex object.
eon_mobile_str_paymentsubmit	<i>astr_paymentsubmit</i>	The payment record to be submitted. (For details, refer to the property of the <a href="#">eon_mobile_str_paymentsubmit</a> structure.)

**Return value**

Integer.

1 - Success.

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

### **of\_triggerevent**

#### **Description**

Triggers the object event that is bound with oe\_error or the other events.

Supported on Android client only (Discontinued on iOS since version 2019).

#### **Syntax**

*payment.of\_triggerevent ( string as\_event )*

Argument Type	Argument	Description
eon_mobile_paymentex	<i>payment</i>	The name of the eon_mobile_paymentex object.
string	<i>as_event</i>	The event name bound with oe_error or the other events.

#### **Return value**

None.

#### **1.4.14.1.4 Structures**

##### **eon\_mobile\_str\_paymentinit**

#### **Description**

Struct.

Detailed information about the initialized payment record.

#### **Property**

Type	Variable Name	Description
string	s_clientid	PayPal ID for the merchant.

##### **eon\_mobile\_str\_paymentssubmit**

#### **Description**

Struct.

Detailed information about the payment record to be submitted.

#### **Property**

Type	Variable Name	Description
double	dbl_countmoney	Total amount of money.
string	s_currency	Currency.

Type	Variable Name	Description
string	s_productname	Product name.

#### 1.4.14.1.5 Code example

Below is the sample code for making an online payment via PayPal.

```

eon_mobile_paymentex lnv_payment
eon_mobile_str_paymentinit lstr_paymentinit
eon_mobile_str_paymentsubmit lstr_paymentsubmit
integer li_re

lnv_payment = create eon_mobile_paymentex

//bind oe_paymentok, oe_paymentcancel, oe_paymentfailed, and oe_error events of
lnv_payment with
//ue_paymentok, ue_paymentcancel, ue_paymentfailed, and ue_error events of the
parent window.
lnv_payment.of_register ( parent, "ue_paymentok", "ue_paymentcancel",
"ue_paymentfailed", "ue_error" )

//initialize platform
lnv_payment.of_setappplatform(1)

//initialize payment
lstr_paymentinit.s_clientid = "AXMN1BBgmfChHPgrMAtrrdTtkWk52THb9H154uZ6vASDWS0118"
li_re = lnv_payment.of_init (lstr_paymentinit)

if li_re = 1 then
    //submit payment
    lstr_paymentsubmit.dbl_countmoney = 9.99
    lstr_paymentsubmit.s_currency = "USD"
    lstr_paymentsubmit.s_productname = "shoes"

    li_re = lnv_payment.of_submit (lstr_paymentsubmit)
    if li_re = 1 then
        MessageBox ( "", "Payment is successful!" )
    else
        MessageBox ( "", "Payment failed!" )
    end if
else
    MessageBox( "", "Failed to initialize payment!" )
end if

destroy lnv_payment

```

#### 1.4.15 Textchecker

Checks spelling in a text field.

##### 1.4.15.1 eon\_mobile\_textcheckex object

###### 1.4.15.1.1 Functions

###### of\_completionsforpartialword

###### Description

Returns a list of words that are possible completions for a partially entered 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 <a href="#">eon_mobile_str_textcheckoption</a> .
string	<i>as_completions[]</i>	Returns a list of words, 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 PowerServer 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 PowerServer 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 PowerServer 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

Argument Type	Argument	Description
		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 PowerServer 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 <a href="#">eon_mobile_str_textcheckoption</a> .
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 PowerServer Web, or there is an error.

### **of\_guessesforword**

#### Description

Returns a list of words that are possible valid replacements for a misspelled word.

Supported on mobile client only.

### Syntax

```
textchecker.of_guessesforword ( 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 <a href="#">eon_mobile_str_textcheckoption</a> .
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 PowerServer 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.

Argument Type	Argument	Description
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 PowerServer 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 PowerServer 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 PowerServer Web, or there is an error.

#### 1.4.15.1.2 Structures

##### **eon\_mobile\_str\_textcheckoption**

###### Description

Struct.

Checks spelling.

###### Property

Type	Variable Name	Description
string	s_source	The string which you want to check.
long	l_start	A long indicating where the check will begin in s_source.
long	l_length	Starting from the position specified by l_start, the number of characters needed to be checked.
string	s_language	The language of the words to be checked for correct spelling. This variable takes no effect on the Android OS if the text is input in multiple languages.

#### 1.4.15.1.3 Code example

The following script is in the EditChanged event for a SingleLineEdit control, used to detect if the input word is correct.

```
long ll_start,ll_length,ll_return1,ll_return2
string ls_return[],ls_data
eon_mobile_str_textcheckoption  leon_str_textcheck
str_textcheck  lstr_textcheck
```

```

leon_str_textcheck.s_language = 'en_US'
leon_str_textcheck.s_source = this.text
leon_str_textcheck.l_start = 1
leon_str_textcheck.l_length = len(this.text)
ll_return1 = ieon_textchecker.of_getmisspelledword ( leon_str_textcheck, ll_start,
ll_length) //returns the starting position and length of the first misspelled word
if ll_return1 = 1 then
    leon_str_textcheck.l_start = ll_start
    leon_str_textcheck.l_length = ll_length //IMPORTANT: passes the length to the
structure so it can exactly obtain the misspelled word
    if cbx_1.checked then
        ll_return2 = ieon_textchecker.of_guessesforword ( leon_str_textcheck,
ls_return) //returns a list of possible guesses
    else
        ll_return2 = ieon_textchecker.of_completionsforpartialword
( leon_str_textcheck, ls_return) //returns a list of possible completions
    end if
    if upperbound(ls_return) > 0 then
        lstr_textcheck.s_old = mid(this.text,ll_start,ll_length)
        lstr_textcheck.s_new = ls_return
        lstr_textcheck.l_x = parent.pointerx( )
        lstr_textcheck.l_y = parent.pointery( )
        openwithparm(w_rightword,lstr_textcheck)
        ls_data = message.stringparm
        if len(ls_data) > 0 then
            this.text = replace(this.text,ll_start,ll_length,ls_data)
        end if
    end if
end if
end if

```

## 1.4.16 WebView

Loads the HTML page and/or runs JavaScript, displays files such as text file, XML file, PDF etc. and explores the Web page.

### 1.4.16.1 eon\_mobile\_webviewex object

#### 1.4.16.1.1 Properties

##### **integer ii\_errorcode**

Error code which is returned when the [oe\\_error](#) event is triggered.

##### **string is\_errortext**

Error message which is returned when the [oe\\_error](#) event is triggered.

#### 1.4.16.1.2 Events

##### **oe\_error**

##### **Description**

Occurs when there is any error in this functionality. In the Android OS, when this event is triggered, an error page from the Android OS will be displayed. However, unlike Android, iOS will not return any error page or error message box, therefore, it is recommended that you write code in this event to display an error message box or an error page which you specially prepared for the iOS device.

Supported on mobile client only.

## Syntax

None.

## Return value

None.

### **oe\_jcallbackapeon**

#### Description

Occurs automatically when the Web page loaded into the eon\_mobile\_webviewex object calls this JavaScript function: `callbackapeon (as_data)` for iOS, or `eon_android.callbackapeon (as_data)` for Android. So this event and the JavaScript function together can be utilized as communication bridge between the Web page and the PowerServer mobile app.

`callbackapeon (as_data)` is a JavaScript function predefined in PowerServer for use in iOS only, and the `eon_android` JavaScript object and its `callbackapeon (as_data)` function are predefined in PowerServer for use in Android only. Users need not to declare them before calling them; and users should prevent declaring and overloading them too. If these functions are called by a Web page that is not loaded into the `eon_mobile_webviewex` object, they will return errors such as "`"callbackapeon`' is undefined" or "`"eon_android`' is undefined".

Supported on mobile client only.

## Syntax

`oe_jcallbackapeon ( string as_data )`

Argument Type	Argument	Description
string	<code>as_data</code>	Receives the string passed in by the parameter of the <code>callbackapeon</code> JavaScript function. You will need to make sure the value to be passed in is a string, not of any other data type.

## Return value

None.

## Code example

The following code shows how to encapsulate a `callbackapeon (as_data)` function and then call it instead of calling the `callbackapeon (as_data)` or `eon_android.callbackapeon (as_data)` directly.

```
function jcallbackapeon(as_data)
{
    if ( typeof(callbackapeon) == "function" )
    {
        callbackapeon(as_data);
    }
    else if (typeof(eon_android) == "object" )
    {
    }
```

```

        eon_android.callbackappeon(as_data);
    }
}

```

**oe\_loadend****Description**

Occurs when the [of\\_startloading](#) function finishes loading the page.

Supported on mobile client only.

**Syntax**

None.

**Return value**

None.

**oe\_loadstart****Description**

Occurs when the [of\\_startloading](#) function starts loading the page.

Supported on mobile client only.

**Syntax**

None.

**Return value**

None.

**1.4.16.1.3 Functions****of\_cangoback****Description**

Indicates whether the user can move backward.

Supported on mobile client only.

**Syntax**

`webview.of_cangoback ()`

Argument Type	Argument	Description
<code>eon_mobile_webviewex</code>	<code>webview</code>	The name of the <code>eon_mobile_webviewex</code> object.

**Return value**

Integer.

0 - Unable to move backward.

1 - Able to move backward.

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

### **of\_cangoforward**

#### **Description**

Indicates whether the user can move forward.

Supported on mobile client only.

#### **Syntax**

```
webview.of_cangoforward()
```

Argument Type	Argument	Description
eon_mobile_webviewex	<i>webview</i>	The name of the eon_mobile_webviewex object.

#### **Return value**

Integer.

0 - Unable to move forward.

1 - Able to move forward.

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

### **of\_goback**

#### **Description**

Loads the previous location in the back-forward list.

Supported on mobile client only.

#### **Syntax**

```
webview.of_goback()
```

Argument Type	Argument	Description
eon_mobile_webviewex	<i>webview</i>	The name of the eon_mobile_webviewex object.

#### **Return value**

Integer.

1 - Success.

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

### **of\_goforward**

#### **Description**

Loads the next location in the back-forward list.

Supported on mobile client only.

### Syntax

```
webview.of_goforward( )
```

Argument Type	Argument	Description
eon_mobile_webviewex	<i>webview</i>	The name of the eon_mobile_webviewex object.

### Return value

Integer.

1 - Success.

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

### of\_loadhtmlstring

#### Description

Loads the string of the HTML page. Only HTM or HTML file is supported.

Supported on mobile client only.

### Syntax

```
webview.of_loadhtmlstring( value string as_htmlstring )
```

Argument Type	Argument	Description
eon_mobile_webviewex	<i>webview</i>	The name of the eon_mobile_webviewex object.
string	<i>as_htmlstring</i>	The string of the HTML page which you want to load.

### Return value

Integer.

1 - Success.

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

### Code example

```
string ls_filepath,ls_file
integer li_ret
long ll_file

ls_file = appeongetcachedir()+"plugin/customer.html"

ll_file = fileopen(ls_file,textmode!,read!,shared!)
filereadex(ll_file,ls_filepath)
messagebox('',ls_filepath)
```

```

li_ret = ue_webview.of_loadhtmlstring (ls_filepath)
if li_ret > -1 then
  messagebox("note:", "success of_loadhtmlstring: Filepath#" +ls_filepath + ",",
  return value#" + string(li_ret))
else
  messagebox("note:", "failed of_readBlock:: Filepath#" +ls_filepath + ", return
  value#" + string(li_ret))
end if

fileclose(ll_file)

```

## **of\_loadlocalfile**

### **Description**

Loads and displays the local file.

In iOS, the following file types can be loaded and displayed: PDF, TXT, XML, Word, Excel, and PPT. Android supports fewer file types as it can only support text types such as TXT, XML, and HTML. Android does not support loading the PDF file, therefore you will not be able to load the local PDF file in Android, but if you put the PDF file on a Web site accessible via Internet, and allow the mobile device to connect with the Web site and Google Docs (docs.google.com), then PowerServer can parse the PDF file online via Google Docs (docs.google.com) and then load the PDF file into the Android device, for example, if you put the PDF file online as [http://demo.appeon.com/AEM/appeon\\_mobile\\_2013R2\\_datasheet.pdf](http://demo.appeon.com/AEM/appeon_mobile_2013R2_datasheet.pdf), then PowerServer mobile app will automatically display the online PDF file via URL [http://docs.google.com/gview?embedded=true&url=http://demo.appeon.com/AEM/appeon\\_mobile\\_2013R2\\_datasheet.pdf](http://docs.google.com/gview?embedded=true&url=http://demo.appeon.com/AEM/appeon_mobile_2013R2_datasheet.pdf).

Supported on mobile client only.

### **Syntax**

`webview.of_loadlocalfile ( value string as_filepath )`

Argument Type	Argument	Description
eon_mobile_webviewex	<code>webview</code>	The name of the eon_mobile_webviewex object.
string	<code>as_filepath</code>	The path of the local file which you want to load.

### **Return value**

Integer.

1 - Success.

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

### **Code example**

```

int li_ret
string ls_filepath

ls_filepath = appeongetcachedir() + "/plugin/pig.txt"
li_ret = ue_webview.of_loadlocalfile (ls_filepath)
if li_ret > -1 then

```

```

messagebox("note:", "success of_loadlocalfile:FilePath:" +ls_filepath + ", return
value#" + string(li_ret))
else
  messagebox("note:", "failed of_loadlocalfile:FilePath:" +ls_filepath + ", return
value#" + string(li_ret))
end if

```

**of\_reload****Description**

Reloads the current page.

Supported on mobile client only.

**Syntax**

```
webview.of_reload()
```

Argument Type	Argument	Description
eon_mobile_webviewex	<i>webview</i>	The name of the eon_mobile_webviewex object.

**Return value**

Integer.

1 - Success.

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

**of\_runjavascript****Description**

Runs the JavaScript code snippet.

Supported on mobile client only.

**Syntax**

```
webview.of_runjavascript ( value string as_strjs, ref string as_strret )
```

Argument Type	Argument	Description
eon_mobile_webviewex	<i>webview</i>	The name of the eon_mobile_webviewex object.
string	<i>as_strjs</i>	The script which you want to run. You can use the syntax "functionname()" to call the JavaScript methods or the methods of the current page.
string	<i>as_strret</i>	The result of running script. Any value returned from JavaScript will be returned

Argument Type	Argument	Description
		here; if there is no return value from JavaScript (like Alert, or simple script for assigning values to variables), -1 will be returned, if there is any error running JavaScript, then an empty string will be returned.

## Return value

Integer.

1 - If the *as\_strret* parameter returns a non-empty string.

-1 - If the *as\_strret* parameter returns an empty string, or if it is called in PowerBuilder or PowerServer Web.

## Code example

```
string ls_strjs,ls_strret
integer li_ret
ls_strjs= 'function MyJavaScript() { var message; message = "Say Hello"; return
message; } MyJavaScript()'

li_ret = uo_webview.of_runjavascript (ls_strjs,ls_strret)

if li_ret > -1 then
    mle_1.text = "success of_runjavascript:" +ls_strret+ ",return value#" +
    string(li_ret)
else
    mle_1.text = "failed of_runjavascript:" +ls_strret+ ", return value#" +
    string(li_ret)
end if
```

## of\_startloading

### Description

Starts the loading of the page.

Supported on mobile client only.

### Syntax

```
webview.of_startloading ( value string as_url )
```

Argument Type	Argument	Description
eon_mobile_webviewex	<i>webview</i>	The name of the eon_mobile_webviewex object.
string	<i>as_url</i>	The URL of the page which you want to load.

## Return value

Integer.

1 - Success.

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

### Code example

```
int li_ret
string ls_url

ls_url= "www.google.com"
li_ret = uo_webview.of_startloading(ls_url)
if li_ret > -1 then
  messagebox("note:", "success of_startloading:URL:" + ls_url + "#return value#"
+string(li_ret))
else
  messagebox("note:", "failed of_startloading:URL:" + ls_url + "#return value#"
+string(li_ret))
end if
```

## of\_stoploading

### Description

Stops the loading of the current page.

Supported on mobile client only.

### Syntax

```
webview.of_stoploading()
```

Argument Type	Argument	Description
eon_mobile_webviewex	<i>webview</i>	The name of the eon_mobile_webviewex object.

### Return value

Integer.

1 - Success.

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

## 2 Appeon Resize PBL Reference

The Appeon Resize PBL (`appeon_resize.pbl`) has been removed from the product, and the object under the Appeon Resize PBL has been moved to the Appeon Workarounds PBL (`appeon_workarounds.pbl`). For the syntax and code examples of the Appeon resize object, refer to the [Appeon Resize Object](#) section.

# 3 Integrate with Cordova Plugins

## 3.1 Overview

[Apache Cordova plugins](#) provide JavaScript interfaces that can be called to use native device capabilities to enhance the features and functionality of a mobile application. To enable PowerServer mobile apps to take advantage of Cordova plugins, PowerServer Mobile has packaged the common Cordova plugins and provided two solutions for a PowerBuilder developer to call the plugins:

- Solution 1: Call the PowerScript interfaces that Appeon Cordova PBL provides. The Appeon Cordova PBL provides interfaces that can automatically execute the corresponding JavaScript methods and return the results from JavaScript, therefore, the developer is able to call the Cordova plugins as easily as call a PowerBuilder object, without needing to know the JavaScript language.

For detailed instructions, refer to [Appeon Cordova PBL](#).

However, the Appeon Cordova PBL only encapsulates the following Cordova plugins as PowerBuilder objects: Bluetooth, Bluetooth Low Energy, Contacts, Fingerprint authentication, Geolocation, Device orientation, & Vibration.

- Solution 2: Call the JavaScript interfaces that Cordova plugins provide. This requires the developer to at least be able to read the JavaScript code in the Cordova plugins. This solution gives the developer most flexibility to use any Cordova plugins that are included in PowerServer ([view list](#)). If you found a Cordova plugin not included in PowerServer, you can contact Appeon Support (support@appeon.com) to make plans for including it.

For detailed instructions, refer to [Cordova plugins](#).

If you found any issue with the Cordova plugins, please first search the [Cordova plugin issue list](#) to see if the issue already exists; if the issue does not exist, you can submit the issue after logged in.

## 3.2 Appeon Cordova PBL ([solution 1 \[207\]](#))

### 3.2.1 Introduction to Appeon Cordova PBL

Appeon Cordova PBL encapsulates the following Cordova plugins as PowerBuilder objects. Though not all of the current Cordova plugins are encapsulated in the Appeon Cordova PBL, you may still find it useful because the most frequently-used ones are in it and they can be called as easily as a PowerBuilder object.

- Bluetooth (<https://github.com/tanelih/phonegap-bluetooth-plugin>)
- Bluetooth Low Energy (<https://www.npmjs.com/package/cordova-plugin-bluetoothle>)
- Contacts (<https://www.npmjs.com/package/cordova-plugin-contacts>)
- Fingerprint authentication (Android: <https://www.npmjs.com/package/cordova-plugin-android-fingerprint-auth>, iOS: <https://www.npmjs.com/package/cordova-plugin-touch-id>)

- Geolocation (<https://www.npmjs.com/package/cordova-plugin-mauron85-background-geolocation>)
- Device orientation (<https://www.npmjs.com/package/cordova-plugin-device-orientation>)
- Vibration (<https://www.npmjs.com/package/cordova-plugin-vibration>)

Appeon Cordova PBL is named **appeon\_cordova.pbl**, located in the *developTempFile\appeon\_workarounds* folder under the PowerServer installation directory. For example, *C:\inetpub\wwwroot\appeon\developTempFile\appeon\_workarounds*.

To use the objects provided in the Appeon Cordova PBL, you will need to add the **appeon\_cordova.pbl** and **appeon\_workarounds.pbl** library to the **Library Search Path** of the application.

### 3.2.2 eon\_cordova\_base object

The eon\_cordova\_base object defines the basic and common functions/events that will be called by the other objects.

#### 3.2.2.1 Functions

##### 3.2.2.1.1 of\_getlasterror

###### Description

Gets the error information of the last function call.

###### Syntax

```
of_getlasterror()
```

###### Return value

String. The error information of the last call.

##### 3.2.2.1.2 of\_getlastreturn

###### Description

Gets the return value of the last function call if that call returns any value.

###### Syntax

```
of_getlastreturn()
```

###### Return value

String. The return value of the last call.

##### 3.2.2.1.3 of\_getvaluefromkey

###### Description

Gets the value from a JSON-format string for the specified key name.

###### Syntax

```
of_getvaluefromkey( string as_json, string as_key )
```

## Parameter

*as\_json* - A JSON-format string.

*as\_key* - Key name. It is case sensitive.

## Return value

String. Key value of the specified key.

### 3.2.2.1.4 of\_init

#### Description

Connects with the Cordova plugin, detects if the plugin is available to call, and binds the current PowerBuilder object with the JavaScript object defined by the plugin. After that, the PowerBuilder object has all of the methods and properties that the JavaScript object has.

#### Syntax

```
of_init()  
of_init( string as_pluginname )
```

#### Parameter

*as\_pluginname* - Name of the JavaScript object defined by the plugin.

#### Return value

Integer.

1 - Success.

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

### 3.2.2.1.5 of\_isios

#### Description

Detects if the current platform is iOS or Android.

#### Syntax

```
of_isios()
```

#### Return value

Boolean.

True - iOS

False - Android

Null - It is called in PowerBuilder or PowerServer Web, or there is an error.

### 3.2.2.1.6 of\_register

#### Description

Registers the object and the event to be bound with the oe\_success event and the oe\_error event.

#### Syntax

```
of_register ( powerobject apb_bind, string as_sucessevent, string  
as_errorevent )
```

**Parameter**

*apb\_bind* - The object to be bound with the oe\_error and oe\_success event.

*as\_sucessevent* - The event to be bound with the oe\_success event.

*as\_errorevent* - The event to be bound with the oe\_error event.

**Return value**

Integer.

1 - Success.

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

**3.2.2.1.7 of\_settimeout****Description**

Sets the timeout value for the function call.

**Syntax**

```
of_settimeout ( long al_timeout )
```

**Parameter**

*al\_timeout* - Milliseconds for the timeout value.

**Return value**

None

**3.2.2.2 Events****3.2.2.2.1 oe\_error****Description**

Occurs when an error occurs during the function call. If the JavaScript function in the Cordova plugin returns any error, that error will be passed to the *as\_error* parameter.

**Syntax**

```
oe_error ( string as_error )
```

**Parameter**

*as\_error* - A JSON-format string which contains error information returned by the function.

**Return value**

None

**3.2.2.2.2 oe\_execjserror****Description**

Occurs when there is a syntax error during the function call. However, this event cannot capture all of the errors.

### Syntax

`oe_execjserror ( string as_error )`

### Parameter

`as_error` - The error information of the last function call. It can also be obtained via [of\\_getlasterror](#) function.

### Return value

None

### 3.2.2.2.3 oe\_success

#### Description

Occurs when the function is successfully executed. If the JavaScript function in the Cordova plugin returns any value, that value will be passed to the `as_message` parameter.

### Syntax

`oe_success ( string as_message )`

### Parameter

`as_message` - A JSON-format string which contains values returned by the function.

### Return value

None

### 3.2.2.3 Properties

*oleobject ieon\_ole*

PowerBuilder OLEObject object to be connected with the Cordova plugin.

*powerobject ipo\_bindevent*

PowerBuilder object to be bound with the JavaScript object.

*string is\_errorText*

Stores the error information returned by the JavaScript function when execution failed.

*string is\_successText*

Stores the value returned by the JavaScript function when execution is successful.

*string is\_jserrorText*

Stores the JavaScript error information when the JavaScript function fails.

*string is\_sucesssevent*

Stores the success event name of the PowerBuilder object.

*string is\_errorevent*

Stores the error event name of the PowerBuilder object.

### 3.2.3 eon\_cordova\_bluetooth object (Android only)

This object enables communication between a mobile device (currently only Android-based device) and the bluetooth devices.

#### 3.2.3.1 Functions

##### 3.2.3.1.1 of\_disable

###### Description

Turns off bluetooth on the device. When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered.

###### Syntax

```
of_disable()
```

###### Return value

None

##### 3.2.3.1.2 of\_enable

###### Description

Turns on bluetooth on the device. When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered.

###### Syntax

```
of_enable()
```

###### Return value

None

##### 3.2.3.1.3 of\_init

###### Description

Connects with the Cordova bluetooth plugin, detects if the plugin is available to call, and binds the current PowerBuilder object with the JavaScript object defined by the plugin. After that, the PowerBuilder object has all of the methods and properties that the JavaScript object has.

###### Syntax

```
of_init()
```

###### Return value

Integer.

1 - Success.

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

##### 3.2.3.1.4 of\_isenabled

###### Description

Detects if bluetooth is turned on or not on the device.

### Syntax

```
of_isenabled()
```

### Return value

Boolean

True - Bluetooth is turned on.

False - Bluetooth is turned off.

### 3.2.3.1.5 of\_pair

#### Description

Pairs your device with a bluetooth device. When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered.

### Syntax

```
of_pair ( string as_address )
```

### Parameter

*as\_address* - The address of the bluetooth device you want to pair with your device.

### Return value

None

### 3.2.3.1.6 of\_startdiscovery

#### Description

Scans for the available bluetooth devices. When successful, the oe\_success event will be triggered and the found device(s) will be passed to the event; when failed, the oe\_error event will be triggered.

### Syntax

```
of_startdiscovery()
```

### Return value

None

### 3.2.3.1.7 of\_unpair

#### Description

Unpairs a bluetooth device with your device. When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered.

### Syntax

```
of_unpair ( string as_address )
```

### Parameter

*as\_address* - The address of the bluetooth device you want to unpair with your device.

## Return value

None

### 3.2.3.2 Events

#### 3.2.3.2.1 oe\_finish

##### Description

Occurs when scanning for bluetooth devices is successful. It can be used to scan for bluetooth devices in a loop.

##### Syntax

```
oe_finish ( string as_message )
```

##### Parameter

*as\_message* - A JSON-format string which contains values returned by the function such as the address and name of the found bluetooth device.

##### Return value

None

#### 3.2.3.2.2 oe\_success

##### Description

Occurs when the function is successfully executed. If the JavaScript function in the Cordova plugin returns any value, that value will be passed to the *as\_message* parameter.

##### Syntax

```
oe_success ( string as_message )
```

##### Parameter

*as\_message* - A JSON-format string which contains values returned by the function such as the address and name of the found bluetooth device.

##### Return value

None

### 3.2.3.3 Properties

*boolean ib\_enabled*

Indicates whether bluetooth is turned on.

*boolean ib\_exist*

Indicates whether the device exists in the search list.

*boolean ib\_search*

Indicates whether search is successful.

*eon\_str\_cordova\_bluetooth ieon\_str\_bluetooth[]*

The structure for storing the address and name of the found bluetooth device.

*long il\_type*

Functionality extension for developers: 1 - isenabled 2 - search.

*oleobject ieon\_ole*

PowerBuilder OLEObject object to be connected with the Cordova plugin.

*powerobject ipo\_bindevent*

PowerBuilder object to be bound with the JavaScript object.

*string is\_errorevent*

Stores the error event name of the PowerBuilder object.

*string is\_errorText*

Stores the error information returned by the JavaScript function when execution failed.

*string is\_jserrorText*

Stores the JavaScript error information when JavaScript call fails.

*string is\_suksessevent*

Stores the success event name of the PowerBuilder object.

*string is\_successText*

Stores the value returned by the JavaScript function when execution is successful.

### **3.2.4 eon\_cordova\_bluetoothle object**

This object allows you to interact with Bluetooth Low Energy devices on Android and iOS mobile devices.

#### **3.2.4.1 Functions**

##### **3.2.4.1.1 of\_base64tojson**

###### **Description**

Converts a base64 encoded string to a JSON array.

###### **Syntax**

`of_base64tojson ( string as_data )`

###### **Parameter**

`as_data` - The base64 encoded string.

###### **Return value**

String. The decoded data.

##### **3.2.4.1.2 of\_characteristics**

###### **Description**

Discovers the service's characteristics. When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered.

### Syntax

```
of_characteristics ( string as_address, string as_service )
```

### Parameter

*as\_address* - The address/identifier provided by the scan's return object.

*as\_service* - Service UUID.

### Return value

None.

#### **3.2.4.1.3 of\_close**

### Description

Closes the connection with a bluetooth LE device. When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered.

### Syntax

```
of_close ( string as_address )
```

### Parameter

*as\_address* - The address/identifier provided by the scan's return object.

### Return value

None

#### **3.2.4.1.4 of\_close\_no\_event**

### Description

Closes the connection with a bluetooth LE device. No oe\_success or oe\_error event will be triggered.

### Syntax

```
of_close_no_event ( string as_address )
```

### Parameter

*as\_address* - The address/identifier provided by the scan's return object.

### Return value

None

#### **3.2.4.1.5 of\_connect**

### Description

Connects to a bluetooth LE device. When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered.

### Syntax

```
of_connect ( string as_address )
```

**Parameter**

*as\_address* - The address/identifier provided by the scan's return object.

**Return value**

None

**3.2.4.1.6 of\_decode****Description**

Decodes the data after it is read.

**Syntax**

```
of_decode ( string as_data )
```

**Parameter**

*as\_data* - The encoded data.

**Return value**

String. The decoded data.

**3.2.4.1.7 of\_disconnect****Description**

Disconnects from a bluetooth LE device. When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered.

**Syntax**

```
of_disconnect ( string as_address )
```

**Parameter**

*as\_address* - The address/identifier provided by the scan's return object.

**Return value**

None

**3.2.4.1.8 of\_discover****Description**

Discovers all the device's services, characteristics, and descriptors after connecting with a device successfully.

When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered.

Supported on Android only. Use [of\\_services](#) and [of\\_characteristics](#) instead on iOS.

**Syntax**

```
of_discover ( string as_address )
```

**Parameter**

*as\_address* - The address/identifier provided by the scan's return object.

### Return value

None

#### 3.2.4.1.9 of\_encode

### Description

Encodes the data before it is written to the device.

### Syntax

```
of_encode ( string as_data )
```

### Parameter

*as\_data* - The decoded data.

### Return value

String. The encoded data.

#### 3.2.4.1.10 of\_init

### Description

Connects with the Cordova bluetooth LE plugin, detects if the plugin is available to call, and binds the current PowerBuilder object with the JavaScript object defined by the plugin. After that, the PowerBuilder object has all of the methods and properties that the JavaScript object has.

### Syntax

```
of_init ()
```

### Return value

Integer.

1 - Success.

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

#### 3.2.4.1.11 of\_initialize

### Description

Initializes bluetooth on the device. This function must be called after of\_init and before any others and it should only be called once. When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered.

### Syntax

```
of_initialize ( boolean ab_request )
```

### Parameter

*ab\_request* - Whether to prompt the user to enable bluetooth.

### Return value

None

#### **3.2.4.1.12 of\_isconnected**

##### **Description**

Determines whether the device is connected.

##### **Syntax**

```
of_isconnected()
```

##### **Return value**

Boolean.

True - Connected.

False - Not connected.

#### **3.2.4.1.13 of\_isenabled**

##### **Description**

Detects if bluetooth is turned on or not on the device. The oe\_enable event will be triggered.

Supported on the Android device only.

##### **Syntax**

```
of_isenabled()
```

##### **Return value**

None.

#### **3.2.4.1.14 of\_read**

##### **Description**

Reads a particular service's characteristics. When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered.

##### **Syntax**

```
of_read( string as_address, string as_service, string as_characteristic )
```

##### **Parameter**

*as\_address* - The address/identifier provided by the scan's return object.

*as\_service* - The service's UUID.

*as\_characteristic* - The characteristic's UUID.

##### **Return value**

None

#### **3.2.4.1.15 of\_services**

##### **Description**

Discovers the device's services. The UUID of the service can be used by of\_characteristics.

When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered; and value returned by the JavaScript function will be passed to parameter of the event.

Supported on iOS only.

### Syntax

```
of_services ( string as_address )
```

### Parameter

*as\_address* - The address/identifier provided by the scan's return object.

### Return value

None

#### 3.2.4.1.16 of\_services\_no\_event

### Description

Discovers the device's services. The UUID of the service can be used by of\_characteristics.

Supported on iOS only.

### Syntax

```
of_services_no_event ( string as_address )
```

### Parameter

*as\_address* - The address/identifier provided by the scan's return object.

### Return value

String. The service UUID.

#### 3.2.4.1.17 of\_startscan

### Description

Scans for bluetooth LE devices. When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered.

### Syntax

```
of_startscan()
```

### Return value

None

#### 3.2.4.1.18 of\_stopscan

### Description

Stops scanning for bluetooth LE devices. When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered.

### Syntax

```
of_stopscan()
```

## Return value

None

### 3.2.4.1.19 of\_write

#### Description

Writes a particular service's characteristic. When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered.

#### Syntax

```
of_write ( string as_address, string as_service, string as_characteristic,  
string as_data )
```

#### Parameter

*as\_address* - The address/identifier provided by the scan's return object.

*as\_service* - The service's UUID.

*as\_characteristic* - The characteristic's UUID.

*as\_data* - Base64 encoded string.

#### Return value

None

## 3.2.4.2 Events

### 3.2.4.2.1 oe\_enable

#### Description

Occurs when the of\_isenabled function is executed; asks to turn on bluetooth if it is not turned on. If the JavaScript function in the Cordova plugin returns any value, that value will be passed to the *as\_message* parameter.

Supported on the Android device only.

#### Syntax

```
oe_enable ( string as_message )
```

#### Parameter

*as\_message* - A JSON-format string which contains values returned by the function.

#### Return value

None

### 3.2.4.2.2 oe\_error

#### Description

Occurs when an error occurs during the function call. If the JavaScript function in the Cordova plugin returns any error, that error will be passed to the *as\_error* parameter.

#### Syntax

`oe_error ( string as_error )`

### Parameter

`as_error` - A JSON-format string which contains error information returned by the function.

### Return value

None

### 3.2.4.2.3 oe\_success

### Description

Occurs when the function is successfully executed. If the JavaScript function in the Cordova plugin returns any value, that value will be passed to the `as_message` parameter.

### Syntax

`oe_success ( string as_message )`

### Parameter

`as_message` - A JSON-format string which contains values returned by the function.

### Return value

None

### 3.2.4.3 Properties

`boolean ib_enable`

Indicates whether bluetooth is enabled on the device.

`boolean ib_exist`

Indicates whether the newly scanned device exists in the scan result list.

`eon_str_cordova_bluetooth ieon_str_bluetooth[]`

The structure for storing the address and name of the found bluetooth device.

`long il_type`

Functionality extension for developers: 1 - scan, 2 - connect, 3 - read

`oleobject ieon_ole`

PowerBuilder OLEObject object to be connected with the Cordova plugin.

`powerobject ipo_bindevent`

PowerBuilder object to be bound with the JavaScript object.

`string is_connectinfo`

Stores the success or error information when establishing a connection.

`string is_errorevent`

Stores the error event name of the PowerBuilder object.

`string is_errorText`

Stores the error information returned by the JavaScript function when execution failed.

*string is\_jserrorText*

Stores the JavaScript error information when JavaScript call fails.

*string is\_sucessevent*

Stores the success event name of the PowerBuilder object.

*string is\_successText*

Stores the value returned by the JavaScript function when execution is successful.

#### 3.2.4.4 Structures

##### 3.2.4.4.1 eon\_str\_cordova\_bluetooth

###### Description

Struct.

Stores the information of the bluetooth device including address, name, status etc.

###### Property

Type	Variable Name	Description
string	address	Address of the bluetooth device.
string	name	Name of the bluetooth device.
string	rssi	RSSI of the bluetooth broadcasting.
string	advertisement	Advertisement of the bluetooth broadcasting.
string	status	Status of the bluetooth broadcasting.

#### 3.2.5 eon\_cordova\_contact object

This object allows you to access the device contacts database.

##### 3.2.5.1 Functions

###### 3.2.5.1.1 of\_addcontactaddress

###### Description

Adds an address to the JSON node. It will be used when creating a new contact.

###### Syntax

```
of_addcontactaddress ( string as_type, string as_formatted, string
as_streetaddress, string as_locality, string as_region, string
as_postalcode, string as_country )
```

## Parameter

*as\_type* - A string indicating what type of field this is, home for example.

*as\_formatted* - The full address formatted for display. It is unsupported on iOS.

*as\_streetaddress* - The full street address.

*as\_locality* - The city or locality.

*as\_region* - The state or region.

*as\_postalcode* - The zip code or postal code.

*as\_country* - The country name.

## Return value

None

### 3.2.5.1.2 of\_addcontactemail

#### Description

Adds an email to the JSON node. It will be used when creating a new contact.

#### Syntax

```
of_addcontactemail ( string as_email, string as_emailtype, boolean abl_emailpref )
```

#### Parameter

*as\_email* - Email address.

*as\_emailtype* - Email type. Values: home, work, mobile, iPhone, or any other value that is supported by a particular device platform's contact database.

*abl\_emailpref* - Whether used as the preference email. Values: true, false. It is unsupported on some platform.

#### Return value

None

### 3.2.5.1.3 of\_addcontactname

#### Description

Adds a contact name to the JSON node. It will be used when creating a new contact.

#### Syntax

```
of_addcontactname ( string as_contactname, string as_givenname, string as_familyname, string as_middlename, boolean abl_pref )
```

#### Parameter

*as\_contactname* - The contact's complete name you want to set.

*as\_givenname* - The contact's first name you want to set.

*as\_familyname* - The contact's last name you want to set.

*as\_middlename* - The contact's middle name you want to set.

*abl\_pref* - Whether used as the preference name. Values: true, false. It is unsupported on some platform.

### Return value

None

#### 3.2.5.1.4 of\_addcontactorganization

##### Description

Adds an organization to the JSON node. It will be used when creating a new contact.

##### Syntax

```
of_addcontactorganization ( string as_name, string as_department, string as_title, string as_type )
```

```
of_addcontactorganization ( string as_name, string as_department, string as_title, string as_type, boolean abl_pref )
```

##### Parameter

*as\_name* - The name of the organization.

*as\_department* - The department the contact works for.

*as\_title* - The contact's title at the organization.

*as\_type* - A string that indicates what type of field this is, home for example.

*abl\_pref* - Whether used as the preference organization. Values: true, false. It is unsupported on some platform.

### Return value

None

#### 3.2.5.1.5 of\_addcontactphonenumbers

##### Description

Adds a phone number to the JSON node. It will be used when creating a new contact.

##### Syntax

```
of_addcontactphonenumbers ( string as_phonenumber, string as_phonetyp, boolean abl_phoneref )
```

##### Parameter

*as\_phonenumber* - Phone number.

*as\_phonetyp* - Phone number type. Values: home, work, mobile, iPhone, or any other value that is supported by a particular device platform's contact database.

*abl\_phoneref* - Whether used as the preference number. It is unsupported on some platform.

### Return value

None

### **3.2.5.1.6 of\_addcontactphoto**

#### **Description**

Adds a photo to the JSON node. It will be used when creating a new contact.

#### **Syntax**

```
of_addcontactphoto ( string as_photopath, string as_phototype, boolean abl_photopref )
```

#### **Parameter**

*as\_photopath* - Photo URL/path.

*as\_phototype* - Format of the returned image: url (when the value attribute contains a URL to the photo image), or base64 (when the value contains a base64-encoded image string).

*abl\_photopref* - Whether used as the preference photo. It is unsupported on some platform.

#### **Return value**

None

### **3.2.5.1.7 of\_addcontacturls**

#### **Description**

Adds a contact URL to the JSON node. It will be used when creating a new contact.

#### **Syntax**

```
of_addcontacturls ( string as_value, string as_type, boolean abl_pref )
```

#### **Parameter**

*as\_value* - The URL to be added.

*as\_type* - The type of the URL.

*abl\_pref* - Whether used as the preference URL. It is unsupported on some platform.

#### **Return value**

None

### **3.2.5.1.8 of\_associatejswithpb**

#### **Description**

Associates the current PowerBuilder object with a JavaScript object.

#### **Syntax**

```
of_associatejswithpb ( string as_contactobjectname )
```

#### **Parameter**

*as\_contactobjectname* - Name of the JavaScript object.

## Return value

None

### 3.2.5.1.9 of\_clear\_jsonarray

#### Description

Clears the values of the JSON array.

#### Syntax

```
of_clear_jsonarray (eon_cjsonnodearray aeon_cjsonarray)
```

#### Parameter

*eon\_cjsonnodearray* - Name of the JSON array.

#### Return value

None

### 3.2.5.1.10 of\_create

#### Description

Creates a new contact and saves the contact to the device contacts database.

The contact information will be returned as a JSON string to the parameter of the oe\_success event.

#### Syntax

```
of_create ( string as_contactname, eon_str_cordova_contact_name  
astr_contact_name, eon_str_cordova_contact_address astr_address[],  
eon_str_cordova_contact_organization astr_organization[],  
eon_str_cordova_contact_urls astr_urls[], eon_str_cordova_contact_field  
astr_phone[], eon_str_cordova_contact_field astr_email[], string as_note )  
  
of_create ( string as_contactname, string as_givenname, string  
as_familyname, string as_middlename, string as_phonenumber, string  
as_phonenumbertype, string as_email, string as_emailtype )  
  
of_create ( string as_contactname, string as_phonenumber )  
  
of_create ( string as_contactname, string as_phonenumber, string  
as_phonenumbertype, string as_email, string as_emailtype )
```

#### Parameter

*as\_contactname* - Name of the new contact.

*astr\_contact\_name* - Structure for the contact name information such as first name, last name, middle name etc.

*astr\_address[]* - Structure array for the contact address information such as multiple addresses.

*astr\_organization[]* - Structure array for the contact organization information such as multiple organizations.

*astr\_urls[]* - Structure array for the contact URL such as multiple URLs.

*astr\_phone[ ]* - Structure array for the contact phone number such as smart phone number, home phone number etc.

*astr\_email[ ]* - Structure array for the contact email such as multiple email addresses.

*as\_note* - Note of the new contact.

*as\_givenname* - First name of the new contact.

*as\_familynname* - Last name of the new contact.

*as\_middlename* - Middle name of the new contact.

*as\_phonenumber* - Phone number of the new contact.

*as\_phoneNumberType* - Phone number type. Values: home, work, fax, mobile.

*as\_email* - Email address of the new contact.

*as\_emailType* - Email type of the new contact.

### **Return value**

String.

0: Success

error text: Error returned when failing to create the contact.

### **3.2.5.1.11 of\_create\_jsonpar**

#### **Description**

Creates a contact using a JSON-format string and saves the information.

The contact information will be returned as a JSON string to the parameter of the oe\_success event.

#### **Syntax**

`of_create_jsonpar ( string as_contact_infor )`

#### **Parameter**

*as\_contact\_infor* - A JSON-format string that contains the contact information.

### **Return value**

String.

0: Success

error text: Error returned when failing to create the contact.

### **3.2.5.1.12 of\_delete**

#### **Description**

Deletes the contact according to the specified name or phone number.

#### **Syntax**

`of_delete ( string as_nameorphone )`

#### **Parameter**

*as\_nameorphone* - Name or phone number of the contact to be deleted. If there are multiple contacts found, only the first one on the list will be deleted.

**Return value**

Integer.

1 - Success.

-1 - Failure (due to errors or other problems, or failing to find the contact).

0 - Not found.

**3.2.5.1.13 of\_delete\_contactid****Description**

Deletes the contact according to the specified contact ID.

**Syntax**

```
of_delete_contactid( string as_id )
```

**Parameter**

*as\_id* - The unique ID of the contact to be deleted.

**Return value**

Integer.

1 - Success.

-1 - Failure (due to errors or other problems, or failing to find the contact).

0 - Not found.

**3.2.5.1.14 of\_destroy\_object****Description**

Destroys the JSON object.

**Syntax**

```
of_destroy_object()
```

**Return value**

None

**3.2.5.1.15 of\_find****Description**

Finds a contact.

**Syntax 1**

```
of_find( string as_contactfield, string as_contactopt_filter, boolean  
        abl_contactopt_multiple, string as_contactopt_desiredfields, boolean  
        abl_contactopt_hasphonenumber )
```

**Parameter 1**

*as\_contactfield* - The contact fields used as a search qualifier.

*as\_contactopt\_filter* - The search string used to find navigator.contacts.

*abl\_contactopt\_multiple* - Whether the find operation returns multiple navigator.contacts (default: false).

*as\_contactopt\_desiredfields* - The contact fields to be returned. If specified, the returned result only contains values for these fields.

*abl\_contactopt\_hasphonenumber* - Search only returns contacts with a phone number. It is supported on Android only.

## Return value 1

String. Returns a JSON-format string with the found contacts.

## Syntax 2

```
of_find( string as_contactopt_filter )
```

## Parameter 2

*as\_contactopt\_filter* - The search string used to find navigator.contacts.

## Return value 2

Any. eon\_str\_cordova\_contact\_find[]: if successful, the structure stores the contact name, ID, note, URL, phone number, email, address, organization, and photo; if not found, eon\_str\_cordova\_contact\_find.s\_errortext = notfind; if error occurs, eon\_str\_cordova\_contact\_find.s\_errortext = error.

### 3.2.5.1.16 of\_find\_contactname

#### Description

Finds a contact's name, ID, display name, and nick name.

#### Syntax

```
of_find_contactname ( string as_contactopt_filter )
```

#### Parameter

*as\_contactopt\_filter* - The search string used to find navigator.contacts.

#### Return value

Any. eon\_str\_cordova\_contact\_find[]: if successful, the structure stores the contact name, phone number, email, address, organization, and photo; if not found, eon\_str\_cordova\_contact\_find.s\_errortext = notfind; if error occurs, eon\_str\_cordova\_contact\_find.s\_errortext = error.

### 3.2.5.1.17 of\_find\_event

#### Description

Finds a contact. When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered; and the return value will be passed to the parameter of the event.

#### Syntax

```
of_find_event ( string as_contactfield, string as_contactopt_filter )
of_find_event ( string as_contactfield, string as_contactopt_filter,
boolean abl_contactopt_multiple, string as_contactopt_desiredfields,
boolean abl_contactopt_hasphonenumber )
of_find_event ( string as_contactfield, string as_contactopt_filter,
string as_contactopt_desiredfields )
```

### Parameter

*as\_contactfield* - The contact fields used as a search qualifier.

*as\_contactopt\_filter* - The search string used to find navigator.contacts.

*abl\_contactopt\_multiple* - Whether the find operation returns multiple navigator.contacts (default: false).

*as\_contactopt\_desiredfields* - The contact fields to be returned. If specified, the returned result only contains values for these fields.

*abl\_contactopt\_hasphonenumber* - Search only returns contacts with a phone number. It is supported on Android only.

### Return value

None.

#### 3.2.5.1.18 of\_getcontactfieldtype

### Description

Gets the contact field type. This can get all of the properties of a contact.

### Syntax

```
of_getcontactfieldtype()
```

### Return value

String.

#### 3.2.5.1.19 of\_getphoto\_android

### Description

Gets the photo from the device's contact database and creates a copy under the PowerServer application folder.

Supported on Android only.

### Syntax

```
of_getphoto_android ( string as_photo )
```

### Parameter

*as\_photo* - The photo in the device's contact database (obtained from eon\_str\_cordova\_contact\_find[]).

### Return value

String. The directory of the photo in the plugin folder of the PowerServer application.

### 3.2.5.1.20 of\_init

#### Description

Connects with the Cordova contacts plugin, detects if the plugin is available to call, and binds the current PowerBuilder object with the JavaScript object defined by the plugin. After that, the PowerBuilder object has all of the methods and properties that the JavaScript object has.

#### Syntax

```
of_init()
```

#### Return value

Integer.

1 - Success.

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

### 3.2.5.1.21 of\_init\_jsonobject

#### Description

Creates a JSON object, to help add the contact's property.

#### Syntax

```
of_init_jsonobject()
```

#### Return value

None.

### 3.2.5.1.22 of\_remove

#### Description

Deletes a contact. When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered; and the return value will be passed to the parameter of the event.

#### Syntax

```
of_remove( string as_contact_id )
```

#### Parameter

*as\_contact\_id* - ID of the contact that you want to remove.

#### Return value

None.

### 3.2.5.1.23 of\_save

#### Description

Saves the newly created contact to the device. When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered; and the return value will be passed to the parameter of the event.

#### Syntax

```
of_save ( string as_contactobjectname )
```

**Parameter**

*as\_contactobjectname* - Name of the contact object.

**Return value**

None.

**3.2.5.1.24 of\_setcontactaddress****Description**

Creates a contactaddress JSON node and sets the value.

**Syntax**

```
of_setcontactaddress ( string as_type, string as_formatted, string  
as_streetaddress, string as_locality, string as_region, string  
as_postalcode, string as_country )
```

**Parameter**

*as\_type* - A string indicating what type of field this is, home for example.

*as\_formatted* - The full address formatted for display.

*as\_streetaddress* - The full street address.

*as\_locality* - The city or locality.

*as\_region* - The state or region.

*as\_postalcode* - The zip code or postal code.

*as\_country* - The country name.

**Return value**

*eon\_cjsonnode*. The JSON node with the values.

**3.2.5.1.25 of\_setcontactfield****Description**

Creates a contactfield JSON node and sets the value.

**Syntax**

```
of_setcontactfield ( string as_value )
```

```
of_setcontactfield ( string as_value, string as_type, boolean abl_pref )
```

**Parameter**

*as\_value* - The value of the field, such as a phone number or email address.

*as\_type* - In most instances, there are no pre-determined values for this parameter. For example, a phone number can specify type values of home, work, mobile, iPhone, or any other value that is supported by a particular device platform's contact database. However, for the Contact photos field, the type indicates the format of the returned image: url when the value attribute contains a URL to the photo image, or base64 when the value contains a base64-encoded image string.

*abl\_pref* - Whether used as the preference value. It is unsupported on some platform.

### Return value

eon\_cjsonnode. The JSON node with the values.

#### 3.2.5.1.26 of\_setcontactname

##### Description

Sets the contact's name.

##### Syntax 1

```
of_setcontactname ( string as_contactname )
```

##### Parameter 1

*as\_contactname* - The contact's name you want to set.

##### Return value 1

None.

##### Syntax 2

```
of_setcontactname ( string as_givenname, string as_familyname, string  
as_formatted, string as_middlename, string as_honorificprefix, string  
as_honorificsuffix, boolean abl_pref )
```

Overloads the of\_setcontactname function for adding the other name related information.

##### Parameter 2

*as\_givenname* - The contact's first name you want to set.

*as\_familyname* - The contact's last name you want to set.

*as\_formatted* - The contact's complete name you want to set.

*as\_middlename* - The contact's middle name you want to set.

*as\_honorificprefix* - The contact's prefix (example Mr. or Dr.)

*as\_honorificsuffix* - The contact's suffix (example Esq.).

*abl\_pref* - Whether used as the preference name. Values: true, false. It is unsupported on some platform.

##### Return value 2

eon\_cjsonnode. Returns the JSON node with the values.

#### 3.2.5.1.27 of\_setcontactoptions

##### Description

Sets the options' value for the find parameter.

##### Syntax

```
of_setcontactoptions ( string as_contactopt_filter, boolean  
abl_contactopt_multiple, string as_contactopt_desiredfields, boolean  
abl_contactopt_hasphonenumer )
```

## Parameter

*as\_contactopt\_filter* - The string used to find navigator.contacts.

*abl\_contactopt\_multiple* - Whether the find operation returns multiple navigator.contacts (default: false).

*as\_contactopt\_desiredfields* - Contact fields to be returned.

*abl\_contactopt\_hasphonenumber* - (Android only) Search only returns contacts with a phone number (default: false).

## Return value

String. Returns the JSON string with the options.

### 3.2.5.1.28 of\_setcontactorganization

#### Description

Creates a contactorganization JSON node and sets the value.

#### Syntax

```
of_setcontactorganization ( string as_name, string as_department, string as_title, string as_type )
```

#### Parameter

*as\_name* - The name of the organization. It is partially supported on iOS.

*as\_department* - The department the contact works for. It is partially supported on iOS.

*as\_title* - The contact's title at the organization. It is partially supported on iOS.

*as\_type* - A string that indicates what type of field this is, home for example. It is unsupported on iOS.

#### Return value

*eon\_cjsonnode*. The JSON node with the values.

### 3.2.5.2 Properties

*eon\_cjsonnode iec\_contactname*

The JSON node object for adding the contact's name.

*eon\_cjsonnode iec\_contactpro*

The JSON node object for adding the contact's properties.

*eon\_cjsonnodearray iec\_addressarray*

The JSON array object for adding the address information.

*eon\_cjsonnodearray iec\_emailsarray*

The JSON array object for adding the email information.

*eon\_cjsonnodearray iec\_organizationarray*

The JSON array object for adding the organization information.

*eon\_cjsonnodearray iec\_phonenumberarray*

The JSON array object for adding the phone number information.

*eon\_cjsonnodearray iec\_photoarray*

The JSON array object for adding the photo information.

*eon\_cjsonnodearray iec\_urlsarray*

The JSON array object for adding the URL.

*oleobject ieon\_ole*

PowerBuilder OLEObject object to be connected with the Cordova plugin.

*powerobject ipo\_bindevent*

PowerBuilder object to be bound with the JavaScript object.

*string is\_errorevent*

Stores the error event name of the PowerBuilder object.

*string is\_errortext*

Stores the error information returned by the JavaScript function when execution failed.

*string is\_jserrortext*

Stores the JavaScript error information when JavaScript call fails.

*string is\_navigatorcontacts*

Stores the object name of the Contact plugin.

*string is\_sucessevent*

Stores the success event name of the PowerBuilder object.

*string is\_sucessstext*

Stores the value returned by the JavaScript function when execution is successful.

### 3.2.5.3 Structures

#### 3.2.5.3.1 eon\_str\_cordova\_contact\_address

##### Description

Struct.

Stores the information of the contact address.

##### Property

Type	Variable Name	Description
string	s_type	Type of the field, home for example.
string	s_streetaddress	The street address.
string	s_locality	The city or locality.
string	s_region	The state or region.
string	s_postalcode	The zip code or postal code.

Type	Variable Name	Description
string	s_country	The country name.
string	s_formatted	The complete address.

### 3.2.5.3.2 eon\_str\_cordova\_contact\_field

#### Description

Struct.

Stores the information of the contact field.

#### Property

Type	Variable Name	Description
string	s_type	Type of the contact field.
string	s_value	Value of the contact field.

### 3.2.5.3.3 eon\_str\_cordova\_contact\_find

#### Description

Struct.

Stores the information of the found contact.

#### Property

Type	Variable Name	Description
string	s_contactname	Name of the found contact.
string	id	Unique ID of the found contact.
eon_str_cordova_contact_field	str_phonenumbers[ ]	Phone numbers of the found contact.
eon_str_cordova_contact_field	str_emails[ ]	Emails of the found contact.
string	s_photos[ ]	Photos of the found contact.
eon_str_cordova_contact_address	str_address[ ]	Address of the found contact.
eon_str_cordova_contact_organization	str_organizations[ ]	Organization of the found contact.
string	s_errortext	Error text.
date	d_birthday	Birthday of the found contact.
eon_str_cordova_contact_name	str_contactname	Name of the found contact.
string	s_note	Note of the found contact.
eon_str_cordova_contact_urls	str_urls[ ]	URLs of the found contact.

### 3.2.5.3.4 eon\_str\_cordova\_contact\_name

#### Description

Struct.

Stores the information of the contact name.

### Property

Type	Variable Name	Description
string	s_givenname	First name of the contact.
string	s_formatted	Complete name of the contact.
string	s_middleName	Middle name of the contact.
string	s_familyname	Last name of the contact.
string	s_honorificPrefix	Prefix (such as Mr. or Dr.) of the contact.
string	s_honorificSuffix	Suffix (such as Esq.) of the contact.

### 3.2.5.3.5 eon\_str\_cordova\_contact\_organization

#### Description

Struct.

Stores the information of the contact organization.

#### Property

Type	Variable Name	Description
string	s_name	The name of the organization.
string	s_department	The department the contact works for.
string	s_title	The contact's title at the organization.
string	s_type	Type of the field, home for example.

### 3.2.5.3.6 eon\_str\_cordova\_contact\_urls

#### Description

Struct.

Stores the information of the contact URL.

#### Property

Type	Variable Name	Description
string	s_value	The URL.
string	s_type	The type of the URL.

### **3.2.6 eon\_cordova\_fingerprint object**

On the Android-based devices, this object will open a native dialog prompting the user to authenticate using their fingerprint.

On the iOS devices, this object will scan the fingerprint of your user with the TouchID sensor.

#### **3.2.6.1 Functions**

##### **3.2.6.1.1 of\_init**

###### **Description**

Connects with the Cordova fingerprint plugin, detects if the plugin is available to call, and binds the current PowerBuilder object with the JavaScript object defined by the plugin. After that, the PowerBuilder object has all of the methods and properties that the JavaScript object has.

###### **Syntax**

`of_init()`

###### **Return value**

Integer.

1 - Success.

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

##### **3.2.6.1.2 of\_isavailable**

###### **Description**

Detects if the fingerprint/TouchID sensor works on the device.

###### **Syntax**

`of_isavailable()`

###### **Return value**

Boolean.

True - Fingerprint/TouchID sensor is working.

False - Fingerprint/TouchID sensor is not working.

##### **3.2.6.1.3 of\_show**

###### **Description**

Opens a native dialog to use the device hardware fingerprint scanner to authenticate against fingerprints registered for the device. When the function call is successful, oe\_success event will be triggered; when the function call failed, oe\_error event will be triggered.

Supported on Android only.

###### **Syntax**

`of_show()`

**Return value**

None.

**3.2.6.1.4 of\_verifyfingerprint\_ios****Description**

Verifies the fingerprint function. When the function call is successful, oe\_success event will be triggered; when the function call failed, oe\_error event will be triggered.

Supported on iOS only.

**Syntax**

```
of_verifyfingerprint_ios()
```

**Return value**

None.

**3.2.6.2 Events****3.2.6.2.1 oe\_error****Description**

Occurs when an error occurs during the function call. If the JavaScript function in the Cordova plugin returns any error, that error will be passed to the *as\_error* parameter.

**Syntax**

```
oe_error ( string as_error )
```

**Parameter**

*as\_error* - A JSON-format string which contains error information returned by the function.

**Return value**

None

**3.2.6.2.2 oe\_success****Description**

Occurs when the function is successfully executed. If the JavaScript function in the Cordova plugin returns any value, that value will be passed to the *as\_message* parameter.

**Syntax**

```
oe_success ( string as_message )
```

**Parameter**

*as\_message* - A JSON-format string which contains values returned by the function. If the function called is of\_isavailable, the return value indicates the fingerprint sensor is working or not.

**Return value**

None

### 3.2.6.3 Properties

*boolean ib\_ios*

Indicates whether the platform is iOS.

*oleobject ieon\_ole*

PowerBuilder OLEObject object to be connected with the Cordova plugin.

*powerobject ipo\_bindevent*

PowerBuilder object to be bound with the JavaScript object.

*string is\_errorevent*

Stores the error event name of the PowerBuilder object.

*string is\_errorText*

Stores the error information returned by the JavaScript function when execution failed.

*string is\_jserrorText*

Stores the JavaScript error information when JavaScript call fails.

*string is\_sucessevent*

Stores the success event name of the PowerBuilder object.

*string is\_successText*

Stores the value returned by the JavaScript function when execution is successful.

### 3.2.7 eon\_cordova\_gps object

This object provides information about the device's location, such as latitude and longitude.

#### 3.2.7.1 Functions

##### 3.2.7.1.1 of\_configure

###### Description

Configures the parameters for the GPS search. When successful, the oe\_success event will be triggered; when failed, the oe\_error event will be triggered.

###### Syntax

`of_configure( eon_str_cordova_gps astr_gps )`

###### Parameter

*astr\_gps* - This structure carries the following values:

- *desiredaccuracy* - The desired accuracy of the geolocation system. Values: 0 (highest power, highest accuracy), 10, 100, 1000 (lowest power, lowest accuracy).
- *stationaryRadius* - When stopped, the minimum distance the device must move beyond the stationary location for aggressive background-tracking to engage.
- *distanceFilter* - The minimum distance (measured in meters) a device must move horizontally before an update event is generated.

## Return value

None.

### 3.2.7.1.2 of\_init

#### Description

Connects with the Cordova geolocation plugin, detects if the plugin is available to call, and binds the current PowerBuilder object with the JavaScript object defined by the plugin. After that, the PowerBuilder object has all of the methods and properties that the JavaScript object has.

#### Syntax

```
of_init()
```

#### Return value

Integer.

1 - Success.

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

### 3.2.7.1.3 of\_start

#### Description

Enables location tracking.

#### Syntax

```
of_start()
```

#### Return value

None.

### 3.2.7.1.4 of\_stop

#### Description

Disables location tracking.

#### Syntax

```
of_stop()
```

#### Return value

None.

## 3.2.7.2 Events

### 3.2.7.2.1 oe\_success

#### Description

Occurs when the function is successfully executed. If the JavaScript function in the Cordova plugin returns any value, that value will be passed to the *as\_message* parameter.

#### Syntax

`oe_success ( string as_message )`

### Parameter

`as_message` - A JSON-format string which contains the location data.

### Return value

None

### 3.2.7.3 Properties

`eon_str_cordova_gps_info ieon_gps_info`

The structure for storing the detailed location data.

`oleobject ieon_ole`

PowerBuilder OLEObject object to be connected with the Cordova plugin.

`powerobject ipo_bindevent`

PowerBuilder object to be bound with the JavaScript object.

`string is_errorevent`

Stores the error event name of the PowerBuilder object.

`string is_errorText`

Stores the error information returned by the JavaScript function when execution failed.

`string is_jserrorText`

Stores the JavaScript error information when JavaScript call fails.

`string is_sucessevent`

Stores the success event name of the PowerBuilder object.

`string is_successText`

Stores the value returned by the JavaScript function when execution is successful.

### 3.2.7.4 Structures

#### 3.2.7.4.1 eon\_str\_cordova\_gps

##### Description

Struct.

Stores the information of the device GPS.

##### Property

Type	Variable Name	Description
long	desiredaccuracy	The desired accuracy of the geolocation system.
long	stationaryradius	When stopped, the minimum distance the device must move beyond the stationary

Type	Variable Name	Description
		location for aggressive background-tracking to engage.
long	distancefilter	The minimum distance (measured in meters) a device must move horizontally before an update event is generated.

### 3.2.7.4.2 eon\_str\_cordova\_gps\_info

#### Description

Struct.

Stores the information of the device's location.

#### Property

Type	Variable Name	Description
long	time	The timestamp of the location.
decimal { 2 }	speed	The speed.
decimal { 2 }	altitude	The altitude of the location.
decimal { 2 }	bearing	The bearing of the location
decimal { 2 }	longitude	The longitude of the location.
decimal { 2 }	latitude	The latitude of the location.
decimal { 2 }	accuracy	The accuracy of the location.
string	serviceprovider	The GPS service provider.
boolean	debug	

### 3.2.8 eon\_cordova\_orientation object

This object provides access to the device's compass. The compass is a sensor that detects the direction or heading that the device is pointed, typically from the top of the device.

#### 3.2.8.1 Functions

##### 3.2.8.1.1 of\_clearwatch

#### Description

Stops watching the device's heading according to the specified watch ID.

#### Syntax

```
of_clearwatch( string as_watchid )
```

#### Parameter

*as\_watchid* - Watch ID returned by `of_watchheading`.

### Return value

None

#### 3.2.8.1.2 `of_getcurrentheading`

### Description

Gets the device's current heading/direction. When successful, the `oe_success` event will be triggered; when failed, the `oe_error` event will be triggered; and the return value will be passed to the parameter of the event.

### Syntax

```
of_getcurrentheading()
```

### Return value

None.

#### 3.2.8.1.3 `of_init`

### Description

Connects with the Cordova device orientation plugin, detects if the plugin is available to call, and binds the current PowerBuilder object with the JavaScript object defined by the plugin. After that, the PowerBuilder object has all of the methods and properties that the JavaScript object has.

### Syntax

```
of_init()
```

### Return value

Integer.

1 - Success.

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

#### 3.2.8.1.4 `of_watchheading`

### Description

Gets the device's current heading/direction at a regular interval.

### Syntax

```
of_watchheading( long al_frequency, decimal ad_filter )
```

### Parameter

*al\_frequency* - How often to retrieve the device's heading in milliseconds (default: 100)

*ad\_filter* - The change in degrees required to retrieve the heading. It is supported in iOS only. When this parameter is set, *al\_frequency* is ignored.

### Return value

String. Watch ID is returned and can be used by [of\\_clearwatch](#).

### 3.2.8.2 Events

#### 3.2.8.2.1 oe\_success

##### Description

Occurs when the function is successfully executed. If the JavaScript function in the Cordova plugin returns any value, that value will be passed to the *as\_message* parameter.

##### Syntax

```
oe_success ( string as_message )
```

##### Parameter

*as\_message* - A JSON-format string which contains the heading information.

##### Return value

None

### 3.2.8.3 Properties

*eon\_str\_cordova\_orientation ieon\_str\_orientation*

The structure for storing the detailed heading information.

*oleobject ieon\_ole*

PowerBuilder OLEObject object to be connected with the Cordova plugin.

*powerobject ipo\_bindevent*

PowerBuilder object to be bound with the JavaScript object.

*string is\_errorevent*

Stores the error event name of the PowerBuilder object.

*string is\_errorText*

Stores the error information returned by the JavaScript function when execution failed.

*string is\_jserrorText*

Stores the JavaScript error information when JavaScript call fails.

*string is\_sucessevent*

Stores the success event name of the PowerBuilder object.

*string is\_successText*

Stores the value returned by the JavaScript function when execution is successful.

### 3.2.8.4 Structures

#### 3.2.8.4.1 eon\_str\_cordova\_orientation

##### Description

Struct.

Stores the information of the device orientation.

## Property

Type	Variable Name	Description
decimal	magneticHeading	The device's magnetic heading.
decimal	trueHeading	The device's true heading.
decimal	HeadingAccuracy	The device's heading accuracy.
decimal	timestamp	The device's timestamp.

## 3.2.9 eon\_cordova\_vibration object

This object provides a way to vibrate the device.

### 3.2.9.1 Functions

#### 3.2.9.1.1 of\_init

##### Description

Connects with the Cordova vibration plugin, detects if the plugin is available to call, and binds the current PowerBuilder object with the JavaScript object defined by the plugin. After that, the PowerBuilder object has all of the methods and properties that the JavaScript object has.

##### Syntax

```
of_init()
```

##### Return value

Integer.

1 - Success.

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

#### 3.2.9.1.2 of\_vibrate

##### Description

Vibrates the device for a given amount of time.

##### Syntax

```
of_vibrate( long al_time )
```

```
of_vibrate( long al_time[] ) (Supported on Android only)
```

##### Parameter

*al\_time* - Milliseconds to vibrate the device.

*al\_time[]* - Sequence of durations (in milliseconds) for which to turn on or off the vibrator.

##### Return value

None

### 3.2.9.2 Properties

*oleobject ieon\_ole*

PowerBuilder OLEObject object to be connected with the Cordova plugin.

*powerobject ipo\_bindevent*

PowerBuilder object to be bound with the JavaScript object.

*string is\_errorevent*

Stores the error event name of the PowerBuilder object.

*string is\_errorText*

Stores the error information returned by the JavaScript function when execution failed.

*string is\_jserrorText*

Stores the JavaScript error information when JavaScript call fails.

*string is\_sucessevent*

Stores the success event name of the PowerBuilder object.

*string is\_successText*

Stores the value returned by the JavaScript function when execution is successful.

## 3.3 Cordova plugins ([solution 2 \[207\]](#))

PowerServer Mobile has packaged the following most popular Cordova plugins according to the download times on the plugin website. If you know the basics of JavaScript language and can understand the JavaScript code in the plugin, you can access these plugins using a special object Appeon creates -- **AppeonMobile.CordovaPlugin**.

Table 3.1:

Plugin Type	Name	Version	Download Page
Contacts	cordova-plugin-contacts	2.0.1	<a href="https://www.npmjs.com/package/cordova-plugin-contacts">https://www.npmjs.com/package/cordova-plugin-contacts</a>
Bluetooth	org.apache.cordova.bluetooth	0.0.9th	<a href="https://github.com/tanelih/phonegap-bluetooth-plugin">https://github.com/tanelih/phonegap-bluetooth-plugin</a>
Calendar	cordova-plugin-calendar	4.4.7	<a href="https://www.npmjs.com/package/cordova-plugin-calendar">https://www.npmjs.com/package/cordova-plugin-calendar</a>
File	cordova-plugin-file	4.1.1	<a href="https://www.npmjs.com/package/cordova-plugin-file">https://www.npmjs.com/package/cordova-plugin-file</a>
Api.ai voice-enabling engine	cordova-plugin-apiai	1.7.0	<a href="https://www.npmjs.com/package/cordova-plugin-apiai">https://www.npmjs.com/package/cordova-plugin-apiai</a>
Device	cordova-plugin-device	1.1.1	<a href="https://www.npmjs.com/package/cordova-plugin-device">https://www.npmjs.com/package/cordova-plugin-device</a>

Plugin Type	Name	Version	Download Page
Network information	cordova-plugin-network-information	1.2.0	<a href="https://www.npmjs.com/package/cordova-plugin-network-information">https://www.npmjs.com/package/cordova-plugin-network-information</a>
Globalization	cordova-plugin-globalization	1.0.3	<a href="https://www.npmjs.com/package/cordova-plugin-globalization">https://www.npmjs.com/package/cordova-plugin-globalization</a>
Geolocation	cordova-plugin-mauron85-background-geolocation	0.9.6	<a href="https://www.npmjs.com/package/cordova-plugin-mauron85-background-geolocation">https://www.npmjs.com/package/cordova-plugin-mauron85-background-geolocation</a>
Device orientation	cordova-plugin-device-orientation	1.0.3	<a href="https://www.npmjs.com/package/cordova-plugin-device-orientation">https://www.npmjs.com/package/cordova-plugin-device-orientation</a>
Media	cordova-plugin-media-capture	1.2.0	<a href="https://www.npmjs.com/package/cordova-plugin-media-capture">https://www.npmjs.com/package/cordova-plugin-media-capture</a>
Vibration	cordova-plugin-vibration	2.1.0	<a href="https://www.npmjs.com/package/cordova-plugin-vibration">https://www.npmjs.com/package/cordova-plugin-vibration</a>
Battery status	cordova-plugin-battery-status	1.1.2	<a href="https://www.npmjs.com/package/cordova-plugin-battery-status">https://www.npmjs.com/package/cordova-plugin-battery-status</a>
Sim	cordova-plugin-sim	1.2.1	<a href="https://www.npmjs.com/package/cordova-plugin-sim">https://www.npmjs.com/package/cordova-plugin-sim</a>
Fingerprint authentication	Android: cordova-plugin-android-fingerprint-auth iOS: cordova-plugin-touch-id	Android: 0.0.5 iOS: 3.1.0	Android: <a href="https://www.npmjs.com/package/cordova-plugin-android-fingerprint-auth">https://www.npmjs.com/package/cordova-plugin-android-fingerprint-auth</a> iOS: <a href="https://www.npmjs.com/package/cordova-plugin-touch-id">https://www.npmjs.com/package/cordova-plugin-touch-id</a>
Bluetooth low energy	cordova-plugin-bluetoothle	4.0.0	<a href="https://www.npmjs.com/package/cordova-plugin-bluetoothle">https://www.npmjs.com/package/cordova-plugin-bluetoothle</a>
TTS	cordova-plugin-tts	0.2.3	<a href="https://www.npmjs.com/package/cordova-plugin-tts">https://www.npmjs.com/package/cordova-plugin-tts</a>
Clipboard	cordova.plugins.clipboard	0.1.0	<a href="https://www.npmjs.com/package/cordova-universal-clipboard">https://www.npmjs.com/package/cordova-universal-clipboard</a>
Web Socket	cordova-plugin-websocket	0.12.0	<a href="https://www.npmjs.com/package/cordova-plugin-websocket">https://www.npmjs.com/package/cordova-plugin-websocket</a>

### 3.3.1 Connecting with **AppeonMobile.CordovaPlugin**

**AppeonMobile.CordovaPlugin** acts as a bridge for the PowerBuilder code to execute the JavaScript code. You will need to connect with **AppeonMobile.CordovaPlugin** first and then use the functions it provides to execute the JavaScript code.

To connect/disconnect with the **AppeonMobile.CordovaPlugin** object, you can use the PowerBuilder OLEObject's `ConnectToNewObject` and `DisconnectObject` functions, for example,

```
oleobject ole
ole = Create Oleobject
//connects with the AppeonMobile.CordovaPlugin object
ole.Connecttonewobject ( "AppeonMobile.CordovaPlugin" )
```

```
//disconnects with the AppeonMobile.CordovaPlugin object  
ole.DisconnectObject ()
```

### 3.3.2 AppeonMobile.CordovaPlugin object

#### 3.3.2.1 SetTimeout function

##### Description

Sets the time out value in milliseconds. By default it is 5 seconds. If 0, no timeout period is in effect.

##### Syntax

```
SetTimeout ( milliseconds )
```

##### Return value

None

##### Usage

This function is required when ExecJavaScriptWithReturn is called, because the Cordova plugin may connect with the network, or plugin may have bugs or invalid parameters, etc.

This function is required when @ (instead of @EventName) is used to wait for the execution result.

#### 3.3.2.2 AssociateJSwithPB function

##### Description

Associates a Cordova plugin (JavaScript object) with a PowerBuilder object. After associated, the PowerBuilder object will have all of the methods and properties that the JavaScript object has.

##### Syntax

```
AssociateJSwithPB ( string, PBobject )
```

##### Parameter

*string* - The JavaScript object defined by the Cordova plugin. If it is not a global object, then it will need to be created first before it can be associated with.

*PBobject* - The PowerBuilder object. If callback functions are used in the JavaScript object, then the corresponding callback events should be defined in this object.

##### Return value

None

##### Code example

```
oleobject ole  
Ole = Create Oleobject  
//Connects with AppeonMobile.CordovaPlugin  
ole.Connecttonewobject ( "AppeonMobile.CordovaPlugin" )  
//Associates with the plugin object  
//navigator.contacts is a global object according to the Cordova help
```

```
ole.AssociateJSwithPB ( "navigator.contacts", this )
```

### 3.3.2.3 ExecJavaScript function

#### Description

Executes a line of JavaScript code and returns no value. This function can be used to execute a line of JavaScript code that returns no value (asynchronous execution).

To execute a line of JavaScript code that returns a value, call the [ExecJavaScriptWithReturn function](#).

To execute a line of JavaScript code that contains a callback function, it is not recommended to use ExecJavaScript, as it will be very complicated to write scripts to use the callback function in ExecJavaScript; instead, it is recommended that the user associates with the JavaScript object where the function will be called, and then use dot notation to execute this function. Dot notation will be described more in [Using dot notation](#).

#### Syntax

```
ExecJavaScript ( string )
```

#### Return value

None

#### Usage

This function must be used to execute the JavaScript code that creates an object. If the JavaScript code contains a string, then the string must be included in double quotation marks. The semi colon (;) at the end of the JavaScript code can be preserved or omitted.

#### Example 1:

JavaScript code:

```
var myContact = navigator.contacts.create ( { "displayName": "Test User" } );
```

Corresponding PowerBuilder code:

```
oleobject.ExecJavaScript ( 'var mycontact = navigator.contacts.create  
({ "displayname": "TestUser" } );' )
```

#### Example 2:

JavaScript code:

```
var options = new ContactFindOptions () ;
```

Corresponding PowerBuilder code:

```
lole_cordova.ExecJavaScript ( "var options = new ContactFindOptions() ;" )
```

#### Code example

```
Oleobject lole_cordova  
lode_cordova = Create OleObject  
//connects with AppeonMobile.CordovaPlugin  
lode_cordova.connecttonewobject ( "AppeonMobile.CordovaPlugin" )  
//associates this PB object with a JS object navigator.contacts  
lode_cordova.AssociateJSwithPB ( "navigator.contacts", this )  
//sets the timeout value
```

```
lole_cordova.setTimeout (10000)
//executes a line of JS code
//creates a new ContactFindOptions object
lole_cordova.ExecJavaScript ( "var options = new ContactFindOptions();" )
//associates with the new object
lole_cordova.AssociateJSwithPB ( "options", This )
//as the value of filter is a string, include "Test" in quotation marks
lole_cordova.ExecJavaScript ( "options.filter=~\"Test~\"" )
//as the multiple object is a boolean, no need to include it in quotation marks
lole_cordova.ExecJavaScript ( "options.multiple=true" )
```

### 3.3.2.4 ExecJavaScriptWithReturn function

#### Description

Executes a line of JavaScript code and returns the result.

#### Syntax

```
ExecJavaScriptWithReturn ( string )
```

#### Return value

String.

Returns a string if successful. If an object is returned, the object properties will be described in a JSON-format string.

Returns an empty string if timeout is reached (possible causes for timeout: script is too long, script has errors, parameter is incorrect, etc.).

#### Code example

If the JavaScript code contains a string, then the string must be included in double quotation marks. The semi colon (;) at the end of the JavaScript code can be preserved or omitted.

```
String ls_Return
Oleobject lole_cordova
lode_cordova = Create OleObject
//connects the PB OLEObject with AppeonMobile.cordovaPlugin
lode_cordova.connecttowebobject ("AppeonMobile.CordovaPlugin")
//associates this PB object with a JS object navigator.contacts
lode_cordova.AssociateJSwithPB ( "navigator.contacts", this)
//sets the timeout value
lode_cordova.setTimeout (10000)
//executes a line of JS code and returns no value
lode_cordova.ExecJavaScript ( "var options = new ContactFindOptions();" )
//executes a line of JS code and returns a value. The value is a string in the JSON
//format containing all of the properties of options object
ls_Return = lode_cordova.ExecJavaScriptWithReturn ( "options" )
```

### 3.3.2.5 ExecJavaScriptGetLastFuncCallRetVal function

#### Description

Gets the return value of the last method that uses callback functions. Executing this function will get the return value of the last executed JavaScript method that uses callback functions.

#### Syntax

```
ExecJavaScriptGetLastFuncCallRetVal ()
```

#### Return value

String.

### Code example

```
oleobject ole_orientation
ole_orientation = Create oleobject
Integer li_Return
String ls_Return
li_Return = ole_orientation.connecttonewobject ("AppeonMobile.CordovaPlugin")
ole_orientation.AssociateJSwithPB ("navigator.compass", this)
//ls_Return stores the return value of the callback event
ls_Return = ole_orientation.watchHeading ('@', '@', '{frequency: 3000}')
//gets the return value of watchHeading()
ls_Return = ole_orientation.ExecJavaScriptGetLastFuncCallRetVal()
ole_orientation.watchHeading ('@ue_contactfind', '@ue_contactfind_error',
    '{frequency: 50}')
//gets the return value of watchHeading()
ls_Return = ole_orientation.ExecJavaScriptGetLastFuncCallRetVal()
ole_orientation.disconnectobject()
```

#### 3.3.2.6 GetLastErrorExecJS function

##### Description

Gets the error information if the JavaScript code has an error, including incorrect function name, misspellings, execution errors etc. The error information is returned from the JavaScript code. You can use GetLastErrorExecJS together with [oe\\_execjserror](#) (string *errormessage*) to help determine the cause.

##### Syntax

```
GetLastErrorExecJS ()
```

##### Return value

String.

The following error types can be detected by GetLastErrorExecJS:

- Missing "@" when using the callback event
- Misspelling of the JavaScript function name or object property name in a dot notation
- Misspelling of the PowerServer-provided function name
- Incorrect JavaScript object name or function name such as case mismatch
- Format error of the JSON string, such as missing open bracket ("[")
- Incorrect parameter or invalid property value in a dot notation of calling JavaScript code

The following error types **cannot** be detected by GetLastErrorExecJS:

- Missing "@" if you do not want to use the callback event
- Misspelling of the JavaScript object property name
- Incorrect keyword name in a JSON string

### 3.3.2.7 oe\_execjserror event

#### Description

Occurs when the JavaScript code has an error, including incorrect function name, misspellings, and execution errors etc. It is an asynchronous event.

#### Syntax

```
oe_execjserror ( string errormessage )
```

#### Usage

To use this event, you need to add an event with the same name in the PowerBuilder object that is bound with the JavaScript object via `AssociateJSwithPB ( string, PBobject )`.

## 3.3.3 Accessing the JavaScript object properties/methods

### 3.3.3.1 Using dot notation

After associating a PowerBuilder object with a JavaScript object, you can access the JavaScript object properties and methods using dot notation (.) .

Regarding the return value, only when the method uses a callback function, the method's return value will be returned as a string. If the method uses no callback function, no value will be returned; if you want to get the return value, you can execute the JavaScript code.

#### To access a JavaScript method:

JavaScript code:

```
navigator.contacts.find (fields, onSuccess, onError);
```

The corresponding PowerBuilder code:

```
//creates and connects the PB OLEObject with AppeonMobile.cordovaPlugin. Scripts  
//are omitted here.  
//associates this PB object with a JS object  
oleobject.AssociateJSwithPB ("navigator.contacts", this)  
//executes the find method. The method parameter will be described in detail later.  
//ls_Return is a JSON-format string. The string contains the information of the  
//found contacts including name, phone number, address etc.  
ls_Return = oleobject.find (fields, "@", "@")
```

#### To access a JavaScript property:

JavaScript code:

```
//create the object  
var optionsreturn = new ContactFindOptions();  
//sets the property value  
optionsreturn.filter = "Bob"; //property value is a string  
//obtains the property value  
var filtervalue = optionsreturn.filter;
```

The corresponding PowerBuilder code:

```
String filtervalue  
//creates the object  
oleobject.ExecJavaScript ("var optionsreturn = new ContactFindOptions();")
```

```
//associates the object
oleobject.AssociateJSwithPB ("optionsreturn", this)
//accesses the property
//sets the property value
oleobject.filter ="~"Bob~" //string data type, included in quotation marks
//obtains the property value
filtervalue = oleobject.filter
```

### 3.3.3.2 Using a JSON string

You can use a [JSON](#) string as the parameter to set the object property. You will first need to know the JSON format and the data type of the property values. For example,

Suppose there is an "options" object parameter in the JavaScript method:

```
navigator.contacts.find (fields, onSuccess, onError, options);
```

You can set the value for the "options" object parameter using a JSON string in PowerBuilder code like this:

```
ls_Return = oleobject.find (fields, "@", "@", '{"filter": "Bob"}')
```

Here is another example:

JavaScript code:

```
var options = new ContactFindOptions();
options.filter = "Bob";
```

The corresponding PowerBuilder code (sets the object property using a JSON string when creating the object):

```
oleobject.ExecJavaScript('var options = new ContactFindOptions({"filter": "Bob"});')
```

### 3.3.3.3 Using ExecJavaScript/ExecJavaScriptWithReturn function

You can call the `ExecJavaScript`/`ExecJavaScriptWithReturn` function of **AppeonMobile.CordovaPlugin** object to execute the JavaScript code that sets the object properties, for example,

```
//creates the object
ls_Return = oleobject.ExecJavaScript ("var optionsreturn = new
ContactFindOptions();")
//sets the value of the object property
ls_Return = oleobject.ExecJavaScriptWithReturn ("optionsreturn.filter=~"test1~"")
//obtains the property value and stores in a variable
oleobject.ExecJavaScript("var testfilter = optionsreturn.filter")
//gets the variable value and returns the value
ls_Return = oleobject.ExecJavaScriptWithReturn ("testfilter")
```

### 3.3.3.4 Using `eon_cjsonnode/eon_cjsonnodearray` object

[`eon\_cjsonnode`](#) and [`eon\_cjsonnodearray`](#) objects are provided in the Appeon Workarounds PBL for handling the JSON-format strings. These two objects can be used to access the JS object properties if the object property structure are complex, as it may be very difficult to get the value from the returned string which is long and complex. You can determine which object to use according to the brace/bracket used in the string:

- values in braces ({ }), node type, use `eon_cjsonnode`

- values in brackets ([]), array type, use **eon\_cjsonnodearray**

Here is an example that shows how to obtain all of the phone numbers from the following JSON string:

```
JSONstring = {"displayName": "test1", "phoneNumbers": [{"type": "work", "value": "212-555-1234", "pref": "false"}, {"type": "mobile", "value": "917-555-5432", "true": "false"}, {"type": "home", "value": "203-555-7890", "pref": "false"}]}
```

Use **eon\_cjsonnode**, as the string is in a node (in {}):

```
Integer li_count //store the total amount of phone numbers
Eon_cjsonnode returnnode //store the returned string
returnnode = create Eon_cjsonnode // create the JSON node object
//formatize the string in the JSON object
returnnode.of_load (JSONstring)
//phoneNumbers is an array, so use Eon_cjsonnodearray to receive values
//You can also use of_gettypebykey to determine the node type.
Eon_cjsonnodearray phoneNumberarray //phoneNumbers is the node value, containing
multiple phone numbers
phoneNumberarray = create Eon_cjsonnodearray
phoneNumberarray = returnnode.of_valuearray ("phoneNumbers")
li_count = phoneNumberarray.of_size()
Eon_cjsonnode phoneNumber //store the returned string
phoneNumber = create Eon_cjsonnode
For li_for = 1 to li_Count
  phoneNumber = phoneNumberarray.Of_getnodebyindex (li_For)
  messagebox ('phonenumber', phoneNumber.of_valuestring (value)) //export the
obtained phone numbers
Next
```

### 3.3.3.5 Code example

Here is a step-by-step guide with code examples that shows how to use different ways to set a JavaScript property.

Step 1: Create the PowerBuilder **OLEObject** object, connect with the **AppeonMobile.CordovaPlugin** object, and associate the PowerBuilder object with the JavaScript object (**navigator.contacts**).

```
Oleobject lole_cordova
lolle_cordova = Create OleObject
li_Return = lolle_cordova.connecttonewobject ("AppeonMobile.CordovaPlugin")
lolle_cordova.AssociateJSwithPB ("navigator.contacts", this)
```

Step 2: Add the properties and values for the **options** object.

#### Using a JSON string:

- Create the **iec\_option** object from the **eon\_cjsonnode** object.

```
eon_cjsonnode iec_option
iec_option = Create eon_cjsonnode
```

- Call the **iec\_option**'s function to add the property and value for the **options** object.

```
iec_option.of_addkeyvalue ("filter", "test")
iec_option.of_addkeyvalue ("multiple", true)
iec_option.of_addkeyvalue ("hasPhoneNumber", false)
```

- Call the **iec\_option**'s function to return the JSON string.

```
ls_Return = iole_contact.find ([ "name" ], "@", "@", iec_option.of_ToString() )
```

### Using dot notation:

1. Create the **options** object.

```
lole_cordova.ExecJavaScript ("var options = new ContactFindOptions();")
ls_Return = lole_cordova.ExecJavaScriptWithReturn ("options")
```

2. Associate the PowerBuilder object with the **options** object.

```
lole_cordova.AssociateJSwithPB ("options", This)
```

3. Set the values of the filter, multiple and hasPhoneNumber properties for the **options** object.

```
//string must be in double quotes
lole_cordova.filter = "~"test~" //'"test"' is also valid
lole_cordova.multiple = true
lole_cordova.hasPhoneNumber = false
ls_Return = lole_cordova.ExecJavaScriptWithReturn ("options")
```

4. Associate the PowerBuilder object with the **navigator.contacts** object again.

```
lole_cordova.AssociateJSwithPB ("navigator.contacts", this)
ls_Return = lole_cordova.find (sle_1.Text, "@", "@", "options" )
```

### Using ExecJavaScript/ExecJavaScriptWithReturn function:

1. Create the **options** object.

```
lole_cordova.ExecJavaScript ("var options = new ContactFindOptions();")
ls_Return = lole_cordova.ExecJavaScriptWithReturn ("options")
```

2. Associate the PowerBuilder object with the **options** object.

```
lole_cordova.AssociateJSwithPB ("options", This)
```

3. Set the values of the filter, multiple and hasPhoneNumber properties for the **options** object.

```
lole_cordova.ExecJavaScript ("options.filter=~"tes~t~")
lole_cordova.ExecJavaScript ("options.multiple=true")
lole_cordova.ExecJavaScript ("options.hasPhoneNumber=true")
ls_Return = lole_cordova.ExecJavaScriptWithReturn ("options")
```

4. Associate the PowerBuilder object with the **navigator.contacts** object again.

```
lole_cordova.AssociateJSwithPB ("navigator.contacts", this)
ls_Return = lole_cordova.find (sle_1.Text, "@", "@", "options" )
```

### 3.3.4 Accessing the JavaScript array

Text strings must be included in double quotation marks, others (object names, boolean values, numeric values etc.) need not. This also complies with the JSON format.

Here is an example that uses the `contacts.create` method to add phoneNumbers. According to the Cordova help, `phoneNumbers` is the property of contact and its data type is an array `ContactField[]` with three properties: `type`, `value`, & `pref`.

JavaScript code:

```
//creates a new contact
var contact = navigator.contacts.create();
//stores the contact phone numbers in ContactField[]
var phoneNumbers = [];
phoneNumbers[0] = new ContactField ('work', '212-555-1234', false);
phoneNumbers[1] = new ContactField ('mobile', '917-555-5432', true); // preferred
number
phoneNumbers[2] = new ContactField ('home', '203-555-7890', false);
contact.phoneNumbers = phoneNumbers;
// save the contact
contact.save();
```

The corresponding PowerBuilder code (using dot notation):

```
oleobject.ExecJavaScript ( "var contact = navigator.contacts.create%;" )
oleobject.AssociateJSwithPB ( 'contact', This )
//creates the ContactField object
oleobject.ExecJavaScript ( "var phoneNumber1 = new ContactField%;" )
//associates with the object
oleobject.AssociateJSwithPB ( "phoneNumber1", this );
//sets the property values
oleobject.value = '"212-555-1234"'
oleobject.type = '"work"'
oleobject.pref = 'false'

//creates the ContactField object
oleobject.ExecJavaScript ( "var phoneNumber2 = new ContactField%;" )
//associates with the object
oleobject.AssociateJSwithPB ( "phoneNumber2", this );
//sets the property values
oleobject.value = '"917-555-5432"'
oleobject.type = '"mobile"'
oleobject.pref = 'true'

//creates the ContactField object
oleobject.ExecJavaScript ( "var phoneNumber3 = new ContactField%;" )
//associates with the object
oleobject.AssociateJSwithPB ( "phoneNumber3", this );
//sets the property values
oleobject.type = '"home"'
oleobject.value = '"203-555-7890"'
oleobject.pref = 'false'

//adds phoneNumbers to the newly created contact object
oleobject.AssociateJSwithPB ( "contact", this ) //associates with the contact
oleobject.phoneNumbers = '[phoneNumber1, phoneNumber2, phoneNumber3]'
oleobject.save ( "@", "@" )
```

The corresponding PowerBuilder code (using JSON):

```
oleobject.ExecJavaScript ( 'var contact = navigator.contacts.create
({ "phoneNumbers": [{"type": "work", "value": "212-555-1234", "pref": "false"}, {"type": "mobile", "value": "917-555-5432", "true": "false"}, {"type": "home", "value": "203-555-7890", "pref": "false"}]});' )
```

### 3.3.5 Using callback functions

Callback function is very common in JavaScript; however there is no callback function in PowerScript. And in order to support the JavaScript callback function, a corresponding PowerScript event must be created and be called back instead (let's call it callback event). Which means, for each JavaScript callback function, define an event in the associated

PowerBuilder object, and use **@EventName** to replace the JavaScript callback function name. If you do not want to use the callback event in PowerBuilder, use **@**, then the function will return the result directly when the execution is complete. The return value is a string (if an object is to be returned, then the object properties will be returned in a JSON string).

- **EventName** is the name of the event of the object that the second parameter of the `AssociateJSwithPB` function points to.
- The event will need to have a parameter for receiving the value returned by the function.
- If the function uses more than one callback event, either all of them are used, or none; it cannot be some used, while others not, because the event to be called back will be in asynchronous execution. For example, the following script is incorrect:

```
iOLE_contact.find ( '["name"]', "@ue_contactfind", "@",
    iec_option.of_ToString() )
```

- When the callback event failed to execute, an error code is returned. The error code is defined in the Cordova help, and you will need to search the error code in the Cordova help to determine the cause.
- As the callback event is executed asynchronously, do not execute `OLEObject.DisconnectObject()` right after **@+EventName** is used, otherwise the callback event may fail.

Here is a code example.

The JavaScript callback function `onSuccess`:

```
function onSuccess(contacts) {
    for (var i = 0; i < contacts.length; i++) {
        for (var j = 0; j < contacts[i].addresses.length; j++) {
            alert("Pref: " + contacts[i].addresses[j].pref + "\n"
+
                "Type: " + contacts[i].addresses[j].type + "\n"
+
                "Formatted: " + contacts[i].addresses[j].formatted + "\n"
+
                "Street Address: " + contacts[i].addresses[j].streetAddress + "\n"
+
                "Locality: " + contacts[i].addresses[j].locality + "\n"
                "Region: " + contacts[i].addresses[j].region + "\n"
+
                "Postal Code: " + contacts[i].addresses[j].postalCode + "\n"
+
                "Country: " + contacts[i].addresses[j].country);
        }
    }
};
```

The JavaScript callback function `onError`:

```
function onError(contactError) {
    alert('onError!');
};
```

The JavaScript method that uses these two callback functions:

```
//finds all contacts
```

```
var options = new ContactFindOptions();
options.filter = "";
options.multiple = true;
var filter = ["displayName", "addresses"];
navigator.contacts.find (filter, onSuccess, onError, options);
```

The PowerBuilder callback event `ue_contactfind` (the counterpart of `onSuccess`) defined in the associated PowerBuilder object:

```
ue_contactfind (string as_return) return none
```

The PowerBuilder callback event `ue_contactfind_error` (the counterpart of `onError`) defined in the associated PowerBuilder object:

```
ue_contactfind_error (string as_return) return none
```

The PowerBuilder code that uses these two callback events:

```
Oleobject lole_cordova
lode_cordova = Create OleObject
li_Return = lode_cordova.ConnectToNewObject ( "AppeonMobile.CordovaPlugin" )
//associates with the current PowerBuilder object
lode_cordova.AssociateJSwithPB ("navigator.contacts", this)
//ue_contactfind & ue_contactfind_error defined in "this" object (the currently
associated PB object)
//iec_option.of_ToString() is a JSON string
ls_Return = lode_contact.find ('[{"name"]', "@ue_contactfind",
"@ue_contactfind_error", iec_option.of_ToString() )
//or replace it with the following script, if not to use any callback event
//ls_Return = lode_contact.find ('[{"name"]', "@", "@", iec_option.of_ToString() )
```

### 3.3.6 Code examples

Let's take Contact Cordova plugin as an example to show you how to add a contact to the mobile device and how to write the PowerBuilder code.

The corresponding Cordova help can be found here: <https://www.npmjs.com/package/cordova-plugin-contacts>.

**Example 1:** To add the following contact to the device:

name: Tester

phone number: 15845562007, 07552698563

Step 1: Create a PowerBuilder **OLEObject** object and call the `ConnectToNewObject` function to connect it to the **AppeonMobile.CordovaPlugin** object.

```
Oleobject ole_cordova
ole_cordova = Create OLEObject
ole_cordova.ConnectToNewObject ( "AppeonMobile.CordovaPlugin" )
```

Step 2: Call the **AppeonMobile.CordovaPlugin**'s `AssociateJSwithPB` function to associate the JavaScript object with the current PowerBuilder object.

**navigator.contacts** is the name of the JavaScript object defined in the Contact Cordova plugin. You can get this name from the Cordova help.

**this** represents the object where the callback function used by **navigator.contacts** should be transferred to.

```
ole_cordova.AssociateJSwithPB ( "navigator.contacts", this)
```

### Step 3: Create a new Contact object -- **mycontact**.

Cordova help has this: "The navigator.contacts.create method is synchronous, and returns a new Contact object."

```
ole_cordova.ExecJavaScript ('var mycontact = navigator.contacts.create();')
```

Step 4: Set the name of the new contact person to "Tester".

According to the Cordova help, the **Contact** object has three properties for name: displayName, name, & nickname. We choose to use nickname to set the new contact's name. These three properties are supported differently by different platform. You can find this out in the Cordova help and determine which property to use according to your request.

```
ole_cordova.ExecJavaScript ('mycontact.nickname = "Tester";')
```

Step 5: Set the phone number of the new contact person to "15845562007" & "07552698563".

According to the Cordova help, the **Contact** object has **ContactField[]** (an array) for phoneNumbers.

Create the **ContactField** object first. According to Cordova help, the **ContactField** object has three properties: type, value, & pref. Then set the phone number to 15845562007:

```
ole_cordova.ExecJavaScript ("var p1 = new ContactField();") //creates the object
ole_cordova.ExecJavaScript ("p1.type = 'home';") //value is a string, so include it
in quotes
ole_cordova.ExecJavaScript ("p1.value = '15845562007';")
ole_cordova.ExecJavaScript ("p1.pref = true;")
```

And set the phone number to 07552698563:

```
ole_cordova.ExecJavaScript ("var p2 = new ContactField();")
ole_cordova.ExecJavaScript ("p2.type = 'work';")
ole_cordova.ExecJavaScript ("p2.value = '07552698563';")
ole_cordova.ExecJavaScript ("p2.pref = false;")
```

Finally, set the values in an array for the phoneNumbers property:

```
ole_cordova.ExecJavaScript ("mycontact.phoneNumbers = [p1,p2];")
```

Step 6: Save the new **Contact** object and the values.

According to the Cordova help, after the new **Contact** object is created, it must be saved using the **save()** method in order for the values to be saved into the database.

The **save()** method uses callback functions, so it cannot be executed using **ExecJavaScript**. We need to associate with the **mycontact** object to execute the **save()** method.

```
ole_cordova.AssociateJSwithPB ("mycontact", this)

String ls_Return //for receiving the returned value from save().
//The new contact information that is successfully saved will be returned in a JSON
string.
ls_Return = ole_cordova.save ("@","@") //we choose not to use callback function
```

**Example 2:** to write the PowerBuilder code according to the JavaScript code.

The following table shows you how to write the PowerBuilder code that will execute and get the same result as the JavaScript code.

The JavaScript code uses **ContactField** to set phoneNumbers, and it is copied from the Cordova help: <https://www.npmjs.com/package/cordova-plugin-contacts>.

**Table 3.2: JavaScript-to-PowerScript mapping**

JavaScript	PowerScript
None	<p>The following preparation is required in the PowerBuilder only.</p> <p>Create an <b>OLEObject</b> object, connect it to <b>AppeonMobile.CordovaPlugin</b>, and associate the PB object with the JS object.</p> <pre>Oleobject ole_cordova ole_cordova = Create OLEObject ole_cordova.ConnectToNewObject   ("AppeonMobile.CordovaPlugin") ole_cordova.AssociateJSwithPB   ("navigator.contacts", this)</pre>
Create a new contact:	<p>Use <b>ExecJavaScript</b> to directly execute the JavaScript code:</p> <pre>ole_cordova.ExecJavaScript ('var contact =   navigator.contacts.create();')</pre>
Store contact phone numbers in <b>ContactField[]</b> :	<p>Use <b>ExecJavaScript</b> to directly execute the JavaScript code:</p> <pre>ole_cordova.ExecJavaScript ('var phoneNumbers =   []; phoneNumbers[0] = new ContactField ('work', '212-555-1234', false); phoneNumbers[1] = new ContactField ('mobile', '917-555-5432', true); phoneNumbers[2] = new ContactField ('home', '203-555-7890', false); contact.phoneNumbers =   phoneNumbers;</pre> <pre>ole_cordova.ExecJavaScript ('var phoneNumbers =   []); ole_cordova.ExecJavaScript ('phoneNumbers[0]   = new ContactField ("work", "212-555-1234",   false);') ole_cordova.ExecJavaScript ('phoneNumbers[1]   = new ContactField ("mobile", "917-555-5432",   true);') ole_cordova.ExecJavaScript ('phoneNumbers[2]   = new ContactField ("home", "203-555-7890",   false);') ole_cordova.ExecJavaScript ('contact.phoneNumbers   = phoneNumbers;')</pre>
Save the contact:	<p>Associate the JavaScript <b>Contact</b> object with the PB <b>OLEObject</b> object, and then use dot notation to execute <b>save()</b>.</p> <p><b>save()</b> uses callback functions, so it cannot be executed by <b>ExecJavaScript</b>.</p> <pre>ole_cordova.AssociateJSwithPB ("contact", this) String ls_Return //for receiving the save()   returned value //the new contact information will be returned in a JSON string ls_Return  = ole_cordova.save ("@", "@")</pre>

# 4 PowerServer open interfaces

## 4.1 PowerServer open interfaces

### 4.1.1 Overview

PowerServer open interfaces give users the opportunity to manage the client, user sessions and transactions for their PowerServer-deployed applications. PowerServer open interfaces are encapsulated as the server-side component and Web service in the PowerServer, therefore, you can call them in the following different ways:

- Method 1: Call the interface through the non-autoinstantiated NVO provided by Appeon Workarounds PBL. **This method allows you to call the interface in the PowerServer-deployed application only.**
  - To call the open interfaces in PowerServer installed to the .NET IIS server, you will need to call the interface through the .NET component ([AppeonDotNetComponent](#)).
  - To call the open interfaces in PowerServer installed to the Java application server (such as JBoss, WebLogic, WebSphere, or JEUS), you will need to call the interface through the EJB component ([EJBOBJECT](#)).

To use the .NET or EJB component provided by PowerServer and make the code effective, you will need to

1. add the Appeon Workarounds PBLs to your PowerBuilder application; and then
  2. deploy your PowerBuilder application to PowerServer the same way you would deploy a normal PowerBuilder application.
- Method 2: Call the open interface Web service provided by PowerServer. **This method allows you to call the interface from applications written in PowerScript, Java, or C# (but not PowerServer-deployed application).**

The Web service provided by PowerServer is a standard Web service, therefore, you can call the interface in the PowerBuilder application, Java application, C# application etc. Authentication is required when calling this Web service; the user name and password is the same as that used for accessing the AEM.

Only the following interfaces (not all of the PowerServer open interfaces) are provided as Web service: getAllClients, getAllSessions, getSessionbyID, getSessionCount, and getServerVersion. The interface name is case-sensitive when called by the other programming language rather than PowerScript.

To call and make the interface effective in the PowerBuilder application, you will need to generate the client proxy object via the .NET Web service engine.

- URL for the open interface Web service in PowerServer installed to the Java application server (such as JBoss, WebLogic, WebSphere, or JEUS):  
`http://AppeonServerName:Port/servlet/AppeonService?wsdl`

- URL for the open interface Web service in PowerServer installed to the .NET IIS server:  
`http://AppeonServerName/servlet/AppeonService.asmx`

## 4.1.2 getAllClients

### Description

`getAllClients` gets the IP addresses of all client machines which corresponds to the active sessions that are opened for the specified application in the specified PowerServer.

#### 4.1.2.1 Syntax for .NET component

```
proxyobject.of_execinterface ( "getAllClients", ref any paralist[] )
```

**Table 4.1:**

Argument	Description
<code>proxyobject</code>	The instance of the proxy object generated for <code>AppeonDotNetComponent</code> .
<code>paralist</code>	<p>Arrays of Any type. Specifies the parameter arrays for the component function.</p> <p><code>paralist[1]</code> - The name of the PowerServer in which the active sessions are created. The value should be IP address or IP address with port number, for example, 192.0.0.123, or 192.0.0.123:8080, and separated with semicolons (;) if there are multiple servers (for example in a PowerServer cluster). If <code>paralist[1]</code> is NULL or an empty string (""), <b>getAllClients</b> returns the IP address of all clients for the specified application in all servers in a PowerServer cluster (you will need to configure the cluster in AEM first).</p> <p><code>paralist[2]</code> - The name of the application that is deployed to the specified PowerServer. If <code>paralist[2]</code> is NULL or empty string (""), <b>getAllClients</b> returns the IP address of clients for all applications in the specified PowerServer.</p> <p><code>paralist[3]</code> - The return value of <b>getAllClients</b> function: the IP address of all clients. This parameter must be declared, but need not to be assigned with values, as it is used to hold the return value.</p>

### Return value of `of_execinterface`

Returns 1 if the `of_execinterface` function succeeds and one of the following negative values if an error occurs.

-1: function fails to execute

-2: the specified PowerServer is not found

-3: the specified application is not found

### Return value of `getAllClients`

Returns the IP address if the **getAllClients** function succeeds and returns an empty string if an error occurs.

The return value is stored in the `paralist[3]` parameter of the `of_execinterface` function.

### Code example

```
Any la_1[]
```

```

Long lRet
...
la_1[1] = ls_servername
la_1[2] = ls_appname
la_1[3] = ls_clientcontent

lRet = loadDBList.of_ExecInterface ("getAllClients", la_1) //result is returned to
la_1[3]

```

#### 4.1.2.2 Syntax for EJB component

```

io_ejb.regstring ( String serverName )
io_ejb.regstring ( String appName )
io_ejb.Invokeretstring ( long objid, "getAllClients", true, ref string retval )

```

**Table 4.2:**

Argument	Description
<i>io_ejb</i>	A reference of Appeon EJBobject.
<i>serverName</i>	The name of the PowerServer in which the active sessions are created. The value should be IP address or IP address with port number, for example, 192.0.0.123, or 192.0.0.123:8080, and separated with semicolons (;) if there are multiple servers (for example in a PowerServer cluster). If <i>serverName</i> is NULL or an empty string (""), <b>getAllClients</b> returns the IP address of all clients for the specified application in all servers in a PowerServer cluster (you will need to configure the cluster in AEM first).
<i>appName</i>	The name of the application that is deployed to the specified PowerServer. If <i>appName</i> is NULL or empty string (""), <b>getAllClients</b> returns the IP address of clients for all applications in the specified PowerServer.
<i>objid</i>	The handle to the component method.
<i>retval</i>	The return value of <b>getAllClients</b> function: the IP address of all clients. This parameter must be declared, but need not to be assigned with values, as it is used to hold the return value.

#### Return value of Invokeretstring

Returns 1 if the **Invokeretstring** function succeeds and one of the following negative values if an error occurs.

- 1: function fails to execute
- 2: the specified PowerServer is not found
- 3: the specified application is not found

#### Return value of getAllClients

Returns the IP address if the **getAllClients** function succeeds and returns an empty string if an error occurs.

The return value is stored in the *retval* parameter of **Invokeretstring** function.

#### Code example

Note that **invokeretstring** function is called because the return value is a string.

```

string retval01
string ls_msg
...
io_ejb.regstring (ls_servername)
io_ejb.regstring (ls_appname)

ls_msg = io_ejb.invokeretstring (il_beanc1, "getAllClients", true, ref retval01) // result is returned to retval01

```

#### 4.1.2.3 Syntax for Web Service

`getAllClients ( string serverName, string appName )`

**Table 4.3:**

Argument	Description
<i>serverName</i>	The name of the PowerServer in which the active sessions are created. The value should be IP address or IP address with port number, for example, 192.0.0.123, or 192.0.0.123:8080, and separated with semicolons (;) if there are multiple servers (for example in a PowerServer cluster). If <i>serverName</i> is NULL or an empty string (""), <b>getAllClients</b> returns the IP address of all clients for the specified application in all servers in a PowerServer cluster (you will need to configure the cluster in AEM first).
<i>appName</i>	The name of the application that is deployed to the specified PowerServer. If <i>appName</i> is NULL or empty string (""), <b>getAllClients</b> returns the IP address of clients for all applications in the specified PowerServer.

#### Return value

Returns the IP address if the function succeeds and returns an empty string if an error occurs.

#### Code example

```

string ls_return,ls_server,ls_app
ieon_soap.createinstance (ieon_ws, 'wsappeonservice')
ieon_head.username = 'admin'
ieon_head.password = 'admin'
ieon_ws.setauthsoapheadervalue (ieon_head)
ls_return = ieon_ws.getAllClients (ls_server, ls_app)

```

#### 4.1.3 getAllSessions

##### Description

`getAllSessions` returns the detail information of active sessions with XML format, which are opened for the specific application in the specific PowerServer.

##### Syntax

###### 4.1.3.1 Syntax for .NET component

`proxyobject.of_execinterface ( "getAllSessions", ref any paralist[] )`

**Table 4.4:**

Argument	Description
<i>proxyobj</i>	The instance of the proxy object generated for <code>AppeonDotNetComponent</code> .

Argument	Description
<i>paralist</i> [ <i>Arrays of Any type</i> . Specifies the parameter arrays for the component function.	<p><i>paralist[1]</i> - The name of the PowerServer that the sessions are created in. The value should be IP address or IP address with port number, for example, 192.0.0.123, or 192.0.0.123:8080, and separated with semicolons (;) if there are multiple servers (for example in a PowerServer cluster). If <i>paralist[1]</i> is NULL or an empty string (""), <b>getAllSessions</b> returns the information of all active sessions in a PowerServer cluster (you will need to configure the cluster in AEM first).</p> <p><i>paralist[2]</i> - The name of the application that is deployed to the PowerServer. If <i>paralist[2]</i> is NULL or an empty string (""), <b>getAllSessions</b> returns the information of active sessions opened for all applications in the specified PowerServer.</p> <p><i>paralist[3]</i> - The return value of <b>getAllSessions</b> function: the session content in XML format. This parameter must be cleared, but need not to be assigned with values, as it is used to hold the return value.</p>

### Return value of `of_execinterface`

Returns 1 if the `of_execinterface` function succeeds and one of the following negative values if an error occurs.

-1: function fails to execute

-2: the specified PowerServer is not found

-3: the specified application is not found

### Return value of `getAllSessions`

Returns an XML string. The XML string is stored in the *paralist[3]* parameter of the `of_execinterface` function.

The session content in the XML format will be like this.

```
<?xml version='1.0' encoding='utf-8'?>
<sessions>
  <error>
    <code>0</code>
    <text></text>
  </error>
  <sever name="172.16.8.40" port="80">
    <session name="826325974">
      <session_id>826325974</session_id>
      <client>172.16.8.40</client>
      <application-name>get_session</application-name>
      <user/>
      <state>0</state>
      <started>2018-11-20 12:56:28</started>
      <last_accessed>2018-11-20 12:57:35</last_accessed>
      <duration>67</duration>
      <server_id>172.16.8.40:80</server_id>
      <aws_id>*</aws_id>
    </session>
    <session name="241910444">
      <session_id>241910444</session_id>
      <client>172.16.100.170</client>
      <application-name>34455</application-name>
      <user/>
    </session>
  </sever>
</sessions>
```

```

<state>0</state>
<started>2018-11-20 12:57:14</started>
<last_accessed>2018-11-20 12:57:15</last_accessed>
<duration>21</duration>
<server_id>172.16.8.40:80</server_id>
<aws_id>DBEA12D26A7E4C44A8715EDC4C46D8F3</aws_id>
</session>
</server>
</sessions>

```

Method execution results are recorded in the **error** node.

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

Values for the *code* parameter:

- 0: function succeeds in execution.
- 1: function fails to execute.
- 2: the specified PowerServer is not found.
- 3: the specified application is not found.

### Code example

```

Any la_1[]
Long lRet
...
la_1[1] = ls_servername
la_1[2] = ls_appname
la_1[3] = ls_sessioncontent

lRet = loadDBList.of_ExecInterface ("getAllSessions", la_1) //result is returned to
la_1[3]

```

#### 4.1.3.2 Syntax for EJB component

For Java:

```

io_ejb.regstring ( String serverName )
io_ejb.regstring ( String appName )
io_ejb.Invokeretstring ( long objid, "getAllSessions", true, ref string retval )

```

**Table 4.5:**

Argument	Description
<i>io_ejb</i>	A reference of Appeon EJBobject.
<i>serverName</i>	The name of the PowerServer in which the active sessions are created. The value should be IP address or IP address with port number, for example, 192.0.0.123, or 192.0.0.123:8080, and separated with semicolons (;) if there are multiple servers (for example in a PowerServer cluster). If <i>serverName</i> is NULL or an empty string (""), <b>getAllSessions</b> returns the information of active sessions opened for the specified application in all servers in a PowerServer cluster (you will need to configure the cluster in AEM first).

Argument	Description
<i>appName</i>	The name of the application that is deployed to the specified PowerServer. If <i>appName</i> is NULL or empty string (""), <b>getAllSessions</b> returns the information of active sessions opened for all applications in the specified PowerServer.
<i>objid</i>	The handle to the component method.
<i>retval</i>	The return value of <b>getAllSessions</b> function: the session content in XML format. This parameter must be cleared, but need not to be assigned with values, as it is used to hold the return value.

### Return value of Invokeretstring

Returns 1 if the **Invokeretstring** function succeeds and one of the following negative values if an error occurs.

- 1: function fails to execute
- 2: the specified PowerServer is not found
- 3: the specified application is not found

### Return value of getAllSessions

Returns an XML string. The XML string is stored in the *retval* parameter of **Invokeretstring** function.

The session content in the XML format will be like this.

```
<?xml version='1.0' encoding='utf-8'?>
<sessions>
  <error>
    <code>0</code>
    <text></text>
  </error>
  <server name="172.16.8.40" port="80">
    <session name="826325974">
      <session_id>826325974</session_id>
      <client>172.16.8.40</client>
      <application-name>get_session</application-name>
      <user/>
      <state>0</state>
      <started>2018-11-20 12:56:28</started>
      <last_accessed>2018-11-20 12:57:35</last_accessed>
      <duration>67</duration>
      <server_id>172.16.8.40:80</server_id>
      <aws_id>*</aws_id>
    </session>
    <session name="241910444">
      <session_id>241910444</session_id>
      <client>172.16.100.170</client>
      <application-name>34455</application-name>
      <user/>
      <state>0</state>
      <started>2018-11-20 12:57:14</started>
      <last_accessed>2018-11-20 12:57:15</last_accessed>
      <duration>21</duration>
      <server_id>172.16.8.40:80</server_id>
      <aws_id>DBEA12D26A7E4C44A8715EDC4C46D8F3</aws_id>
    </session>
  </server>
</sessions>
```

Method execution results are recorded in the **error** node.

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

Values for the *code* parameter:

- 0: function succeeds in execution.
- 1: function fails to execute.
- 2: the specified PowerServer is not found.
- 3: the specified application is not found.

### Code example

Note that **invokeretstring** function is called because the return value is a string.

```
string ls_msg
string retval01
...
io_ejb.regstring (ls_servername)
io_ejb.regstring (ls_appname)

ls_msg = io_ejb.invokeretstring (il_bean1, "getAllSessions", true, ref retval01) // result is returned to retval01
```

#### 4.1.3.3 Syntax for Web Service

`getAllSessions ( string serverName, string appName )`

Table 4.6:

Argument	Description
<i>serverName</i>	The name of the PowerServer in which the active sessions are created. The value should be IP address or IP address with port number, for example, 192.0.0.123, or 192.0.0.123:8080, and separated with semicolons (;) if there are multiple servers (for example in a PowerServer cluster). If <i>serverName</i> is NULL or an empty string (""), <b>getAllSessions</b> returns the information of active sessions opened for the specified application in all servers in a PowerServer cluster (you will need to configure the cluster in AEM first).
<i>appName</i>	The name of the application that is deployed to the specified PowerServer. If <i>appName</i> is NULL or empty string (""), <b>getAllSessions</b> returns the information of active sessions opened for all applications in the specified PowerServer.

### Return value

XML string. For example,

```
<?xml version='1.0' encoding='utf-8'?>
<sessions>
  <error>
    <code>0</code>
    <text></text>
  </error>
  <server name="172.16.8.40" port="80">
    <session name="826325974">
      <session_id>826325974</session_id>
      <client>172.16.8.40</client>
```

```

<application-name>get_session</application-name>
<user/>
<state>0</state>
<started>2018-11-20 12:56:28</started>
<last_accessed>2018-11-20 12:57:35</last_accessed>
<duration>67</duration>
<server_id>172.16.8.40:80</server_id>
<aws_id>*</aws_id>
</session>
<session name="241910444">
  <session_id>241910444</session_id>
  <client>172.16.100.170</client>
  <application-name>34455</application-name>
  <user/>
  <state>0</state>
  <started>2018-11-20 12:57:14</started>
  <last_accessed>2018-11-20 12:57:15</last_accessed>
  <duration>21</duration>
  <server_id>172.16.8.40:80</server_id>
  <aws_id>DBEA12D26A7E4C44A8715EDC4C46D8F3</aws_id>
</session>
</sever>
</sessions>
```

Method execution results are recorded in the **error** node.

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

Values for the *code* parameter:

- 0: function succeeds in execution.
- 1: function fails to execute.
- 2: the specified PowerServer is not found.
- 3: the specified application is not found.

### Code example

```

//--connected to the web service
.....
string ls_return, ls_server, ls_app
ls_server = sle_1.text
ls_app = sle_2.text
ls_return = ieon_ws.getAllSessions (ls_server, ls_app)
```

## 4.1.4 getSessionByID

### Description

getSessionByID returns the detail information of the specified session with XML format.

#### 4.1.4.1 Syntax for .NET component

*proxyobject*.of\_execinterface ( "getSessionByID", ref any *paralist*[] )

**Table 4.7:**

Argument	Description
<i>proxyobj</i>	The instance of the proxy object generated for AppeonDotNetComponent.

Argument	Description
<i>paralist</i> [ <i>Arrays of Any type</i> . Specifies the parameter arrays for the component function.	<p><i>paralist[1]</i> - The ID of the session. Only one ID is allowed. If this parameter is NULL or an empty string (""), in .NET, the current application session will be used as the default value.</p> <p><i>paralist[2]</i> - The return value of <b>getSessionByID</b> function: the session content in XML format. This parameter must be declared, but need not to be assigned with values, as it is used to hold the return value.</p>

### Return value of `of_execinterface`

Returns 1 if the **of\_execinterface** function succeeds and one of the following negative values if an error occurs.

-1: function fails to execute

-2: the specified session ID is not found

### Return value of `getSessionByID`

Returns an XML string. The XML string is stored in the *paralist[2]* parameter of the **of\_execinterface** function.

The session content in the XML format will be like this.

```
<?xml version='1.0' encoding='utf-8'?>
<sessions>
  <error>
    <code>0</code>
    <text></text>
  </error>
  <server name="172.16.8.40" port="80">
    <session name="241910444">
      <session_id>241910444</session_id>
      <client>172.16.100.170</client>
      <application-name>34455</application-name>
      <user/>
      <state>0</state>
      <started>2018-11-20 12:57:14</started>
      <last_accessed>2018-11-20 12:57:15</last_accessed>
      <duration>21</duration>
      <server_id>172.16.8.40:80</server_id>
      <aws_id>DBEA12D26A7E4C44A8715EDC4C46D8F3</aws_id>
    </session>
  </server>
</sessions>
```

Method execution results are recorded in the **error** node.

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

Values for the *code* parameter:

0: function succeeds in execution.

-1: function fails to execute.

### Code example

```

Any la_1[]
Long lRet
...
la_1[1] = ls_sessionconid
la_1[2] = ls_sessioncontent

lRet = loadDBList.of_ExecInterface ("getSessionByID", la_1) //result is returned to
la_1[2]

```

#### 4.1.4.2 Syntax for EJB component

*io\_ejb.regstring ( String SessionconID )*

*io\_ejb.Invokeretstring ( long objid, "getSessionByID", true, ref string retval )*

**Table 4.8:**

Argument	Description
<i>io_ejb</i>	A reference of Appeon EJBobject.
<i>SessionconID</i>	The ID of the session. Only one ID is allowed. If this parameter is NULL or an empty string (""), in Java, no value will be used as the default value.
<i>objid</i>	The handle to the component method.
<i>retval</i>	The return value of <b>getSessionByID</b> function: the session content in XML format. This parameter must be declared, but need not to be assigned with values, as it is used to hold the return value.

#### Return value of Invokeretstring

Returns 1 if the **Invokeretstring** function succeeds and one of the following negative values if an error occurs.

-1: function fails to execute

-2: the specified session ID is not found

#### Return value of getSessionByID

Returns an XML string. The XML string is stored in the *retval* parameter of **Invokeretstring** function.

The session content in the XML format will be like this.

```

<?xml version='1.0' encoding='utf-8'?>
<sessions>
  <error>
    <code>0</code>
    <text></text>
  </error>
  <server name="172.16.8.40" port="80">
    <session name="241910444">
      <session_id>241910444</session_id>
      <client>172.16.100.170</client>
      <application-name>34455</application-name>
      <user/>
      <state>0</state>
      <started>2018-11-20 12:57:14</started>
      <last_accessed>2018-11-20 12:57:15</last_accessed>
      <duration>21</duration>
      <server_id>172.16.8.40:80</server_id>
    </session>
  </server>
</sessions>

```

```

<aws_id>DBEA12D26A7E4C44A8715EDC4C46D8F3</aws_id>
</session>
</sever>
</sessions>

```

Method execution results are recorded in the **error** node.

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

Values for the *code* parameter:

0: function succeeds in execution.

-1: function fails to execute.

### Code example

Note that **invokeretstring** function is called because the return value is a string.

```

string ls_msg
string retval01
...
io_ejb.regstring (ls_sessionconid)

ls_msg = io_ejb.invokeretstring (il_beanc1, "getSessionByID", true, ref retval01) // 
result is returned to retval01

```

#### 4.1.4.3 Syntax for Web Service

`getSessionByID ( string sessionid )`

**Table 4.9:**

Argument	Description
<i>sessionid</i>	The ID of the session. Only one ID is allowed. If this parameter is NULL or an empty string (""), in .NET, the current application session will be used as the default value.

### Return value

XML string. For example,

```

<?xml version='1.0' encoding='utf-8'?>
<sessions>
  <error>
    <code>0</code>
    <text></text>
  </error>
  <sever name="172.16.8.40" port="80">
    <session name="241910444">
      <session_id>241910444</session_id>
      <client>172.16.100.170</client>
      <application-name>34455</application-name>
      <user/>
      <state>0</state>
      <started>2018-11-20 12:57:14</started>
      <last_accessed>2018-11-20 12:57:15</last_accessed>
      <duration>21</duration>
      <server_id>172.16.8.40:80</server_id>
    </session>
  </sever>
</sessions>

```

```

<aws_id>DBEA12D26A7E4C44A8715EDC4C46D8F3</aws_id>
</session>
</sever>
</sessions>
```

Method execution results are recorded in the **error** node.

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

Values for the *code* parameter:

0: function succeeds in execution.

-1: function fails to execute.

### Code example

```

//--connected to the web service
.....
string ls_return, ls_session
ls_session= sle_3.text
ls_return = ieon_ws.getSessionByID (ls_session)
```

## 4.1.5 getSessionCount

### Description

With getSessionCount method, you can get the following three types of information.

- The total number of active sessions opened for the specified application in the specified PowerServer.
- The total number of active sessions in a specified PowerServer.
- The total number of sessions opened for the specified application in a PowerServer cluster.  
To get the number of sessions in a PowerServer cluster, you need to first configure the cluster in AEM.

### Usage

By using the getSessionCount method, you can easily get the total number of active sessions in a specified PowerServer using PowerBuilder code and apply the information in other open interfaces, such as KillSession, to manage the sessions. For example, you can first call getSessionCount and then call KillSession to make the deployed application kill all sessions in PowerServer when there are up to 100 active sessions in the server.

#### 4.1.5.1 Syntax for .NET component

*proxyobject*.of\_execinterface ( "getSessionCount", ref any *paralist*[] )

**Table 4.10:**

Argument	Description
<i>proxyobj</i>	The instance of the proxy object generated for AppeonDotNetComponent.
<i>paralist</i> []	Arrays of Any type. Specifies the parameter arrays for the component function.

Argument	Description
	<p><i>paralist[1]</i> - The name of the PowerServer in which the active sessions are created. The value should be IP address or IP address with port number, for example, 192.0.0.123, or 192.0.0.123:8080, and separated with semicolons (;) if there are multiple servers (for example in a PowerServer cluster). If <i>paralist[1]</i> is NULL or an empty string (""), <b>getSessionCount</b> returns the number of the active sessions in a PowerServer cluster.</p> <p><i>paralist[2]</i> - The name of the application that is deployed to the PowerServer for which you want to know how many sessions are opened. If <i>paralist[2]</i> is NULL or empty string (""), <b>getSessionCount</b> returns the total number of active sessions in the specified PowerServer.</p>

### Return value of `of_execinterface`

Returns 0 if `of_execinterface` function succeeds and one of the following negative values if an error occurs.

- 1: function fails to execute
- 2: the specified PowerServer is not found
- 3: the specified application is not found

### Return value of `getSessionCount`

Returns the number of active sessions if `getSessionCount` function succeeds and returns an empty string if an error occurs.

The return value is stored in the *ReturnValue* property of the proxy object.

### Code example

```
Any la_1[]
Long lRet
...
la_1[1] = ls_servername
la_1[2] = ls_appname

lRet = loadDBList.of_ExecInterface ("getSessionCount", la_1) //result is returned
to loadDBList.ReturnValue
```

#### 4.1.5.2 Syntax for EJB component

```
io_ejb.regstring ( String serverName )
io_ejb.regstring ( String appName )
io_ejb.Invokeretint ( long objid, "getSessionCount", true, ref integer retval )
```

Table 4.11:

Argument	Description
<i>io_ejb</i>	A reference of Appeon EJBobject.

Argument	Description
	If <i>serverName</i> is NULL or an empty string (""), <b>getSessionCount</b> returns the number of the active sessions in a PowerServer cluster.
<i>appName</i>	The name of the application that is deployed to the PowerServer for which you want to know how many sessions are opened.  If <i>appName</i> is NULL or empty string (""), <b>getSessionCount</b> returns the total number of active sessions in the specified PowerServer.
<i>objid</i>	The handle to the component method.
<i>retval</i>	The return value of <b>getSessionCount</b> function: the number of active sessions opened for the specified application in the specified PowerServer. This parameter must be declared, but need not to be assigned with values, as it is used to hold the return value.

### Return value of Invokeretint

Returns 0 if **Invokeretint** function succeeds and one of the following negative values if an error occurs.

- 1: function fails to execute
- 2: the specified PowerServer is not found
- 3: the specified application is not found

### Return value of getSessionCount

Returns the number of active sessions opened for the specified application in the specified PowerServer if **getSessionCount** function succeeds or returns an empty string if an error occurs.

The return value is stored in the *retval* parameter of **Invokeretint** function.

### Code example

Note **invokeretint** function is called because the return value is an integer.

```
string ls_msg
int retval01
...
io_ejb.regstring (ls_servername)
io_ejb.regstring (ls_appname)

ls_msg = io_ejb.invokeretint (il_beant1, "getSessionCount", true, ref retval01) // result is returned to retval01
```

#### 4.1.5.3 Syntax for Web Service

`getSessionCount ( string serverName, string appName )`

Table 4.12:

Argument	Description
<i>serverName</i>	The name of the PowerServer in which the active sessions are created. The value should be IP address or IP address with port number, for example, 192.0.0.123, or 192.0.0.123:8080, and separated with semicolons (;) if there are multiple servers

Argument	Description
	(for example in a PowerServer cluster). If <i>serverName</i> is NULL or an empty string (""), <b>getSessionCount</b> returns the number of the active sessions in a PowerServer cluster (you will need to configure the cluster in AEM first).
<i>appName</i>	The name of the application that is deployed to the PowerServer for which you want to know how many sessions are opened. If <i>appName</i> is NULL or empty string (""), <b>getSessionCount</b> returns the total number of active sessions in the specified PowerServer.

### Return value

Long.

Returns the number of sessions in the specified PowerServer if the function succeeds or returns -1 if an error occurs.

### Code example

```
// -- connected with the web service
.....
Long ll_return
String ls_server, ls_app
ls_server = sle_1.text
ls_app = sle_2.text
ll_return = ieon_ws.getSessionCount (ls_server, ls_app)
```

## 4.1.6 getServerVersion

### Description

getServerVersion gets the version number of PowerServer.

#### 4.1.6.1 Syntax for .NET component

*proxyobject*.of\_execinterface ( "getServerVersion" )

Table 4.13:

Argument	Description
<i>proxyobj</i>	The instance of the proxy object generated for AppeonDotNetComponent.

### Return value of of\_execinterface

Returns 0 if it succeeds and -1 if it fails.

### Return value of getServerVersion

String. Returns the PowerServer version number. The return value is stored in the *ReturnValue* property of the proxy object.

### Code example

```
appeondotnetcomponent ieon_dotnet
long ll_return
ieon_dotnet = create appeondotnetcomponent
ieon_dotnet.ComponentType ='4'
ieon_dotnet.TypeLib = ''
ieon_dotnet.ClassDescript = ''
```

```

ll_return = ieon_dotnet.of_execinterface ("getServerVersion") //result is returned
to ieon_dotnet.returnvalue
if ll_return = 0 then
    mle_1.text = string (ieon_dotnet.returnvalue)
else
    messagebox ("Failed", ll_return)
end if

```

#### 4.1.6.2 Syntax for EJB component

*io\_ejb.Invokeretstring ( long objid, "getServerVersion", true, ref string retval )*

**Table 4.14:**

Argument	Description
<i>io_ejb</i>	A reference of Appeon EJBobject.
<i>objid</i>	The handle to the component method.
<i>retval</i>	The return value of <b>getServerVersion</b> function: the version number of the specified PowerServer. This parameter must be declared, but need not to be assigned with values, as it is used to hold the return value.

#### Return value of Invokeretstring

Returns 0 if it succeeds and -1 if it fails.

#### Return value of getServerVersion

String. Returns the version number of PowerServer. The return value is stored in the *retval* property of the **Invokeretstring** function.

#### Code example

Note **invokeretstring** function is called because the return value is a string.

```

//Connects with EJB and returns the instance handle il_beans
string ls_return, ls_msg
ls_return = ieon_ejb.invokeretstring (il_beans, "getServerVersion", true, ref
ls_msg)
//returns an empty string to ls_return if it succeeds, and returns the PowerServer
version number to ls_msg

```

#### 4.1.6.3 Syntax for Web Service

*getServerVersion ()*

#### Return value

String. Returns the version number of PowerServer if it succeeds and returns an empty string if an error occurs.

#### Code example

```

//-- connected with the web service
.....
String ls_return
ls_return = ieon_ws.getServerVersion()

```

#### 4.1.7 killAllSessions

#### Description

`killAllSessions` kills all active sessions in a PowerServer or a PowerServer cluster and rolls back all associated transactions. To kill all sessions in a PowerServer cluster, you need to first configure the cluster in AEM.

#### 4.1.7.1 Syntax for .NET component

`proxyobject.of_execinterface ( "killAllSessions", ref any paralist[] )`

**Table 4.15:**

Argument	Description
<code>proxyobj</code>	The instance of the proxy object generated for <code>AppeonDotNetComponent</code> .
<code>paralist</code>	[Arrays of Any type. Specifies the parameter arrays for the component function.  <code>paralist[1]</code> - The name of PowerServer for which you want to kill all sessions. The value should be IP address or IP address with port number, for example, 192.0.0.123, or 192.0.0.123:8080, and separated with semicolons (;) if there are multiple servers (for example in a PowerServer cluster).  If <code>paralist[1]</code> is NULL or empty string (""), it kills all sessions and rolls back all associated transactions in a PowerServer cluster.]

#### Return value of `of_execinterface`

Returns 0 if it succeeds and -1 if it fails.

#### Return value of `killAllSessions`

Boolean.

It returns true if it succeeds and false if it fails. The return value is stored in the `ReturnValue` property of the proxy object.

#### Code example

```
Any la_1[]
Long lRet
...
la_1[1] = ls_servername

lRet = loadDBList.of_ExecInterface ( "killAllSessions", la_1) //result is returned
to loadDBList.ReturnValue
```

#### 4.1.7.2 Syntax for EJB component

`io_ejb.regstring ( String serverName )`

`io_ejb.Invokeretbool ( long objid, "killAllSessions", true, ref boolean retval )`

**Table 4.16:**

Argument	Description
<code>io_ejb</code>	A reference of <code>Appeon EJBObject</code> .
<code>serverName</code>	The name of PowerServer for which you want to kill all sessions. The value should be IP address or IP address with port number, for example, 192.0.0.123, or 192.0.0.123:8080, and separated with semicolons (;) if there are multiple servers (for example in a PowerServer cluster).

Argument	Description
	If <i>serverName</i> is NULL or empty string (""), it kills all sessions and rolls back all associated transactions in a PowerServer cluster.
<i>objid</i>	The handle to the component method.
<i>retval</i>	The return value of <b>killAllSessions</b> function: true if it succeeds and false if it fails. This parameter must be declared, but need not to be assigned with values, as it is used to hold the return value.

### Return value of Invokeretbool

Returns 0 if it succeeds and -1 if it fails.

### Return value of killAllSessions

Boolean.

It returns true if it succeeds and false if it fails. The return value is stored in the *retval* property of the **Invokeretbool** function.

### Code example

Note that **invokeretbool** function is called because the return value is a boolean.

```
string ls_msg
boolean retval01
...
io_ejb.regstring (ls_servername)

ls_msg = io_ejb.invokeretbool (il_beans1, "killAllSessions", true, ref retval01) // result is returned to retval01
```

### 4.1.7.3 Syntax for Web Service

**killAllSessions ( String *as\_server* )**

**Table 4.17:**

Argument	Description
<i>as_server</i>	The name of PowerServer for which you want to kill all sessions. The value should be IP address or IP address with port number, for example, 192.0.0.123, or 192.0.0.123:8080, and separated with semicolons (;) if there are multiple servers (for example in a PowerServer cluster).  If <i>as_server</i> is NULL or empty string (""), it kills all sessions and rolls back all associated transactions in a PowerServer cluster.

### Return value

Returns 1 if it succeeds and -1 if it fails.

### Code example

```
//-- connected with the web service
.....
string ls_return,ls_server,ls_app
ls_server = sle_1.text
ls_app = sle_2.text
ls_return = ieon_ws.killallsessions(ls_server)
```

```
mle_1.text = ls_return
```

## 4.1.8 killSessions

### Description

killSessions kills the specified session(s) in a PowerServer or a PowerServer cluster and rolls back the associated transactions. To kill sessions in a PowerServer cluster, you need to first configure the cluster in AEM. The mobile session IDs listed in AEM are not the real session IDs. You can view source to get the real mobile session IDs.

#### 4.1.8.1 Syntax for .NET component

```
proxyobject.of_execinterface ( "killSessions", ref any paralist[] )
```

**Table 4.18:**

Argument	Description
proxyobject	The instance of the proxy object generated for AppeonDotNetComponent.
paralist[ ]	Arrays of Any type. Specifies the parameter arrays for the component function. paralist[1] - The ID of sessions which you want to kill. If there are multiple sessions, separate them with semicolons (;).

### Return value of of\_execinterface

Returns 0 if it succeeds and -1 if it fails.

### Return value of killSessions

Boolean.

It returns true if it succeeds and false if it fails. The return value is stored in the *ReturnValue* property of the proxy object.

### Code example

```
Any la_1[]
Long lRet
...
la_1[1] = ls_sessionids

lRet = loadDBList.of_ExecInterface ( "killSessions", la_1 ) //result is returned to
loadDBList.ReturnValue
```

#### 4.1.8.2 Syntax for EJB component

```
io_ejb.regstring ( String sessionIDs )
```

```
io_ejb.Invokeretbool ( long objid, "killSessions", true, ref boolean retval )
```

**Table 4.19:**

Argument	Description
io_ejb	A reference of Appeon EJBobject.
sessionIDs	The ID of sessions which you want to kill. If there are multiple sessions, separate them with semicolons (;).
objid	The handle to the component method.

Argument	Description
<i>retval</i>	The return value of <b>killSessions</b> function: true if it succeeds and false if it fails. This parameter must be declared, but need not to be assigned with values, as it is used to hold the return value.

### Return value of Invokeretbool

Returns 0 if it succeeds and -1 if it fails.

### Return value of killSessions

Boolean.

It returns true if it succeeds and false if it fails. The return value is stored in the *retval* property of the **Invokeretbool** function.

### Code example

Note that **invokeretbool** function is called because the return value is a boolean.

```
string ls_msg
boolean retval01
...
io_ejb.regstring (ls_sessionids)

ls_msg = io_ejb.invokeretbool (il_beanc1, "killSessions", true, ref retval01) // result is returned to retval01
```

#### 4.1.8.3 Syntax for Web Service

**killSessions ( string sessionstring )**

**Table 4.20:**

Argument	Description
<i>sessions</i>	The ID of the session(s) which you want to kill. Multiple session IDs are separated with semicolons (:).

### Return value

Returns 1 if it succeeds and -1 if it fails.

### Code example

```
// -- connected with the web service
.....
string ls_return,ls_server,ls_app,ls_session
ls_server = sle_1.text
ls_app = sle_2.text
ls_session = sle_3.text
ls_return = ieon_ws.killsessions(ls_session)
mle_1.text = ls_return
```

#### 4.1.9 rollbackAllTransactions

### Description

**rollbackAllTransactions** rolls back all transactions in a PowerServer or a PowerServer cluster. To roll back all transactions in a PowerServer cluster, you need to first configure the cluster in AEM.

#### 4.1.9.1 Syntax for .NET component

*proxyobject.of\_execinterface ( "rollbackAllTransactions", ref any *paralist[]* )*

**Table 4.21:**

Argument	Description
<i>proxyobj</i>	The instance of the proxy object generated for AppeonDotNetComponent.
<i>paralist</i>	Arrays of Any type. Specifies the parameter arrays for the component function.  <i>paralist[1]</i> - The name of the PowerServer that you want to kill all sessions on. The value should be IP address or IP address with port number, for example, 192.0.0.123, or 192.0.0.123:8080, and separated with semicolons (;) if there are multiple servers (for example in a PowerServer cluster).  If <i>paralist[1]</i> is NULL or empty string (""), it rolls back all transactions in a PowerServer cluster.

#### Return value of *of\_execinterface*

Returns 0 if it succeeds and -1 if it fails.

#### Return value of *rollbackAllTransactions*

Boolean.

It returns true if it succeeds and false if it fails. The return value is stored in the *ReturnValue* property of the proxy object.

#### Code example

```
Any la_1[]
Long lRet
...
la_1[1] = ls_servername

lRet = loadDBList.of_ExecInterface ("rollbackAllTransaction", la_1) //result is
    returned to loadDBList.ReturnValue
```

#### 4.1.9.2 Syntax for EJB component

*io\_ejb.regstring ( String *serverName* )*

*io\_ejb.Invokeretbool ( long *objid*, "rollbackAllTransactions", true, ref boolean *retval* )*

**Table 4.22:**

Argument	Description
<i>io_ejb</i>	A reference of Appeon EJBobject.
<i>serverName</i>	The name of the PowerServer that you want to kill all sessions on. The value should be IP address or IP address with port number, for example, 192.0.0.123, or 192.0.0.123:8080, and separated with semicolons (;) if there are multiple servers (for example in a PowerServer cluster).  If <i>serverName</i> is NULL or empty string (""), it rolls back all transactions in a PowerServer cluster.
<i>objid</i>	The handle to the component method.

Argument	Description
<i>retval</i>	The return value of <b>rollbackAllTransactions</b> function: true if it succeeds and false if it fails. This parameter must be declared, but need not to be assigned with values, as it is used to hold the return value.

### Return value of Invokeretbool

Returns 0 if it succeeds and -1 if it fails.

### Return value of rollbackAllTransactions

Boolean.

It returns true if it succeeds and false if it fails. The return value is stored in the *retval* property of the **Invokeretbool** function.

### Code example

Note that **invokeretbool** function is called because the return value is a boolean.

```
string ls_msg
boolean retval01
...
io_ejb.regstring (ls_servername)

ls_msg = io_ejb.invokeretbool (il_beanc1, "rollbackAllTransaction", true, ref
 retval01) //result is returned to retval01
```

## 4.2 Calling PowerServer open interfaces via EJB component

PowerServer open interfaces (methods) in JBoss, JEUS, WebLogic and WebSphere are encapsulated in a standard EJB component named **OpenInterface** that is included in the **ASInterface** package. To invoke the open interfaces in PowerBuilder, you need to follow the instructions of how to call Appeon EJBObject. Details refer to [Section 1.3.6, “Calling EJB Component”](#).

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

Step 1: Load Appeon Workarounds PBL to the application.

Step 2: Deploy Appeon Bridge.

Step 3: Connect to the J2EE application server where PowerServer and Appeon Bridge are installed through Appeon EJBObject.

For example:

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

```
end if
```

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

The JNDI name for WebSphere, WebLogic, JEUS, & JBoss is **OpenInterfaceBean**.

For example:

```
String ls_jndi, ls_home, ls_method, ls_msg
ls_jndi = "OpenInterfaceBean"
ls_msg = io_ejb.createremoteinstance (ls_jndi, ls_home, "create", ref il_beans)
if ls_msg <> "" then
MessageBox ("Failed to create the remote instance!", ls_msg)
return
end if
```

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

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

Step 6: Deploy the PowerBuilder application to PowerServer the same way you would deploy a normal PowerBuilder application.

The above code will take effect in the PowerServer-deployed application.

### 4.3 Calling PowerServer open interfaces via .NET component

PowerServer open interfaces (methods) in .NET are encapsulated in a standard .NET component. To call the .NET component, Appeon PowerServer provides a non-autostarted NVO - **AppeonDotNetComponent** - as the proxy object to call the server-side component. Details refer to [Section 1.3.7, “Calling .NET/COM server components \(.NET only\)”](#).

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

Step 1: Load Appeon Workarounds PBL to the application.

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

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

```
Stringservername, appname
AppeonDotNetComponent loadDBList
Any la_1[]
Long lRet
serverName = ""
appname = ""
la_1[1] = serverName
la_1[2] = appname
```

```
loadDBList = Create AppeonDotNetComponent
```

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

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

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

```
lRet = loadDBList.of_ExecInterface ("getSessionCount", 1a_1)  
If lRet = 0 Then  
    MessageBox ("Get session count successfully.", "Session Count:" +  
        string(loadDBList.ReturnValue))  
End If
```

Step 6: Deploy the PowerBuilder application to PowerServer the same way you would deploy a normal PowerBuilder application.

The above code will take effect in the PowerServer-deployed application.

## 4.4 Calling PowerServer open interfaces via Web service

Here is the code example that shows you how to call the PowerServer open interfaces via Web service in the PowerBuilder application.

Step 1: Generate the proxy object for the Web service using the .NET Web service engine.

- URL for the open interface Web service in PowerServer installed to the Java application server (such as JBoss, WebLogic, WebSphere, or JEUS): `http://AppeonServerName:Port/servlet/AppeonService?wsdl`
- URL for the open interface Web service in PowerServer installed to the .NET IIS server: `http://AppeonServerName/servlet/AppeonService.asmx`

Step 2: Create the instance for the SOAP connection object and the Web service proxy object.

```
soapconnection  ieon_soap  
wsappeonservice ieon_ws      //the Web service proxy object generated by the .NET  
engine  
wsauthsoapheader ieon_head  
long ll_return  
string ls_return, ls_server, ls_app  
  
ieon_soap = create soapconnection  
ieon_head = create wsauthsoapheader
```

Step 3: Connect with the Web service, and pass in the user name and password for security authentication.

```
ll_return = ieon_soap.createinstance (ieon_ws, 'wsappeonservice')  
ieon_head.username = "admin"  
ieon_head.password = "admin"  
ieon_ws.setauthsoapheadervalue (ieon_head)
```

Step 4: Call the PowerServer open interface.

```
ls_server = sle_1.text  
ls_app = sle_2.text
```

```
ls_return = ieon_ws.getAllSessions (ls_server, ls_app)
```

# 5 Workarounds for Unsupported Features

Not all of the PowerBuilder features can be supported by PowerServer. 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.

## 5.1 Objects & Controls

### 5.1.1 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.

### 5.1.2 UserObject object

#### 5.1.2.1 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.

### 5.1.3 System Objects

#### 5.1.3.1 Application object

##### 5.1.3.1.1 DWMessagTitle property

#### Description

The DWMessagTitle is unsupported.

#### Workaround

Use a global variable to record the value of the DWMessagTitle property. Read and write the global variable instead of using the DWMessagTitle property in the Script.

### 5.1.3.1.2 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.

## 5.1.4 System Controls

### 5.1.4.1 ListView control

#### 5.1.4.1.1 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.

#### 5.1.4.1.2 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.

### 5.1.4.2 TreeView control

#### 5.1.4.2.1 SetDropHighlight function

#### Description

The SetDropHightlight 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.

### 5.1.4.3 Tab control

#### 5.1.4.3.1 TabPostEvent function

#### Description

TabPostEvent function of tab control is unsupported.

#### Workaround

Use TabTriggerEvent instead.

## 5.2 System Functions

### 5.2.1 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.

### 5.2.2 FileLength64 function

#### Description

The FileLength64 function is unsupported.

#### Workaround

Replace the FileLength64 function with FileLength function.

### 5.2.3 FileSeek64 function

#### Description

The FileSeek64 function is unsupported.

#### Workaround

Replace the FileSeek64 function with FileSeek function.

### 5.2.4 FileWriteEx function

The FileWriteEx function is supported since Appeon 6.5.

#### Description

The FileWriteEx function is unsupported in PowerServer 5.x, 6.0, 6.1 and 6.2.

#### Workaround

Replace the FileWriteEx function with FileWrite function.

### 5.2.5 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!)
```

## 5.2.6 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)
```

## 5.2.7 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.

## 5.2.8 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.

## 5.2.9 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 PowerServer Help as the HLP file. First upload the HLP file to the Web Server (in this example PowerServer 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 PowerServer 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" )
```

## 5.2.10 HyperLinkToURL function

### Description

While the HyperLinkToURL function is supported, it cannot link to the URL of a PowerServer application. This is because an Internet Explorer process only supports one PowerServer application at a time. Executing the HyperLinkToURL function from one PowerServer 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 a PowerServer application.

Step 2: At the Web server, copy the IEXPLORE.EXE to *C:\WINNT\system* or *C:\WINNT\sys32*. 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.

## 5.2.11 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.

## 5.2.12 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.

## 5.2.13 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 PowerServer database. For detailed information, please refer to **PowerServer Configuration Guide for .NET | AEM User Guide | Application | PB Features | Registry Mode**.

Note: Mobile applications can only read Appeon emulation registry.

## 5.2.14 Shared Object functions

### Description

Shared Object functions like SharedObjectGet, SharedObjectRegister are unsupported by PowerServer, 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.

### 5.2.15 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.

## 5.3 PowerScript Reference

### 5.3.1 PowerScript Topics

#### 5.3.1.1 Calling functions and events

##### 5.3.1.1.1 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])
```

Modified to

```
menu lm_menu
lm_menu = theobject.item[theitem]
poptags(lm_menu)
```

### 5.3.1.2 Declarations

#### 5.3.1.2.1 Shared variables

#### Description

Shared variables are unsupported. For example, In the *pfcmain.pbl* of a PFC application, the *pfc\_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.

### 5.3.1.3 Language basics

#### 5.3.1.3.1 Null values

##### Description

The Null value calculation in PowerServer 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 PowerServer 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 PowerServer
A+1	Null	1
A+B	Null	Null

Expressions	Return Value in PowerBuilder	Return Value in PowerServer
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

### 5.3.2 PowerScript Statements

#### 5.3.2.1 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
```

### 5.3.3 SQL Statements

#### 5.3.3.1 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_usp_codpes, &  
@nomsis = :s_usp_systemname, &  
@staobt = :s_usp_systemstatus,&  
@numseniva = null output ;
```

Modified code:

```
integer li_return  
SetNull(li_return)  
// Declare the procedures  
Declare SPgetseniva procedure for SPgetseniva &  
@codpes = :l_usp_codpes, &  
@nomsis = :s_usp_systemname, &  
@staobt = :s_usp_systemstatus,&  
@numseniva = :li_return output ;
```

### 5.3.4 PowerScript Events

#### 5.3.4.1 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.

#### 5.3.4.2 Other event

##### Description

The Other event is unsupported.

## Workaround

Move the script in the Other event to the supported events.

# 5.4 DataWindow Reference

## 5.4.1 DataWindow operators

### 5.4.1.1 DataWindow operator precedence

#### Description

The AND and OR operators in a DataWindow expression have the same precedence in PowerBuilder, but in PowerServer, 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 PowerServer, 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
```

## 5.4.2 DataWindow Object Properties

### 5.4.2.1 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.

## 5.4.3 DataWindow control

### 5.4.3.1 Functions

#### 5.4.3.1.1 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
```

#### 5.4.3.1.2 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.

#### 5.4.3.1.3 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.

#### 5.4.3.1.4 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.

#### 5.4.3.1.5 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.

#### 5.4.3.1.6 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)
```

#### 5.4.3.1.7 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)
```

#### 5.4.3.1.8 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)
```

#### 5.4.3.2 Events

##### 5.4.3.2.1 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_getappnifile( ), "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_verticalscroll()
next
```

Step 5: Add an instance variable for u\_dw.

```
long il_old_first_row = -1;
```

Step 6: Add a new function of\_sb\_verticalscroll( ) 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.

#### 5.4.3.2.2 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
next
```

#### 5.4.3.3 Properties

##### 5.4.3.3.1 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.

### 5.4.4 Controls in a DataWindow

#### 5.4.4.1 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.

#### 5.4.4.2 Column control

##### 5.4.4.2.1 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 PowerServer 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
```

#### 5.4.4.2.2 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 does not work well with a PowerServer 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, we 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
```

#### 5.4.5 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.

## 5.5 DBParm parameters in Database

### 5.5.1 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 3, *Database Connection Setup* in *PowerServer Configuration Guide for .NET* or PowerServer Configuration Guide for J2EE.

## 6 FAQ & Workarounds

This chapter lists some frequently asked questions and workaround tips regarding the PowerServer application architecture or product features.

### 6.1 What files will go to the plugin folder and how to access them

#### What files will go to the "plugin" folder?

For each PowerServer application, a "plugin" folder will be created automatically under the application directory (so called sandbox on the mobile device and so called cache directory on the Web client) to store any external files created or used by the application. These external files include

- Any files created by the API calls, such as photos or videos created by the [Camera APIs](#), bar code images created by the [Barcode APIs](#), log files created by the [of\\_log API](#) etc.
- Any files downloaded from the Appeon File Server, such as PDF files, image files, multimedia files, text files, etc.

For instructions on how to upload and download files from Appeon File Server, see [Configuring and deploying Appeon File Server](#) and [Uploading and downloading files](#).

- Any files deployed to PowerServer from PowerServer Toolkit and downloaded from the PowerServer at runtime, such as PDF files, image files, multimedia files, text files, DLL/OCX files, etc.

For instructions on how to deploy these files to PowerServer, see Deploy External Files in *PowerServer Toolkit User Guide*.

- Any files created by the SaveAs function of the DataWindow control, the DataWindowChild object, or the DataStore object, such as PDF files, text files, HTML files, excel files etc.

#### How files go there?

All of the above files are automatically saved to the "plugin" folder, no matter they are new files created by the API or function calls or pre-existing files downloaded from the server to the client.

You can also save a file to the blob fields in the database, then read the blob data via the SQL statement, and then write the file to the "plugin" folder by using the PB FileWrite function.

#### How to reference these files in the script?

To programmatically use the files under the "plugin" folder, you will need to take advantage of a function provided by PowerServer, to get the full directory of the application directory. You can either use the [AppeonGetCacheDir](#) global function or the [of\\_getcachedir function](#) of the AppeonExtFuncs object. They work exactly the same. For example, `appeongetcachedir() + "/plugin/test.pdf"` or `of_getcachedir() + "/plugin/test.pdf"`.

For some functions, such as the PB file functions (except for operating the INI files), the directory is automatically default to the "plugin" directory. For example, `FileExists ("c:\test\test.txt")` will be executed as `FileExists (appeongetcachedir() + "/plugin/test.txt")`.

## 6.2 How to generate a PDF file and send it via email

Follow steps below to generate a PDF file and send it via email on a mobile device:

Step 1: Generate the PDF file.

- Method 1: Generate the PDF file by calling the SaveAs function of the DataWindow control, the DataWindowChild object, or the DataStore object.

Syntax:

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

For example:

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

Note:

- filename cannot be empty string or null value.
  - The file path will be ignored, because the generated file is automatically saved to the %cachedir%/plugin/ directory (%cachedir% indicates the cache directory for the current mobile application). You can determine the cache directory by calling the [AppeonGetCacheDir function](#), as shown in the code example below for step 2.
- Method 2: Download an existing PDF file from the Appeon File Server to the mobile device.

The PDF file will be automatically downloaded to the %cachedir%/plugin/ directory. You can determine the cache directory by calling the [AppeonGetCacheDir function](#), as shown in the code example below for step 2.

You can follow the instructions in [Configuring and deploying Appeon File Server](#) to install and configure the Appeon File Server and [Uploading and downloading files](#) to upload and download the PDF file from the Appeon File Server.

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_mobile_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 have an email account configured already on the mobile device to send the PDF file via email as attachment.

## 6.3 How to make a phone call in the mobile application

Although Appeon PowerServer provides no APIs to call a phone number within the mobile application, you can still implement it using the following code (effective for both iOS and Android applications):

```
inet l_inet
string ls_code, ls_protocol
int rc

ls_code = "800-555-1212"
ls_protocol= "tel:"

rc = messagebox (ls_code~r~n, 'Dial it?', question!, yesno!, 1)

if rc = 1 then
    l_inet = create inet
    l_inet.hyperlinktouri ( ls_protocol+ls_code )
    Destroy l_inet
end if
```

## 6.4 How to open Google Map in the mobile application

You can call the Google Map native app or the Google Map Web app, depending on the mobile OS. For iOS, you can either call the native app or the Web app; while for Android, you can only call the Web app.

### To call the Google Map native app:

First define a function to call the Google Map native app via its URL scheme. Note that URL scheme is supported in iOS only.

```
public function integer of_gm_direction (string as_saddr, string as_daddr , string
as_dirmode);

// Only iOS can support google map URL Scheme
```

```
// as_dirmode:driving,transit,bicycling,walking

string ls_url, ls_urlscheme
inet l_inet
integer li_re

ls_urlscheme = "comgooglemaps://?saddr="+as_saddr+"&daddr="+as_daddr
+"&dirflag="+as_dirmode

l_inet = create inet
li_re = l_inet.hyperlinktouri(ls_urlscheme)

destroy l_inet
end function
```

Call the function by passing in the start address, destination address, and direction mode. For example:

- The address can either be the name of places (using + as separator) or the latitude and longitude.
- The direction mode can be: driving, transit, bicycling, walking.

```
of_gm_direction("22.549721,113.948371","22.5584634,114.1337181","bicycling")
```

Or

```
of_gm_direction("appeon+sz+Guangdong+china","22.5584634,114.1337181","driving")
```

### To call the Google Map Web app:

First define a function to call the Google Map Web app via URL and parameters.

```
public function integer of_gmweb_direction (string as_saddr, string as_daddr ,
  string as_dirmode);

// as_dirmode:d,t,b,w

string ls_url, ls_as_dirmode4web
inet l_inet
integer li_re

ls_url = "https://maps.google.com/maps?saddr="+as_saddr+"&daddr="+as_daddr
+"&dirflag="+ls_as_dirmode4web

l_inet = create inet
li_re = l_inet.hyperlinktouri(ls_url)
Destroy l_inet
return li_re
end function
```

Call the function by passing in the start address, destination address, and direction mode. For example,

- The address can either be the name of places (using + as separator) or the latitude and longitude.
- The direction mode can be: d (for driving), t (for transit), b (for bicycling), w (for walking).

```
of_gmweb_direction("22.549721,113.948371","22.5584634,114.1337181","b")
```

Or

```
of_ gmweb _direction( "appeon+sz+Guangdong+china" , "22.5584634,114.1337181" , "d" )
```

## 6.5 How to download an image to the mobile device and show it in the application

Follow steps below to download an image file to the mobile device and show the picture in the mobile application:

Step 1: Install the Appeon File Server. Detailed instructions can be found in [Configuring and deploying Appeon File Server](#).

Step 2: Upload the image file to the Appeon File Server, and then download it from the Appeon File Server to the mobile device. Here are the specific steps:

1. Add Appeon Workarounds PBL to your PowerBuilder application.
2. Define a variable and create an instance for the AppeonFileService object.
3. Call of\_logonfileserver to connect to the file server.
4. Call of\_appeonupload to upload the image file to the file server.
5. Call of\_appeondownload to download the image file from the file server to the mobile device.

The file is automatically saved to the %cachedir%/plugin/ directory. %cachedir% indicates the cache directory for the current mobile application. You can determine the cache directory by calling the [AppeonGetCacheDir function](#).

6. Call of\_logofffileserver to disconnect from the file server.

Step 3: Access the image file through AppeonGetCacheDir() + "/plugin/" + %filename% in your mobile application.

**Alternatively**, if you only need to show the image file in the application and does not need to download the image to the mobile device, you can also take advantage of an Appeon enhanced feature to efficiently implement it. Appeon has enhanced the PictureName property of the Picture/PictureButton/PictureHyperLink control, the DataWindow column's content (when Display as Picture option is on), and the DataWindow Bitmap function to support the Internet URL. Therefore, you can post the image file to a Web site that is accessible by the mobile device, and utilize the aforementioned features to directly show the picture in the mobile application.

## 6.6 How to replace the icon that shows in the IE tab or address bar

### Description

When you launch an application in Internet Explorer, an icon shows in the IE tab and the address bar. By default, the icon is the IE icon, but you can change it to any other icon.

### Workaround

Step 1: Prepare an icon that you want to display in the IE tab and the address bar. Name it to "app.ico".

Step 2: Go to the Web folder where the application is deployed, for example, C:\Inetpub\wwwroot\codeexamples, copy and paste the **app.ico** file to the **images** folder, and modify the **x32\_application.htm** and **x64\_application.htm** files by following instructions below:

The original code:

```
<head>
  <title>Appeon Web Library</title>
</head>
```

Should be added with the following line of script (in bold text):

```
<head>
  <link rel="Shortcut Icon" href=".\\images\\app.ico">
  <title>Appeon Web Library</title>
</head>
```

## 6.7 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 PowerServer, but you can change it to any other image.

### Workaround

Once you change the PowerServer-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 PowerServer-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: %powerserver%\weblibrary\_ax\debug\image\ and %powerserver%\weblibrary\_ax\release\image\.

## 6.8 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 PowerServer 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://appeonserver: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.

## 6.9 How to deploy a PowerServer application without PowerServer Toolkit

### Description

PowerServer Toolkit is needed to deploy the application to at least one PowerServer and Web Server. Once deployed to a PowerServer and a Web Server, you can replicate the deployed application to other PowerServer and Web Servers without using PowerServer Toolkit.

## Workaround

Use the Package Wizard provided in the PowerServer Toolkit to generate a portable installation package for your PowerServer deployed application and install the package to the other PowerServer and Web Servers. For step-by-step instructions on how to package and install applications, please refer to Chapter 9, *Packaging Applications in PowerServer Toolkit User Guide*.

## 6.10 How to log in the PowerServer Web application with single sign-on

Suppose you have a PowerServer Web application and another Web application (created using Visual Studio, COBOL, Web Services, SOA, etc.); and you want to access the PowerServer Web application from another Web application (suppose called ABC) without needing to log in again since you have already logged into the application ABC. In such a scenario, you can pass the user credentials to the PowerServer Web application without requiring re-login using either of the following two methods.

### Method 1: Apply the command line argument.

PowerServer 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 a PowerServer application in the following way:

`http://192.0.1.94:8080/MyTest/index.html?user=appeon&password=appeon`

This attaches the string "`index.html?arguments`" to the end of the original application URL ("`index.html`" must be included in the string).

If the user wants to launch a PowerServer application after logging on to another application (created using Visual Studio, COBOL, Web Services, SOA, etc.), the application passes the user information via the URL of the PowerServer application, and the user starts the PowerServer 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 application, with a session ID assigned to each session, and the session information containing user information. When the PowerServer 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 PowerServer application without further login procedures.

### Method 2: Use the single sign-on server to pass user information.

If you have set up the single sign-on server, you can take advantage of the single sign-on server to pass the user credentials. Normally the single sign-on server provides the authentication mechanism interface via Web service/EJB/JSP function etc. Make sure that your PowerBuilder can support the interface, then the PowerServer Web application should work with it.

## 6.11 How to add headers & footers to a Web application

### Description

Use this solution to add headers and footers to a corporate website with a PowerServer 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.

## 6.12 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 PowerServer 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).

## 6.13 How to modify the storage location of Web application files in PowerServer Toolkit

PowerServer Toolkit will generate and store Web application files on the local machine before deploying it to PowerServer. You can configure to store these files to other location rather than the default one. By default the Web files will be stored in %Appeon\_Developer%\Project%\application\_name% (e.g., C:\Program Files\Appeon\PowerServer\Toolkit\Project\sales). %Appeon\_Developer% indicates the installation path of PowerServer Toolkit.

### How to configure the location for Web files

Step 1: Open the ADTConfig.xml file in the directory Appeon\PowerServer\Toolkit.

Step 2: Find the following lines:

```
<PathCfg>
<Project value="%Appeon%\PowerServer\Toolkit\Project\"/>
</PathCfg>
```

Step 3: Replace the value in bold with another location where you want to store the Web files.

## 6.14 How to integrate PowerServer Web applications with JSP/ASP

### 6.14.1 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 a PowerServer application

Apply the JSP/ASP programming method to add parameters to the URL of the PowerServer application. Based upon the parameters, any functionality can be built in the PowerServer Web application, such as opening windows and retrieving data to provide client-side integration.

The PowerServer application receives the parameters using the CommandParm function and CommandLine parameter of the Open event.

Workaround for passing parameters from a PowerServer application to a JSP/ASP application

PowerServer 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 PowerServer application to a JSP/ASP application.

PowerServer 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.

### 6.14.2 Using Internet Explorer Frame

#### Description

PowerServer applications can be accessed in an Internet Explorer frame. It is possible to set the PowerServer 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 PowerServer application and one for running the ASP/JSP application.

Step 2: Apply the Appeon CommandParm feature for launching the PowerServer application from the ASP/JSP application.

Step 3: Apply the Appeon Hyperlink feature for launching the ASP/JSP application from the PowerServer application.

### 6.14.3 Integration through intermediate n-Tier Server-level solutions

#### Description

It is possible to pass parameters between PowerServer 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.

### 6.15 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](mailto:support@appeon.com)> for help.

# Index

- eon\_editmask, [73](#)
- eon\_singlelineedit, [73](#)
- eon\_statictext, [73](#)
- ii\_keyboardtype, [73](#)
- ii\_vertical, [73](#)
- A**
- Appeon Client Functions, [15](#)
- Appeon Commit Label, [45](#)
- Appeon Commit/Rollback Label, [44](#)
- Appeon Immediate Call Label, [49](#)
- Appeon Labels, [43](#)
- Appeon Queue Labels, [47](#)
- Appeon requirements for EJB development, [63](#)
- Appeon resize object, [89](#)
- Appeon Resize PBL Reference, [206](#)
- Appeon Rollback Label, [46](#)
- Appeon Update Label, [50](#)
- Appeon Workarounds PBL Reference, [1](#)
- Appeon Workspace
  - eon\_mobile\_str\_mailcontent, [122](#)
  - oe\_urlschemesucceed, [103](#)
  - of\_checkpermissionforonet, [103](#)
  - of\_clearlog, [104](#)
  - of\_getappinfo, [105](#)
  - of\_getapportionation, [105](#)
  - of\_getapportionationlock, [106](#)
  - of\_getassistivetouchbtnvisible, [106](#)
  - of\_getassistivetouchmode, [107](#)
  - of\_getcloseappiconvisible, [107](#)
  - of\_getdwinputmode, [108](#)
  - of\_getdwmousemovemode, [108](#)
  - of\_gettitlebarvisible, [109](#)
  - of\_geturlschemeparm, [109](#)
  - of\_getversion, [110](#)
  - of\_getwindowlisticonvisible, [110](#)
  - of\_getworkspaceheight, [111](#)
  - of\_getworkspaceidappname, [111](#)
  - of\_getworkspacewidth, [112](#)
  - of\_log, [112](#)
  - of\_register, [113](#)
  - of\_sendmail, [113](#)
  - of\_setapportionation, [114](#)
  - of\_setapportionationlock, [115](#)
  - of\_setassistivetouchbtnvisible, [116](#)
- of\_setassistivetouchmode, [116](#)
- of\_setcloseappiconvisible, [117](#)
- of\_setdwinputmode, [118](#)
- of\_setdwmousemovemode, [119](#)
- of\_setlongpressinterval, [120](#)
- of\_settitlebarvisible, [121](#)
- of\_setwindowlisticonvisible, [121](#)
- of\_triggerevent, [122](#)
- AppeonDotNetComponent object, [65](#)
- AppeonExtFuncs object, [2](#)
- AppeonFileService object, [37](#)
- AppeonGetAppeonUserName function, [15](#)
- AppeonGetBrowserVersion function, [15](#)
- AppeonGetCacheDir function, [15](#)
- AppeonGetClientID function, [16](#)
- AppeonGetClientIP function, [16](#)
- AppeonGetClientType function, [16](#)
- AppeonGetClientVersion function, [17](#)
- AppeonGetHttpInfo function, [17](#)
- AppeonGetIEHandle function, [17](#)
- AppeonGetIEURL function, [18](#)
- AppeonGetOSType function, [18](#)
- AppeonGetRemainingdays function, [18](#)
- AppeonGetServerType function, [19](#)
- AppeonGetSessionCount function, [19](#)
- appeonisin64browser function, [19](#)
- AppeonLDAPLogon function, [20](#)
- AppeonNetworkTest function, [20](#)
- AppeonPopupMenu function, [21](#)
- AppeonPopupMenuOn function, [21](#)
- AppeonPrint2File function, [22](#)
- AppeonPrint2PDF function, [23](#)
- AppeonSetHintText function, [23](#)
- AppeonSetLongPressInterval function, [24](#)
- AppeonSwitchRealTimeCalc function, [24](#)
- appeonwebservicecomponent object, [69](#)
- Application object, [289](#)
- Applying Appeon CommandParm and Hyperlink features, [318](#)
- Audio
  - ii\_errorcode, [99](#)
  - ipo\_bindevent, [99](#)
  - is\_bindwitherrorevent, [99](#)
  - is\_errortext, [99](#)
  - oe\_error, [99](#)
  - of\_getstatus, [100](#)
  - of\_pauseorresume, [100](#)
  - of\_play, [101](#)

of\_register, [101](#)

of\_stop, [102](#)

of\_triggerevent, [102](#)

## B

Barcode

of\_create, [123](#)

of\_getfiledata, [125](#)

of\_read, [126](#)

Best practices for calling APIs, [1](#)

bundle EJB proxy with Appeon Bridge, [53](#)

## C

call EJB Component, [52](#)

call EJB component in PowerBuilder application, [54](#)

Calling .NET/COM server components (.NET only), [64](#)

Calling functions and events, [295](#)

Calling PowerServer open interfaces via .NET component, [286](#)

Calling PowerServer open interfaces via EJB component, [285](#)

Calling Web Service, [68](#)

Camera

eon\_mobile\_str\_cameraoption, [131](#)

of\_getfiledata, [127](#)

of\_openalbums, [128](#)

of\_takefile, [129](#)

CanUndo function, [301](#)

Char data type column, [306](#)

Client Log, [25](#)

Code Examples, [68, 71](#)

Column control, [306](#)

Configuring and deploying Appeon File Server, [25](#)

Configuring the Appeon File Server, [29, 30, 34, 35, 37](#)

Connection

of\_getconnectioninfo, [132](#)

ConnectServer function, [57](#)

ConnectionString parameter, [308](#)

Constructor, [67, 70](#)

Controls in a DataWindow, [306](#)

CreateRemoteInstance function, [62](#)

## D

DataWindow control, [301](#)

DataWindow Object Properties, [300](#)

DataWindow operator precedence, [300](#)

DataWindow operators, [300](#)

DataWindow Reference, [300](#)

DBCANCEL function, [301](#)

DBParm parameters in Database, [308](#)

DDE Server functions, [291](#)

Declarations, [296](#)

DeleteItem function, [289](#)

Deploying the Appeon File Server, [32, 34, 35, 37](#)

DestroyRemoteInstance function, [63](#)

Destructor, [67, 71](#)

Device

of\_getdeviceid, [133](#)

of\_getdevicetype, [134](#)

of\_getdpi, [134](#)

of\_getfreememory, [135](#)

of\_getname, [136](#)

of\_getorientation, [136](#)

of\_getosversion, [138](#)

of\_getplatform, [137](#)

of\_getppi, [135](#)

of\_getresolution, [138](#)

of\_getstatusbarvisible, [139](#)

of\_gettotalmemory, [135](#)

of\_setstatusbarvisible, [139](#)

DisConnection function, [58](#)

Duplicate arguments for a function, [295](#)

DWMessagTitle property, [289](#)

## E

EJBObject object, [57](#)

eon\_appeon\_resize object, [89](#)

eon\_cordova\_base object, [208](#)

eon\_cordova\_bluetooth object, [212](#)

eon\_cordova\_bluetoothle object, [215](#)

eon\_cordova\_contact object, [223](#)

eon\_cordova\_fingerprint object, [239](#)

eon\_cordova\_gps object, [241](#)

eon\_cordova\_orientation object, [244](#)

eon\_cordova\_vibration object, [247](#)

eon\_editmask

ii\_keyboardtype, [73](#)

ii\_vertical, [73](#)

eon\_logex, [25](#)

eon\_mobile\_audioex

ii\_errorcode, [99](#)

ipo\_bindevent, [99](#)

is\_bindwitherrorevent, [99](#)  
is\_errortext, [99](#)  
oe\_error, [99](#)  
of\_getstatus, [100](#)  
of\_pauseorresume, [100](#)  
of\_play, [101](#)  
of\_register, [101](#)  
of\_stop, [102](#)  
of\_triggerevent, [102](#)  
**eon\_mobile\_awsex**  
  eon\_mobile\_str\_mailcontent, [122](#)  
  oe\_urlschemesucceed, [103](#)  
  of\_checkpermissionforntet, [103](#)  
  of\_clearlog, [104](#)  
  of\_getappinfo, [105](#)  
  of\_getapportionment, [105](#)  
  of\_getapprotationlock, [106](#)  
  of\_getassistivetouchbtnvisible, [106](#)  
  of\_getassistivetouchmode, [107](#)  
  of\_getcloseappiconvisible, [107](#)  
  of\_getdwinputmode, [108](#)  
  of\_getdwmousemovemode, [108](#)  
  of\_gettitlebarvisible, [109](#)  
  of\_geturlschemeparm, [109](#)  
  of\_getversion, [110](#)  
  of\_getwindowlisticonvisible, [110](#)  
  of\_getworkspaceheight, [111](#)  
  of\_getworkspaceidappname, [111](#)  
  of\_getworkspacewidth, [112](#)  
  of\_log, [112](#)  
  of\_register, [113](#)  
  of\_sendmail, [113](#)  
  of\_setapportionment, [114](#)  
  of\_setapprotationlock, [115](#)  
  of\_setassistivetouchbtnvisible, [116](#)  
  of\_setassistivetouchmode, [116](#)  
  of\_setcloseappiconvisible, [117](#)  
  of\_setdwinputmode, [118](#)  
  of\_setdwmousemovemode, [119](#)  
  of\_setlongpressinterval, [120](#)  
  of\_settitlebarvisible, [121](#)  
  of\_setwindowlisticonvisible, [121](#)  
  of\_triggerevent, [122](#)  
**eon\_mobile\_barcodeex**  
  of\_create, [123](#)  
  of\_getfiledata, [125](#)  
  of\_read, [126](#)  
**eon\_mobile\_cameraex**  
  eon\_mobile\_str\_cameraoption, [131](#)  
  of\_getfiledata, [127](#)  
  of\_openalbums, [128](#)  
  of\_takefile, [129](#)  
**eon\_mobile\_connectionex**  
  of\_getconnectioninfo, [132](#)  
**eon\_mobile\_deviceex**  
  of\_getdeviceid, [133](#)  
  of\_getdevicetype, [134](#)  
  of\_getdpi, [134](#)  
  of\_getfreememory, [135](#)  
  of\_getname, [136](#)  
  of\_getorientation, [136](#)  
  of\_getosversion, [138](#)  
  of\_getplatform, [137](#)  
  of\_getppi, [135](#)  
  of\_getresolution, [138](#)  
  of\_getstatusbarvisible, [139](#)  
  of\_gettotalmemory, [135](#)  
  of\_setstatusbarvisible, [139](#)  
**eon\_mobile\_geolocationex**  
  code Example, [147](#)  
  eon\_mobile\_str\_coordinates, [147](#)  
  ii\_errorcode, [140](#)  
  ipo\_bindevent, [141](#)  
  is\_bindwitherrorevent, [141](#)  
  is\_bindwithlocationchangedevent, [141](#)  
  is\_errortext, [141](#)  
  oe\_error, [141](#)  
  oe\_locationchanged, [141](#)  
  of\_close, [142](#)  
  of\_getcurrentposition, [143](#)  
  of\_isenabled, [144](#)  
  of\_open, [145](#)  
  of\_register, [146](#)  
**eon\_mobile\_mapex**  
  eon\_mobile\_str\_annotation, [161](#)  
  eon\_mobile\_str\_mapoption, [160](#)  
  ii\_errorcode, [152](#)  
  istr\_clickedannotation, [152](#)  
  is\_errortext, [153](#)  
  oe\_annotationclicked, [153](#)  
  oe\_changed, [154](#)  
  oe\_error, [154](#)  
  oe\_loadend, [153](#)  
  oe\_loadstart, [153](#)  
  of\_addannotation, [154](#)  
  of\_addressescoordinate, [155](#)

of\_close, [156](#)  
of\_coordinatetoaddress, [156](#)  
of\_open, [157](#)  
of\_refresh, [158](#)  
of\_removeallannotation, [159](#)  
of\_removeannotation, [159](#)  
of\_selectannotation, [160](#)  
eon\_mobile\_mediaex  
    of\_play, [162](#)  
eon\_mobile\_nfcecx  
    code example, [176](#)  
    eon\_mobile\_str\_nfcrecord, [175](#)  
    ii\_errorcode, [163](#)  
    ii\_nfccardtype, [163](#)  
    ipo\_bindevent, [164](#)  
    is\_bindwitherrorevent, [164](#)  
    is\_bindwithnewtagfoundevent, [164](#)  
    is\_errortext, [164](#)  
    oe\_error, [165](#)  
    oe\_newtagfound, [164](#)  
    of\_addrecord, [165](#)  
    of\_close, [165](#)  
    of\_getblocksize, [166](#)  
    of\_getmaxblockcount, [166](#)  
    of\_getmaxsectorcount, [167](#)  
    of\_getndefrecord, [167](#)  
    of\_getndefrecordcount, [168](#)  
    of\_getsize, [168](#)  
    of\_gettagid, [169](#)  
    of\_gettimeout, [169](#)  
    of\_open, [170](#)  
    of\_readblock, [170](#)  
    of\_register, [171](#)  
    of\_setkey, [171](#)  
    of\_settimeout, [172](#)  
    of\_startblockoperate, [173](#)  
    of\_stopblockoperate, [173](#)  
    of\_triggerevent, [173](#)  
    of\_writeblock, [174](#)  
    of\_writerecords, [175](#)  
eon\_mobile\_notificationex  
    of\_addmessage, [178](#)  
    of\_removemessage, [179](#)  
eon\_mobile\_paymentex  
    code example, [190](#)  
    eon\_mobile\_str\_paymentinit, [189](#)  
    eon\_mobile\_str\_paymentssubmit, [189](#)  
    ii\_appplatformtype, [184](#)  
            ii\_errorcode, [184](#)  
            ipo\_bindevent, [184](#)  
            is\_bindwitherrorevent, [184](#)  
            is\_errortext, [184](#)  
            is\_paymentcancel, [184](#)  
            is\_paymentfailed, [184](#)  
            is\_paymentok, [185](#)  
            oe\_error, [185](#)  
            oe\_paymentcancel, [185](#)  
            oe\_paymentfailed, [185](#)  
            oe\_paymentok, [185](#)  
            of\_getappplatform, [186](#)  
            of\_init, [186](#)  
            of\_register, [187](#)  
            of\_setappplatform, [188](#)  
            of\_submit, [188](#)  
            of\_triggerevent, [189](#)  
    eon\_mobile\_str\_annotation, [161](#)  
    eon\_mobile\_str\_cameraoption, [131](#)  
    eon\_mobile\_str\_coordinates, [147](#)  
    eon\_mobile\_str\_imageoption, [150](#)  
    eon\_mobile\_str\_mailcontent, [122](#)  
    eon\_mobile\_str\_mapoption, [160](#)  
    eon\_mobile\_str\_nfcrecord, [175](#)  
    eon\_mobile\_str\_paymentinit, [189](#)  
    eon\_mobile\_str\_paymentssubmit, [189](#)  
    eon\_mobile\_str\_textcheckoption, [196](#)  
    eon\_mobile\_textcheckex  
        code example, [196](#)  
        eon\_mobile\_str\_textcheckoption, [196](#)  
        of\_completionsforpartialword, [190](#)  
        of\_getignoredwords, [191](#)  
        of\_getlanguages, [192](#)  
        of\_getmisspelledword, [193](#)  
        of\_guessesforword, [193](#)  
        of\_haslearnedword, [194](#)  
        of\_learnword, [195](#)  
        of\_setignoredwords, [192](#)  
        of\_unlearnword, [195](#)  
    eon\_mobile\_webviewex  
        ii\_errorcode, [197](#)  
        is\_errortext, [197](#)  
        oe\_error, [197](#)  
        oe\_jcallbackappeon, [198](#)  
        oe\_loadend, [199](#)  
        oe\_loadstart, [199](#)  
        of\_cangoback, [199](#)  
        of\_cangoforward, [200](#)

of\_goback, [200](#)  
of\_goforward, [200](#)  
of\_loadhtmlstring, [201](#)  
of\_loadlocalfile, [202](#)  
of\_reload, [203](#)  
of\_runjavascript, [203](#)  
of\_startloading, [204](#)  
of\_stoploading, [205](#)  
eon\_offlineex  
    of\_applydbupdate, [180](#)  
    of\_applyupdate, [181](#)  
    of\_checkappeonserver, [181](#)  
    of\_checkupdate, [183](#)  
eon\_singlelineedit  
    ii\_vertical, [73](#)  
eon\_statictext  
    ii\_vertical, [73](#)  
Evaluating DataWindow expressions in scripts, [307](#)  
Events for AppeonDotNetComponent, [67](#)  
Events for appeonwebservicecomponent, [70](#)  
Events for DataWindow control, [303](#)  
External user object, [289](#)

## F

FAQ & Workarounds, [309](#)  
File Upload and Download, [25](#)  
FileLength64 function, [291](#)  
FileSeek64 function, [291](#)  
FileWriteEx function, [291](#)  
FreeDBLibraries property, [290](#)  
FromAnsi function, [291](#)  
FromUnicode function, [292](#)  
Functions for AppeonDotNetComponent, [66](#)  
Functions for AppeonFileService object, [37](#)  
Functions for appeonwebservicecomponent, [69](#)  
Functions for DataWindow control, [301](#)  
Functions of DataStore, [307](#)

## G

Garbage Collection functions, [292](#)  
General API, [1](#)  
generate the EJB proxy, [52](#)  
GenerateResultSet function, [301](#)  
Geolocation, [140](#)  
GetItemAtPointer function, [290](#)  
GetLibraryList function, [292](#)  
GetTrans function, [301](#)

GOTO statement, [298](#)  
GPS  
    code Example, [147](#)  
    eon\_mobile\_str\_coordinates, [147](#)  
    ii\_errorcode, [140](#)  
    ipo\_bindevent, [141](#)  
    is\_bindwitherrorevent, [141](#)  
    is\_bindwithlocationchangedevent, [141](#)  
    is\_errortext, [141](#)  
    oe\_error, [141](#)  
    oe\_locationchanged, [141](#)  
    of\_close, [142](#)  
    of\_getcurrentposition, [143](#)  
    of\_isenabled, [144](#)  
    of\_open, [145](#)  
    of\_register, [146](#)

## H

Help event, [299](#)  
Help functions, [292](#)  
How to add headers & footers to a Web application, [316](#)  
How to deploy a PowerServer application without PowerServer Toolkit, [315](#)  
How to get the user name and password of the operating system, [317](#)  
How to integrate PowerServer Web applications with JSP/ASP, [318](#)  
How to log in the PowerServer Web application with single sign-on, [316](#)  
How to modify the storage location of Web application files in PowerServer Toolkit, [317](#)  
How to remove the Internet Explorer menu, [314](#)  
How to replace Appeon image that displays at the running of applications, [314](#)  
HyperLinkToURL function, [293](#)

## I

il\_Id, [43](#)  
InitLocalLanguage function, [59](#)  
Installing the Appeon File Server, [26](#)  
integrate with Cordova plugins, [207](#)  
Integration through intermediate n-Tier Server-level solutions, [318](#)  
Introduction to Appeon Cordova PBL, [207](#)  
Introduction to Appeon Workarounds, [1](#)  
Invoking component functions, [59](#)  
is\_FileServerName, [43](#)

is\_LogOnParams, [43](#)  
ItemActivate event, [290](#)

## L

Language basics, [297](#)  
Large Binary/Text database OLE object, [306](#)  
LibraryDirectory function, [293](#)  
ListView control, [290](#)  
LiveScroll property, [305](#)  
LookUpJndi function, [59](#)

## M

Map  
eon\_mobile\_str\_annotation, [161](#)  
eon\_mobile\_str\_mapoption, [160](#)  
ii\_errorcode, [152](#)  
istr\_clickedannotation, [152](#)  
is\_errortext, [153](#)  
oe\_annotationclicked, [153](#)  
oe\_changed, [154](#)  
oe\_error, [154](#)  
oe\_loadend, [153](#)  
oe\_loadstart, [153](#)  
of\_addannotation, [154](#)  
of\_addressescoordinate, [155](#)  
of\_close, [156](#)  
of\_coordinatetoaddress, [156](#)  
of\_open, [157](#)  
of\_refresh, [158](#)  
of\_removeallannotation, [159](#)  
of\_removeannotation, [159](#)  
of\_selectannotation, [160](#)

Media  
of\_play, [162](#)  
Mobile Device API, [99](#)

## N

NFC  
code example, [176](#)  
eon\_mobile\_str\_nfcrecord, [175](#)  
ii\_errorcode, [163](#)  
ii\_nfccardtype, [163](#)  
ipo\_bindevent, [164](#)  
is\_bindwitherrorevent, [164](#)  
is\_bindwithnewtagfoundevent, [164](#)  
is\_errortext, [164](#)  
oe\_error, [165](#)  
oe\_newtagfound, [164](#)  
of\_addrecord, [165](#)

of\_close, [165](#)  
of\_getblocksize, [166](#)  
of\_getmaxblockcount, [166](#)  
of\_getmaxsectorcount, [167](#)  
of\_getndefrecord, [167](#)  
of\_getndefrecordcount, [168](#)  
of\_getsize, [168](#)  
of\_gettagid, [169](#)  
of\_gettimeout, [169](#)  
of\_open, [170](#)  
of\_readblock, [170](#)  
of\_register, [171](#)  
of\_setkey, [171](#)  
of\_settimeout, [172](#)  
of\_startblockoperate, [173](#)  
of\_stopblockoperate, [173](#)  
of\_triggerevent, [173](#)  
of\_writeblock, [174](#)  
of\_writerecords, [175](#)

Notification  
of\_addmessage, [178](#)  
of\_removemessage, [179](#)  
Null values, [297](#)

## O

Objects & Controls, [289](#)  
oe\_annotationclicked, [153](#)  
oe\_changed, [154](#)  
oe\_error (Audio object), [99](#)  
oe\_error (Geolocation object), [141](#)  
oe\_error (Map object), [154](#)  
oe\_error (NFC object), [165](#)  
oe\_error (Payment object), [185](#)  
oe\_error (WebView object), [197](#)  
oe\_jcallbackappeon (WebView object), [198](#)  
oe\_loadend, [153, 199](#)  
oe\_loadstart, [153, 199](#)  
oe\_locationchanged, [141](#)

oe\_newtagfound, [164](#)  
oe\_paymentcancel, [185](#)  
oe\_paymentfailed, [185](#)  
oe\_paymentok, [185](#)  
oe\_urlschemesucceed, [103](#)

Offline  
of\_applydbupdate, [180](#)  
of\_applyupdate, [181](#)  
of\_checkappeonserver, [181](#)  
of\_checkupdate, [183](#)

of\_addannotation, [154](#)  
of\_addmessage, [178](#)  
of\_addrecord, [165](#)  
of\_addressesstocoordinate, [155](#)  
of\_appeondownload, [41](#)  
of\_appeonupload, [39](#)  
of\_applydbupdate, [180](#)  
of\_applyupdate, [181](#)  
of\_callwebservice, [69](#)  
of\_cangoback, [199](#)  
of\_cangoforward, [200](#)  
of\_checkappeonserver, [181](#)  
of\_checkpermissionfornet (Appeon Workspace), [103](#)  
of\_checkupdate, [183](#)  
of\_clearlog, [4](#)  
of\_clearlog (Appeon Workspace), [104](#)  
of\_close (Geolocation object), [142](#)  
of\_close (Map object), [156](#)  
of\_close (NFC object), [165](#)  
of\_completionsforpartialword, [190](#)  
of\_coordinatestoaddress, [156](#)  
of\_create, [123](#)  
of\_execinterface, [66](#)  
of\_FileExists, [42](#)  
of\_getappeonusername function, [4](#)  
of\_getappinfo, [105](#)  
of\_getapporientation, [105](#)  
of\_getappplatform, [186](#)  
of\_getapprotationlock, [106](#)  
of\_getassistivetouchbtnvisible, [106](#)  
of\_getassistivetouchmode, [107](#)  
of\_getblocksize, [166](#)  
of\_getbrowserversion function, [4](#)  
of\_getcachedir function, [5](#)  
of\_getclientid function, [5](#)  
of\_getclientip function, [6](#)  
of\_getclienttype function, [6](#)  
of\_getcloseappiconvisible, [107](#)  
of\_getconnectioninfo, [132](#)  
of\_getcurrentposition, [143](#)  
of\_getdeviceid, [133](#)  
of\_getdevicetype, [134](#)  
of\_getdpi, [134](#)  
of\_getdwinputmode, [108](#)  
of\_getdwmousemovemode, [108](#)  
of\_getfiledata (Barcode object), [125](#)  
of\_getfiledata (Camera object), [127](#)  
of\_getfreememory, [135](#)  
of\_gethttpinfo function, [6](#)  
of\_getiehandle function, [7](#)  
of\_getieurl function, [7](#)  
of\_getignoredwords, [191](#)  
of\_getlanguages, [192](#)  
of\_getmaxblockcount, [166](#)  
of\_getmaxsectorcount, [167](#)  
of\_getmisspelledword, [193](#)  
of\_getname, [136](#)  
of\_getndefrecord, [167](#)  
of\_getndefrecordcount, [168](#)  
of\_getorientation, [136](#)  
of\_getostype function, [8](#)  
of\_getosversion, [138](#)  
of\_getplatform, [137](#)  
of\_getppi, [135](#)  
of\_getresolution, [138](#)  
of\_getrunmode function, [8](#)  
of\_getservertype function, [8](#)  
of\_getsessioncount function, [9](#)  
of\_getsize, [168](#)  
of\_getstatus, [100](#)  
of\_getstatusbarvisible, [139](#)  
of\_gettagid, [169](#)  
of\_gettimeout, [169](#)  
of\_gettitlebarvisible, [109](#)  
of\_gettotalmemory, [135](#)  
of\_geturlschemeparm, [109](#)  
of\_getversion, [110](#)  
of\_getwindowlisticonvisible, [110](#)  
of\_getworkspaceheight, [111](#)  
of\_getworkspaceidappname, [111](#)  
of\_getworkspacewidth, [112](#)  
of\_goback, [200](#)  
of\_goforward, [200](#)  
of\_guessesforword, [193](#)  
of\_haslearnedword, [194](#)  
of\_init, [186](#)  
of\_is64browser function, [9](#)  
of\_isenabled, [144](#)  
of\_ldaplogon function, [10](#)  
of\_learnword, [195](#)  
of\_loadhtmlstring, [201](#)  
of\_loadlocalfile, [202](#)  
of\_log, [10](#)  
of\_log (Appeon Workspace), [112](#)  
of\_logofffileserver, [39](#)

- of\_logonfileserver, [38](#)
- of\_networktest, [11](#)
- of\_open (Geolocation object), [145](#)
- of\_open (Map object), [157](#)
- of\_open (NFC object), [170](#)
- of\_openalbums, [128](#)
- of\_pauseorresume, [100](#)
- of\_play (Audio object), [101](#)
- of\_play (Media object), [162](#)
- of\_popupmenu function, [11](#)
- of\_popupmenuon function, [12](#)
- of\_print2file function, [13](#)
- of\_Print2PDF, [14](#)
- of\_read, [126](#)
- of\_readblock, [170](#)
- of\_refresh, [158](#)
- of\_register, [113](#)
- of\_register (Audio object), [101](#)
- of\_register (Geolocation object), [146](#)
- of\_register (NFC object), [171](#)
- of\_register (Payment object), [187](#)
- of\_reload, [203](#)
- of\_removeallannotation, [159](#)
- of\_removeannotation, [159](#)
- of\_removemessage, [179](#)
- of\_runjavascript, [203](#)
- of\_selectannotation, [160](#)
- of\_sendmail, [113](#)
- of\_setapporientation, [114](#)
- of\_setappplatform, [188](#)
- of\_setapprotationlock, [115](#)
- of\_setassistivetouchbtnvisible, [116](#)
- of\_setassistivetouchmode, [116](#)
- of\_setcloseappiconvisible, [117](#)
- of\_setdwinputmode, [118](#)
- of\_setdwmousemovemode, [119](#)
- of\_sethinttext, [119](#)
- of\_sethinttextcolor, [119](#)
- of\_setignoredwords, [192](#)
- of\_setkey, [171](#)
- of\_setlongpressinterval, [120](#)
- of\_setstatusbarvisible, [139](#)
- of\_settimeout, [172](#)
- of\_settitlebarvisible, [121](#)
- of\_setwindowlisticonvisible, [121](#)
- of\_startblockoperate, [173](#)
- of\_startloading, [204](#)
- of\_stop, [102](#)
- of\_stopblockoperate, [173](#)
- of\_stopuploading, [205](#)
- of\_submit, [188](#)
- of\_switchRealTimeCalc function, [14](#)
- of\_takefile, [129](#)
- of\_triggerevent, [122](#)
- of\_triggerevent (Audio object), [102](#)
- of\_triggerevent (NFC object), [173](#)
- of\_triggerevent (Payment object), [189](#)
- of\_unlearnword, [195](#)
- of\_writeblock, [174](#)
- of\_writerecords, [175](#)
- Online Payment
  - code example, [190](#)
  - eon\_mobile\_str\_paymentinit, [189](#)
  - eon\_mobile\_str\_paymentssubmit, [189](#)
  - ii\_appplatformtype, [184](#)
  - ii\_errorcode, [184](#)
  - ipo\_bindevent, [184](#)
  - is\_bindwitherrorevent, [184](#)
  - is\_errortext, [184](#)
  - is\_paymentcancel, [184](#)
  - is\_paymentfailed, [184](#)
  - is\_paymentok, [185](#)
  - oe\_error, [185](#)
  - oe\_paymentcancel, [185](#)
  - oe\_paymentfailed, [185](#)
  - oe\_paymentok, [185](#)
  - of\_getappplatform, [186](#)
  - of\_init, [186](#)
  - of\_register, [187](#)
  - of\_setappplatform, [188](#)
  - of\_submit, [188](#)
  - of\_triggerevent, [189](#)
- Other event, [299](#)

## P

- Passing arguments to functions and events, [295](#)
- Passing Menu object as a reference parameter, [295](#)
- PopulateError function, [294](#)
- PowerScript Events, [299](#)
- PowerScript Reference, [295](#)
- PowerScript Statements, [298](#)
- PowerScript Topics, [295](#)
- PowerServer open interfaces, [263](#), [263](#)
- Properties for AppeonDotNetComponent, [65](#)

Properties for AppeonFileService object, [43](#)  
Properties for appeonwebservicecomponent, [69](#)

Properties of the DataWindow Control, [305](#)

## R

Reference parameter, [296](#)

Registering parameter functions, [61](#)

Registry function, [294](#)

RegStruct and RegStructArray functions, [61](#)

ReselectRow function, [302](#)

ResetTransObject function, [302](#)

Retrieve.AsNeeded, [300](#)

RetrieveRow event, [305](#)

## S

Scroll function, [303](#)

ScrollVertical event, [303](#)

SetDropHighlight function, [290](#)

SetTrans function, [303](#)

Shared Object functions, [294](#)

Shared variables, [296](#)

SignalError function, [295](#)

SQL Statements, [299](#)

Stored procedure with Null output, [299](#)

System Controls, [290](#)

System Functions, [291](#)

System Objects, [289](#)

## T

Tab control, [290](#)

TabPostEvent function, [290](#)

Textchecker

  code example, [196](#)

  eon\_mobile\_str\_textcheckoption, [196](#)

  of\_completionsforpartialword, [190](#)

  of\_getignoredwords, [191](#)

  of\_getlanguages, [192](#)

  of\_getmisspelledword, [193](#)

  of\_guessesforword, [193](#)

  of\_haslearnedword, [194](#)

  of\_learnword, [195](#)

  of\_setignoredwords, [192](#)

  of\_unlearnword, [195](#)

TreeView control, [290](#)

## U

Uploading and downloading files, [37](#)

UserObject object, [289](#)

Using Internet Explorer Frame, [318](#)

## W

WebView

  ii\_errorcode, [197](#)

  is\_errortext, [197](#)

  oe\_error, [197](#)

  oe\_jcallbackappeon, [198](#)

  oe\_loadend, [199](#)

  oe\_loadstart, [199](#)

  of\_cangoback, [199](#)

  of\_cangoforward, [200](#)

  of\_goback, [200](#)

  of\_goforward, [200](#)

  of\_loadhtmlstring, [201](#)

  of\_loadlocalfile, [202](#)

  of\_reload, [203](#)

  of\_runjavascript, [203](#)

  of\_startloading, [204](#)

  of\_stoploading, [205](#)

Workarounds for Unsupported Features, [289](#)