Contents

1 About This Book ...................................................................................................................... 1
  1.1 Audience .............................................................................................................................. 1
  1.2 How to use this book ............................................................................................................ 1
  1.3 Related documents .............................................................................................................. 2
  1.4 If you need help .................................................................................................................... 3

2 Introduction .......................................................................................................................... 4
  2.1 Overview .............................................................................................................................. 4
  2.2 Appeon Developer toolbar in PowerBuilder ........................................................................ 4
  2.3 Web application development with Appeon Developer ...................................................... 6

3 Before You Begin .................................................................................................................. 7
  3.1 Installing Appeon for PowerBuilder .................................................................................. 7
  3.2 Configuring application database connectivity ................................................................ 7
  3.3 Starting Appeon Server ...................................................................................................... 7

4 Configuring Appeon Developer ............................................................................................ 9
  4.1 Overview ............................................................................................................................... 9
  4.2 Using Appeon Developer Configuration Wizard .............................................................. 9
    4.2.1 Configuring basic settings .............................................................................................. 10
    4.2.2 Selecting PBL file(s) .................................................................................................... 11
    4.2.3 Configuring deployment settings .................................................................................. 12
    4.2.4 Selecting DB type(s) .................................................................................................... 13
    4.2.5 Declaring transaction object(s) .................................................................................... 14
    4.2.6 Selecting image files ..................................................................................................... 17
    4.2.7 Selecting INI files ......................................................................................................... 17
    4.2.8 Selecting DLL/OCX files ............................................................................................. 18
    4.2.9 Configuring command line options ............................................................................. 19
    4.2.10 Configuring language options .................................................................................... 20
    4.2.11 Specifying log mode .................................................................................................... 21
    4.2.12 Summary ...................................................................................................................... 22
  4.3 Using Configure tool ......................................................................................................... 23
    4.3.1 Managing application profiles ....................................................................................... 23
    4.3.2 Managing database type profiles .................................................................................. 42
    4.3.3 Managing server profiles .............................................................................................. 45
    4.3.4 Managing deployment profiles ...................................................................................... 53
    4.3.5 Managing connection cache profiles ............................................................................. 55

5 Using UFA Tool ...................................................................................................................... 60
  5.1 Overview .............................................................................................................................. 60
  5.2 Analyzing an application ................................................................................................. 61
    5.2.1 Tasks required before you perform feature analysis .................................................... 61
    5.2.2 Accessing the UFA tool ............................................................................................... 62
    5.2.3 Performing feature analysis ........................................................................................ 64
  5.3 Working with UFA Report ................................................................................................ 66
    5.3.1 Modifying unsupported features .................................................................................. 66
1 About This Book

1.1 Audience
This book is written for PowerBuilder developers using Appeon Developer.

1.2 How to use this book
There are sixteen chapters in this book.

Chapter 1: About This Book
A general description of this book.

Chapter 2: Introduction
An overview of Appeon Developer.

Chapter 3: Before You Begin
Some important instructions that you MUST follow before using Appeon Developer.

Chapter 4: Configuring Appeon Developer
Instructions for configuring Appeon Developer.

Chapter 5: Using UFA Tool
Instructions for analyzing the unsupported features in a PowerBuilder application.

Chapter 6: Deploying PowerBuilder Applications
Instructions for deploying PowerBuilder applications to the Web.

Chapter 7: Debugging Appeon Web Applications
Instructions for using Appeon Debugger to debug an Appeon Web application.

Chapter 8: Running Web Applications
Instructions for running Web applications after deployment.

Chapter 9: Using Information Manager
Instructions for viewing reports and log files generated in the Appeon migration process.

Chapter 10: Packaging and Installing Web Applications
Instructions for using Web application packaging tool to generate installation packages.

Chapter 11: Undeploying Web Applications
Instructions on how to undeploy Appeon Web applications.

Chapter 12: Developing with Code Insight
Instructions on how to use the Appeon Code Insight feature.

Chapter 13: Launching Appeon Enterprise Manager
Instructions for launching Appeon Enterprise Manager from Appeon Developer.
Chapter 14: Appeon for PowerBuilder Help
Instructions for launching the Appeon HTML help file.

Chapter 15: Extended Toolkit
Instructions for using the extended tools provided by Appeon: the DLL/OCX Files Package tool, the Appeon Silent Installer, and the Appeon Developer Toolbar Register tool.

Chapter 16: Technical Support
Important information on technical support.

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

- **Appeon Demo Applications Tutorial:**
  Introduces Appeon’s demo applications, including the Appeon Sales Application Demo, Appeon Code Examples, Appeon ACF Demo, and Appeon Pet World, which show Appeon’s capability in converting PowerBuilder applications to the Web.

- **Appeon Developer User Guide (or Working with Appeon Developer Toolbar)**
  Provides instructions on how to use the Appeon Developer toolbar in Appeon 6.5. 
  *Working with Appeon Developer Toolbar* is an HTML version of the *Appeon Developer User Guide*.

- **Appeon Server Configuration Guide**
  Provides instructions on how to configure Appeon Server Status Monitor, establish connections between Appeon Server and Database Server, and configure AEM for maintaining Appeon Server and Appeon deployed Web applications.

- **Appeon Supported Features Guide (or Appeon Features Help):**
  Provides a detailed list of what PowerBuilder features are supported and can be converted to the Web with Appeon 6.5 and what features are unsupported. 
  *Appeon Features Help* is an HTML version of the *Appeon Supported Features Guide*.

- **Appeon Installation Guide:**
  Provides instructions on how to install *Appeon for PowerBuilder* successfully.

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

- **Appeon Performance Tuning Guide:**
  Provides instructions on how to modify a PowerBuilder application to achieve better performance with its corresponding Web application.

- **Appeon Troubleshooting Guide:**
Provides information about troubleshooting issues, covering topics such as product installation, Web deployment, AEM, Web application runtime, etc.

- **Introduction to Appeon:**
  Guides you through all the documents included in Appeon 6.5 for PowerBuilder.

- **New Features Guide (or What’s New in Appeon):**
  Introduces new features and changes in Appeon 6.5 for PowerBuilder.
  *What’s New in Appeon* is an HTML version of the *New Features Guide*.

**1.4 If you need help**

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support, or an Authorized Sybase Support Partner. If you have any questions about this installation, or if you need assistance during the installation process, ask a designated person to contact Sybase Technical Support, or an Authorized Sybase Support Partner based on your support contract. You may access the Technical Support Web site at [http://www.sybase.com/support](http://www.sybase.com/support).
2 Introduction

2.1 Overview

Appeon Developer, a component of Appeon for PowerBuilder, extends the capabilities of PowerBuilder, allowing a new or existing PowerBuilder application to be converted into a *bona fide* Web application using only PowerBuilder skills.

Appeon Developer provides a set of tools that enable the entire PowerBuilder-to-Web process to take place within the PowerBuilder IDE. These tools are accessed via a toolbar in the PowerBuilder IDE, which automatically loads each time PowerBuilder is opened.

**Figure 2-1: Appeon Developer toolbar embedded in the PowerBuilder IDE**

Appeon Developer converts a PowerBuilder application by parsing the PBLs (source code) of PowerBuilder applications and generating a set of corresponding HTML, JavaScript and XML files. When deployed by Appeon Server, these generated files form an exact Web-based replica of the source PowerBuilder application. Any user can open a standard Microsoft Web browser and access the Web-based version of the original PowerBuilder application over the Internet, an intranet, or an extranet.

2.2 Appeon Developer toolbar in PowerBuilder

All Appeon Developer tools conform to the US Government Section 508 Accessibility Guidelines.
Figure 2-2 represents the user interface of the Appeon Developer toolbar.

**Figure 2-2: Appeon Developer toolbar**

The Appeon Developer toolbar has 13 buttons. The functionality of each button is described in Table 2-1.

**Table 2-1: Appeon Developer toolbar functions**

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>![icon]</td>
<td>Config Wizard</td>
<td>Provides a wizard for quick configuration of Appeon Developer for Web conversion.</td>
</tr>
<tr>
<td>![icon]</td>
<td>Configure</td>
<td>Provides configuration and management of Appeon Developer for Web conversion.</td>
</tr>
<tr>
<td>![icon] or ![icon]</td>
<td>Analyze</td>
<td>Analyzes application source code for unsupported features before deployment. This analysis can cover the whole application, at object level or within an inheritance hierarchy. (It displays as ![icon] in PowerBuilder 11.5)</td>
</tr>
<tr>
<td>![icon]</td>
<td>Code Insight</td>
<td>Develops PowerBuilder applications that are free of Appeon unsupported features.</td>
</tr>
<tr>
<td>![icon]</td>
<td>Deploy</td>
<td>Starts the complete process of deploying a PowerBuilder application to the Web.</td>
</tr>
<tr>
<td>![icon]</td>
<td>Run</td>
<td>Runs the deployed Web application in Internet Explorer.</td>
</tr>
<tr>
<td>![icon]</td>
<td>Appeon Debugger</td>
<td>Debugs the Appeon Web application.</td>
</tr>
<tr>
<td>![icon]</td>
<td>Information</td>
<td>Manages logs and reports.</td>
</tr>
<tr>
<td>![icon]</td>
<td>Package</td>
<td>Packages application files for installation.</td>
</tr>
<tr>
<td>![icon]</td>
<td>Undeploy</td>
<td>Undeploys Web applications from Web Server(s) and Appeon Server(s).</td>
</tr>
<tr>
<td>![icon]</td>
<td>AEM</td>
<td>Launches Appeon Enterprise Manager (AEM).</td>
</tr>
<tr>
<td>![icon]</td>
<td>Help</td>
<td>Searches, browses, prints, copies and pastes useful information in the Appeon user guides.</td>
</tr>
<tr>
<td>![icon]</td>
<td>Get Support</td>
<td>Displays the product version and the support information. The Check Update button in it will allow you to check the Appeon updates.</td>
</tr>
</tbody>
</table>

Apart from Code Insight and Information, only one Appeon Developer toolbar function can be launched at any given time. This prevents an Appeon Developer process from interfering with another.
2.3 Web application development with Appeon Developer

For either developing a new Web application, or migrating an existing PowerBuilder application onto the Web, you need to perform three key tasks in Appeon Developer: Analyze, Modify and Deploy Automatically.

If you use Appeon for PowerBuilder for developing a new Web application, the first step is to write a new PowerBuilder application (refer to Chapter 2: Web RAD with Appeon for PowerBuilder in the Appeon Migration Guide). It is recommended that the new PowerBuilder application conforms to Appeon’s coding styles, as laid out in Appeon Features Help, then you will expend less effort in the first two tasks: Analyze and Modify.

- **Task 1 – Analyze**
  You must analyze the application PBLs for unsupported PowerBuilder objects and features, using the PowerBuilder IDE extended with Appeon Developer. General rules of thumb regarding how the application should be structured are outlined in Appeon Features Help under Basic and Architectural Requirements. The application must meet these requirements before the analysis can begin.

  The first step results in an analysis report highlighting unsupported PowerBuilder objects and code within the application.

- **Task 2 – Modify**
  Next, work around or remove unsupported PowerBuilder objects and code that prevent the Web application from running, using standard PowerBuilder programming. Appeon Features Help and Appeon Code Examples (an Appeon demo PowerBuilder application) will guide you through this process by providing information regarding supported PowerBuilder features as well as example code for implementing these features.

  Then, perform a “full build” of the application in PowerBuilder, ensure that there are no bugs in the PowerBuilder code and that the application functions correctly.

  The second task results in a PowerBuilder application that is ready for automatic conversion to the Web.

- **Task 3 – Deploy Automatically**
  At the push of a button, you can now automatically generate a precise replica of the PowerBuilder application, that deploys to the Web n-Tier architecture and can be accessed by standard Internet Explorer Web browsers.

  Lastly, deploy the generated Web application files to Appeon Server. The third task results in a *bona fide* Web application with the look and feel of the source PowerBuilder application.

For detailed information about Web conversion, refer to the Appeon Migration Guide.
3 Before You Begin

3.1 Installing Appeon for PowerBuilder

Follow the Appeon Installation Guide carefully; make sure that both Appeon Developer and Appeon Server have been installed.

Verify that Appeon Developer has been installed to the Developer PC.

Verify that Appeon Server has been installed to the application server, such as, Sybase EAServer, Oracle WebLogic, IBM WebSphere, or JBoss. Once Appeon Server has been installed, the machine hosting Appeon Server is then referred to as Appeon Server.

If using a separate Web Server (e.g. Apache) instead of the application server built-in Web server, verify that the Appeon Server Web Component has been installed to the Web Server.

Verify that all system requirements have been met for the Developer PC, Web Server, Appeon Server, and the Database Server. Refer to the Appeon Installation Guide for details.

3.2 Configuring application database connectivity

Perform the following database configurations for all prospective PowerBuilder applications that will be deployed with Appeon Developer:

1. Set up the database used by the PowerBuilder application to interface with Appeon Server and the JDBC driver. Refer to the documentation from the appropriate database vendor for instructions.

2. Create the connection cache in Appeon Server. Appeon provides systematic instructions for certified database systems. Refer to the Appeon Server Configuration Guide for more information.

3. Map the transaction object in the target PowerBuilder application to the newly created connection cache for the prospective Web application. You can either dynamically set up the mapping via PowerScript following the instructions in the Appeon Server Configuration Guide or establish the mapping statically in Appeon Developer or AEM.

3.3 Starting Appeon Server

Appeon Developer interacts with Appeon Server and Web server during Web deployment. Web applications are deployed to one or more Appeon Servers and one or more Web servers. You should verify that the Appeon Server and Web server are running before deployment. If you use the application server built-in Web server as the Web server, you only need to start application server/Appeon Server. For example, if you use EAServer as the Appeon Server and EAServer HTTP Server as the Web server, you only need to start EAServer/Appeon Server. Otherwise, start both the Web server and Appeon Server. Refer to the documents provided by the Web server vendor for how to start the Web server, and refer to the following instructions to start the Appeon Server.
Table 3-1: How to start Appeon Server

<table>
<thead>
<tr>
<th>Appeon Server</th>
<th>Operating System</th>
<th>Detailed Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAServer</td>
<td>Windows</td>
<td>Select Windows Start</td>
</tr>
<tr>
<td></td>
<td>UNIX/Linux</td>
<td>Run the appeonserverstart.sh file in the $appeon/bin/ folder, for example: $ $JAGUAR/appeon/bin/appeonserverstart.sh. You can run EAServer/Appeon Server in different modes, debug and normal, using different Java runtime versions and different Java VMs. For details on how to specify the mode options in the above syntax, refer to the EAServer System Administration Guide.</td>
</tr>
<tr>
<td>WebLogic</td>
<td>Windows</td>
<td>Select Windows Start</td>
</tr>
<tr>
<td></td>
<td>UNIX/Linux</td>
<td>Run the appeonserver.sh file in the $appeon/bin/ folder, for example: $ $BEA_HOME/user_projects/domains/mydomain/appeon/bin/appeonserver.sh.</td>
</tr>
<tr>
<td>WebSphere</td>
<td>Windows</td>
<td>Select Windows Start</td>
</tr>
<tr>
<td></td>
<td>UNIX/Linux</td>
<td>Run the appeonserver.sh file in the $appeon/bin/ folder, for example: $ $WebSphere/AppServer/appeon/bin/appeonserver.sh.</td>
</tr>
<tr>
<td>JBoss</td>
<td>Windows</td>
<td>Select Start</td>
</tr>
<tr>
<td></td>
<td>UNIX/Linux</td>
<td>Run the run.sh file in the $JBoss/bin/ folder.</td>
</tr>
</tbody>
</table>
4 Configuring Appeon Developer

4.1 Overview

The settings configured in Appeon Developer are critical; they are used throughout the entire PowerBuilder-to-Web process. Appeon Developer settings determine which PowerBuilder application will be converted to the Web, and the manner in which it will be deployed.

Before you attempt to use any other functionality on the toolbar, complete the following tasks:

1. Set up an application profile for each of the PowerBuilder applications intended for conversion. Each application profile tells Appeon Developer important information about the application, such as which PBLs compose the PowerBuilder application, the database type, etc.

   During the application profile setup, set up a profile for the database type used by the application. This enables Appeon Developer to generate the correct database syntax.

2. Set up at least one Appeon Server profile and one Web Server profile. This enables Appeon Developer to utilize the Appeon Server and Web Server for deployment.

3. Set up at least one deployment profile, which links at least one Appeon Server and one Web Server together. This tells Appeon Developer where to deploy the Web application.

   Appeon provides a Config Wizard to quickly configure an application. With your specifications, the wizard creates the application profile, database type profile, Appeon Server profile, Web server profile, connection cache profile, and transaction object mapping. After creation, all these profiles can be modified in the Configure tool. Besides modifying the above profile settings, the Configure tool provides you with advanced settings, such as performance settings, parsing options, Web Services profiles, runtime settings etc.

4.2 Using Appeon Developer Configuration Wizard

The settings in Appeon Developer Configuration Wizard are the same as those in the Configure tool, therefore, only basic description will be provided in this section. For detailed description, refer to Section 4.3: Using Configure tool.

   Click the Config Wizard button on the Appeon Developer toolbar to launch the Appeon Developer Configuration Wizard.

   Carefully read the notes and requirements on the Welcome page of the Appeon Developer Configuration wizard (Figure 4-1). Click Next to proceed.
4.2.1 Configuring basic settings

The basic settings refer to the settings of an application that are essential for deployment, including the application profile name, Web URL, and file generation mode, as shown in Figure 4-2. For more information about them, refer to the Basic Settings section in the Using Configure tool section.
Figure 4-2: Configure basic settings of an application

Step 1 – Specify the application profile name.

The application profile name is used to identify the PowerBuilder application during the entire PowerBuilder-to-Web process.

Step 2 – Specify the application Web URL.

The specified text is also used as the name of the folder created under the Web root of the Web server for storing the Web application files when the application is deployed.

Step 3 – Choose a Web file generation mode.

4.2.2 Selecting PBL file(s)

Specify the version and the location of the PowerBuilder application source code (Figure 4-3).
Figure 4-3: Select PBLs

You can add application PBLs using one of the following methods:

Method 1 – Select the workspace and the PBT file to automatically add PBL files. Detailed steps are as follows:

- **Step 1** – Select the workspace. Select a recent workspace from the dropdown list, or click the ellipsis button to select a workspace.
  
  When a workspace is selected, all PBT files contained in the selected workspace are added to the PBT List.

- **Step 2** – Select the PBT file.
  
  When a PBT file is selected, all PBL files contained in the selected PBT are added to the PBL File List.

This is the recommended method to add PBLs, as it reduces the likelihood of forgetting to add a required PBL.

Method 2: You can also add PBL files without specifying the workspace and PBT file first.

- **Click Add File** to add one or multiple PBL files. This is the only method you can take when using the **Configure** tool. Refer to the **Basic Settings** section for detailed instructions.

### 4.2.3 Configuring deployment settings

The deployment settings associate the Appeon Server(s) and Web server(s) as a group used for application deployment (Figure 4-4).
Step 1 – Select an existing deployment profile.

To create or modify a deployment profile, you must use the Configure tool. For detailed instructions, refer to Section 4.3.4: Managing deployment profiles.

Step 2 – Select the check boxes to include the Appeon Server(s) and the Web server(s) in the deployment profile.

All the Appeon Server profiles and Web server profiles you have created are listed. More than one Appeon Server and one Web server can be selected.

If the required Appeon Server profiles or Web server profiles are not available, click the Add button to create them. Refer to Section 4.3.3.b: Appeon Server profile settings for detailed instructions on creating an Appeon Server profile and refer to Section 4.3.3.c: Web Server profile settings for detailed instructions on creating a Web server profile.

Step 3 – Click Next to proceed.

When Next is clicked, the wizard will test the connection to the Appeon Server and the Web server. If connection is successful, the deployment profile is saved with the associated Appeon Server(s) and Web server(s).

4.2.4 Selecting DB type(s)

The database type is required during the application deployment if the application connects to one or more databases.

Select the database type(s) used by the application by selecting the Used column of the database type, as shown in Figure 4-5.
If the database type is not configured (The Configured column is indicated with “No”), you should select it and click the Edit button to create a profile for it. For detailed instructions on creating a database type profile, refer to Section 4.3.2: Managing database type profiles.

4.2.5 Declaring transaction object(s)

When the application is deployed to the Web, Appeon Server handles the database connection using connection caches rather than transaction objects defined in the PowerBuilder application. You must associate the transaction objects used by the PowerBuilder application with proper connection caches created in Appeon Server.

You can create, modify or delete the mapping between transaction objects and connection caches either in Appeon Developer or AEM.
Figure 4-6: Specify transaction objects

Click *Add* to create the mapping of the transaction object and the connection cache of an application, as shown in Figure 4-7.
Figure 4-7: Add a transaction object

Step 1 – Input the transaction object name used by the application to the Transaction Object text box.

Step 2 – Select the database type from the Database Type list box.

Step 3 – Specify the connection cache name in the Connection Cache text box. The connection cache should connect to the same database that the transaction object connects to. Make sure the connection cache exists in all Appeon Servers used in the deployment. The cache name is case sensitive.

Note: The Connection Cache tool will be available if Appeon Server is installed to EAServer 5.x, and not be available if Appeon Server is installed to EAServer 6.x, WebLogic, WebSphere, or JBoss. You should go to the corresponding application server administration console to configure the data source for EAServer 6.x, WebLogic, WebSphere, or JBoss. For detailed instructions, refer to the Database Connection Setup chapter in the Appeon Server Configuration Guide.
4.2.6 Selecting image files

Click Browse to specify which folder contains the image files that will be used in the Web application. Select Include Subfolders to deploy the sub-folders under the specified directory (Figure 4-8).

For detailed information about the image files, refer to the Deploy Images section.

Figure 4-8: Select image files

4.2.7 Selecting INI files

If a PowerBuilder application uses an initialization (INI) file which contains user preferences, specify the INI file so that Appeon can deploy the file for the application.

Click Add to add one or multiple INI files or click Add Directory to add all INI files in the selected directory (Figure 4-9).

For detailed information about the INI files, refer to the Deploy INI Files section.
4.2.8 Selecting DLL/OCX files

If your application calls any custom user DLL/OCX files, you can specify them here and deploy them to the Web server. Or you can use the DLL/OCX File Packaging tool provided by Appeon to package the files and install them to the client. For detailed instructions, refer to Chapter 15: Extended Toolkit.

Click Add File or Add Directory to add DLL/OCX files to the table. Then select the Auto Register column if the file needs to be automatically registered after it is downloaded to the client (Figure 4-10). If you choose not to automatically register a file, you can register it manually after it is downloaded.

For more information about the DLL/OCX files, refer to the Deploy external non-System DLL/OCX files section.
Figure 4-10: Select DLL/OCX files

4.2.9 Configuring command line options

If an application needs to use command line arguments, enable the command line argument option here, as shown in Figure 4-11.

Step 1 – Click the Require a command line argument for the application check box to enable the command line argument option.

Step 2 – Select the Show the argument input/confirmation dialog option and/or input the default arguments in the Default command argument text box.

Note: They are effective only if you run the application from the Appeon Developer toolbar. For more information, refer to the Command line arguments section.
4.2.10 Configuring language options

Select a language from the System Prompt Language dropdown listbox, as shown in Figure 4-12.

System prompts are error messages, informational messages, warnings, notices, and prompts that are displayed when you run Appeon Web applications.
4.2.11 Specifying log mode

The log-writing mode determines the content of the log files generated by Appeon Developer, including the deployment logs and feature analysis logs.

Select the desired mode from the Appeon Developer Log Mode box, as shown in Figure 4-13. For detailed information of log modes, refer to the *Appeon Developer log-writing mode* section.
4.2.12 Summary

Review the settings. Click Finish to complete the configuration (Figure 4-14).

If the Deploy the application now option is selected, the Appeon Deployment Wizard will be launched for you to deploy the application. For detailed information, refer to Chapter 6: Deploying PowerBuilder Applications.

After the configuration is complete, the application will be added to the Application Profiles tab of the Configure tool where you can manage or change any settings of the application. For detailed instructions, refer to Section 4.3: Using Configure tool.
4.3 Using Configure tool

The Configure tool allows you to modify the settings of existing application profiles, database type profiles, Appeon Server profiles, Web server profiles, deployment profiles and connection cache profiles, and specify the advanced settings for an application profile, such as the performance settings, parsing options, Web service profiles, and runtime settings.

4.3.1 Managing application profiles

The Configure tool provides an Application Profiles tab for you to modify the settings specified in the Appeon Developer Configuration Wizard, and use the following advanced functions:

- Specify the default application profile
- Select the Web file generation mode for an application
- Delete an application profile
- Boost the application runtime performance
- Specify parsing options
- Configure Web Service profiles
- Specify runtime Settings
4.3.1.a Application Profiles tab page

When you click the *Configure* button (◫) on the Appeon Developer toolbar, the Appeon Developer Configuration window pops up with the *Application Profiles* tab selected by default, as shown in Figure 4-15.

Figure 4-15: Appeon Developer Configuration Window

![Appeon Developer Configuration Window](image)

Specifying the default application profile

To specify which application profile will be used as the default application profile, select the *Default* radio button.

The PowerBuilder application defined in the default application profile is selected for unsupported features analysis, PowerBuilder-to-Web conversion, application packaging, and application undeployment.

Selecting a Web file generation mode

To specify a mode for Web file generation, click the *Mode* list box. If you change the mode, you must perform a full deployment to make the changes effective.

Table 4-1: File generation mode

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
<th>Use In The…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debug PS/JS</td>
<td>Generates unencrypted JavaScript files for debug use. The PowerBuilder source code is provided as comments in the JavaScript files for easy reference.</td>
<td>Web application debugging and tuning stage.</td>
</tr>
<tr>
<td>Debug JS</td>
<td>Generates unencrypted JavaScript files for debug use. No PowerBuilder source code is provided in the</td>
<td>Web application debugging and tuning</td>
</tr>
</tbody>
</table>
### Deleting an application profile

Click the *Delete* button to delete a selected application profile.

**Note:** You cannot delete the default application profile.

After the application profile is deleted, delete the temporary folder for the application profile on the Developer machine. The folder has the same name as the application profile and is created in the `\Project folder under the root directory where Appeon Developer is installed (e.g. C:\Program Files\Appeon\Developer6.5`).

### Exporting/Importing an application profile

The Export and Import buttons are mainly to back up and restore the configuration of application profile(s), for instance, if you want to transfer the configuration of application profile(s) to another Appeon Developer.

- Click the Export... button to export the selected application profile(s). In the pop-up dialog box, type a file name, then the selected application profile(s) will be saved into the file as XML file format.

  To select multiple application profiles, you can use Shift+click or Ctrl+click, or drag the mouse pointer to create a selection.

- Click the Import… button to import application profile(s) from the XML file which is generated by using the Export… button.

  If the application profile being imported has the same name as an existing application profile, you will be asked to overwrite or ignore the existing application profile.

### 4.3.1.b Application profile settings

The *Application Profiles Configuration* window provides seven tab pages for you to edit the required application information:

<table>
<thead>
<tr>
<th>Tab</th>
<th>Settings</th>
<th>To make changes effective, you must …</th>
</tr>
</thead>
</table>
| Basic Settings tab | The application profile name, Web URL, PBL version and PBL location of an application.  
See the *Basic Settings* section for more details. | Perform a full or incremental deployment on the application. |
| DB Settings tab   | The database types, transaction objects and the corresponding connection caches.  
See the *DB Settings* section for more details.       | Perform a full or incremental deployment on the application. |
| Additional Files tab | INI files, .NET/COM components, image files and DLL/OCX files.  
See the *Additional Files* section for more details. | Perform a full or incremental deployment on the application. |
<table>
<thead>
<tr>
<th>Misc Settings tab</th>
<th>Command line arguments, application language, runtime performance, parsing option, and log-writing mode. See the Misc Settings section for more details.</th>
<th>Perform a full deployment on the application.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Service Profiles tab</td>
<td>The WSDL file, service name and port for the Web service(s). See the Web Service Profiles section for more details.</td>
<td>Perform a full or incremental deployment on the application.</td>
</tr>
<tr>
<td>Runtime Settings tab</td>
<td>Company name and application name displayed in the Web application downloading page. See the Runtime Settings section for more details.</td>
<td>Perform a full or incremental deployment on the application.</td>
</tr>
<tr>
<td>Summary</td>
<td>A summary view of the settings. See the Summary section for more details.</td>
<td>Perform a full deployment on the application.</td>
</tr>
</tbody>
</table>

**Basic Settings**

After you click the Add or Edit button on the Application Profiles tab, the Application Profile Configuration window is displayed with the Basic Settings tab selected by default, as shown in Figure 4-16.

The Basic Settings tab allows you to modify the Web URL, application PBL version and PBL location.
Figure 4-16: Basic Settings

![Application Profile Configuration](image)

<table>
<thead>
<tr>
<th>Basic Settings</th>
<th>Runtime Settings</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Profile Name:</strong></td>
<td>sales_application_demo</td>
<td></td>
</tr>
<tr>
<td><strong>Web URL</strong></td>
<td><a href="http://web">http://web</a> server:port</td>
<td>sales_application_demo</td>
</tr>
</tbody>
</table>

**Application Profile Name** – The application profile name is used to identify the PowerBuilder application during the entire PowerBuilder-to-Web process. It is also used in the appeondb database on Appeon Server to identify the DataWindow syntax, profile, and registry information for an Appeon Web application.

The Profile Name cannot contain double-byte characters (such as Chinese, Korean, or Japanese characters) or special characters (such as, \, /, :, *, ?, " , <, >, or |).

**Web URL** – A URL of the Appeon Web application can contain a combination of letters, underscores (“_”), and numbers. The specified URL is also used as the name of the folder created under the Web Root of the Web Server for storing the Web application files when the application is deployed.

Avoid setting application Web URLs as existing folder names in the Web server Web root because: (1) If a Web application uses an existing folder under the Web root as the Web URL, a large number of Web files belonging to the application will be deployed to that folder, making it difficult to use the folder for its original purpose. (2) If a Web application uses a folder that is reserved for the Web server or Appeon Server, some important files belonging to the Web server or Appeon Server may be replaced or even removed when the application is undeployed. The following are folder names reserved for the Web server or Appeon Server that should not be used as application Web URLs: appeon, classes, docs, images, ir, wst, and WEB-INF.
The Web URL for an application profile can be changed, and the Web files stored at the Web Server can have different versions for the Client to access, but Appeon only remembers the Web URL in the last deployment, and performs the application packaging and application undeployment based on the Web files generated in the corresponding Web folder.

**Application PBL files** – Select the PBL version and add application PBL files using any of the following methods:

- Click the *Add* button and choose to add an application Target file (*.pbt).
  By adding PBLs using an application Target, all PBLs in the Target are automatically added to the application PBLs list. This is the recommended way to add PBLs, as it reduces the likelihood of forgetting to add a required PBL.

- Click the *Add* button and add PBL files without using an application Target.
  You can add one or multiple PBL files at the same time. To select multiple files, hold down the Ctrl key and choose which files to add.

- Copy and paste (Ctrl+C and Ctrl+V) the path list of PBLs into the PBL list box.

**DB Settings**

The *DB Settings* tab allows you to specify the database types and the transaction objects that the application uses, as shown in Figure 4-17. When configuring the database types, it is not necessary to specify the actual databases that the application uses, it can be any database of the same type. However, when specifying the transaction objects, you must specify the actual transaction objects that the application uses.

*What the database type is used for*

PowerBuilder SQL statements must be converted into the correct database syntax before they can be used to retrieve data from the database that the Web application connects to. The database type settings enable Appeon Developer to apply the correct database driver type for generating correct database syntax for PowerBuilder SQL statements.

*Set the database types for the application*

Select the database type(s) used by the application, as shown in Figure 4-17.
Figure 4-17: Database types

Table 4-3: Database type settings

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used</td>
<td>You can click the checkboxes to select the database types used by the application. More than one database type can be selected, but only the configured database types are selectable.</td>
</tr>
<tr>
<td>Supported Database Type</td>
<td>This column lists all the database types that are supported by Appeon.</td>
</tr>
<tr>
<td>Configured</td>
<td>This column is identical to the Configured column in the DB Type Profiles tab. It indicates whether a profile has been configured for the database type. If this column of the desired database type is “No”, you must click the Configure button to configure a profile for it before you can use it. Configuring a database type profile in this tab is the same as configuring it in the DB Type Profiles tab. Once a database type profile is configured successfully, the Configured column in both this tab and the DB Type Profiles tab changes from “No” to “Yes”.</td>
</tr>
</tbody>
</table>

Specify the transaction object and the connection cache

Click Add or Edit to configure a transaction object and specify the corresponding connection cache. Refer to Section 4.2.5: Declaring transaction object(s) for detailed description.
Additional Files

“Additional files” refer to the files that are outside the application source code and are necessary for running the PowerBuilder application, including INI files, .NET/COM components, image files, and DLL/OCX files.

Figure 4-18: Additional Files tab

Deploy INI Files

If a PowerBuilder application uses an initialization (INI) file which contains user preferences, specify the INI file in this tab so that Appeon can deploy the file for the application.

Click Add to add the INI file(s) into the Deploy INI Files list box.

How INI files are supported in Appeon

During deployment, the specified INI file(s) are converted to XML file(s) and deployed to Appeon Server. The XML file(s) act as mock INI file(s). When a Client runs the deployed application for the first time, a copy of the XML file is created in Appeon Server and keeps the Client’s profile information.

For a Client to use the correct profile information in its subsequent visits to the application, the Client browser must be cookie-enabled. Otherwise, each time the Client visits the application, a new copy of the XML file is created in Appeon Server, and the new copy only contains the information of the original INI file.
You can delete XML files that are unused for a period using Appeon Enterprise Manager (AEM). For more information on how to maintain XML files in Appeon Server, refer to the Auto Cleanup section in the Appeon Server Configuration Guide.

*Deploy .NET/COM components*

If a PowerBuilder application uses .NET/COM components which contains executable files (.exe), and COM/COM+ components, specify the .NET/COM components file in this tab so that Appeon can deploy the file for the application.

Click *Add* to add the .NET/COM components file(s) into the Deploy .NET/COM components list box.

*Deploy Images*

Click *Browse* to specify which folder contains the image files that will be used in the Web application. If you do not specify the path of the image files, the path to the application PBL files will be displayed as the default path of the image files.

The image files can be in any format supported by PowerBuilder, apart from:

- unsupported run-length encoded (RLE), and
- unsupported Windows metafile (WMF)

All the image files in the specified folder will be deployed to the `%Web Deployment Path%\images` folder in the Web Server. `%Web Deployment Path%` stands for the folder storing the Web application files. The image files may fail to be displayed on the Web if the file names are stored in the database or dynamically generated. To resolve this, you can copy the image files directly to the `%Web Deployment Path%\images` folder, but make sure that the file names are all lowercase; otherwise, the Web application may fail to load them.

*Deploy external non-System DLL/OCX files*

If your application calls any custom user DLL/OCX files, which are also called external non-system DLL/OCX files in this tab, you can specify them here and deploy them to the Web server.

If there are many large-size DLL/OCX files files in your application and they could be slow to download from the Web server at runtime, you may prefer to use the DLL/OCX File Packaging tool provided by Appeon to package the files into one setup program (see Chapter 15: `Extended Toolkit` for detailed instructions), then distribute the setup program to the users, and have users install the files to each Client.

Click *Options* in the *Deploy DLL/OCX Files* group box to open the Deploy DLL/OCX Files dialog box (as shown in Figure 4-19) and select the required DLL/OCX files to be deployed.
Figure 4-19: DLL/OCX Automatically Deploy and Download

Table 4-4 introduces the elements on the Deploy DLL/OCX Files dialog box.

Table 4-4: Deploy DLL/OCX Files dialog box

<table>
<thead>
<tr>
<th>column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register</td>
<td>Allows you to select whether the file will be automatically registered after it is downloaded to the Client at runtime. Click Select All to select all files and Unselect All to de-select all of them. If you choose not to automatically register a file, you can register it manually after it is downloaded.</td>
</tr>
<tr>
<td>File Name</td>
<td>Lists the added files. Click the header of this column to display the files in alphabetical order.</td>
</tr>
<tr>
<td>File Path</td>
<td>Displays the file location.</td>
</tr>
</tbody>
</table>

Table 4-5 describes how to use the Deploy DLL/OCX Files dialog box.

Table 4-5: Deploy DLL/OCX Files dialog box

<table>
<thead>
<tr>
<th>To Do This</th>
<th>Do This</th>
</tr>
</thead>
<tbody>
<tr>
<td>To add files</td>
<td>Click the Add button. A standard File Selection dialog box is displayed. Select the files from your local machine. The files can be stored in any location.</td>
</tr>
<tr>
<td>To remove files</td>
<td>Select a file or multiple files by pressing Ctrl or Shift keys, and click the Delete button to remove the selected files from the list.</td>
</tr>
<tr>
<td>To save the settings</td>
<td>Click OK. Appeon will copy the files to a temporary folder and then deploy them to the Web server during deployment.</td>
</tr>
</tbody>
</table>

How the deployed DLL/OCX files work on Web
To successfully call the deployed DLL/OCX files on the Web application, the DLL/OCX files must be downloaded and installed to the Client during the initial run of the deployed Web application.

Appeon Enterprise Manager (AEM) provides options for manually or automatically downloading and installing the DLL/OCX files, and for specifying the installation location (by default the files are saved to the %SystemRoot%\system32\AppeonPlugin folder at the Client. %SystemRoot% indicates the folder where Windows is installed, typically C:\Windows). For detailed instructions, refer to the Appeon Server Configuration Guide.

**Misc Settings**

The Misc Settings tab page enables you to modify settings for command line arguments, application language, and log-writing mode, and allows you to configure the runtime performance, and parsing options.

**Figure 4-20: Misc settings**

![Application Profile Configuration](image)

- **Command Line Arguments**
  Configure how command line arguments used in the PowerBuilder application will be handled on the Web.

- **Language Options**
  Configure options for the application language.

- **Performance**
  Configure additional settings to tweak the application’s run-time performance.

- **Parsing Options**
  - Allow Deployment Wizard to rename the identifiers unsupported by JavaScript or XML.

- **Appeon Developer Logs**
  Set the log-writing mode for Deploy Log and Analysis Log. You may view/manage these logs in the Information Manager.
  - Off mode
  - Summary mode
  - Debug mode

**Command line arguments**

If an application needs to use command line arguments, enable the command line argument option, as shown in Figure 4-21.

Click **Command Line Settings** on the Misc Settings tab. The Command Line Settings dialog box is displayed, as shown in Figure 4-21.
- Require a command line argument for the application option – Enables the command line argument option.

  With this check box selected, the following two options will be enabled. Note that they are effective only if you run the application from the Appeon Developer toolbar.

- Show argument input/confirmation dialog option – A dialog box is displayed after you launch an application from the Appeon Developer toolbar but before the application starts. This dialog box enables you to specify the arguments for the application.

- Specify the default command arguments field – The specified default arguments are directly attached to the application URL when the application is run from the Appeon Developer toolbar.

- If both options are selected – A dialog box pops up with the default arguments filled in, as shown in Figure 4-22. You can either change the default arguments or leave them alone.

Figure 4-22: Command line argument dialog box

To run a Web application with arguments from Clients, type the application URL with the arguments in the Internet Explorer address bar.

The format of a Web application URL with arguments is: “http://WebAppURL/index.htm?ArgValue”. ArgValue indicates the arguments that will be passed to the application. The arguments must be separated by commas if there are more than one argument.

Application language
Step 1 – Click *Language Options* on the *Misc Settings* tab. The Language Settings dialog box is displayed, as shown in Figure 4-23.

**Figure 4-23: Application language settings**

![Language Settings dialog box](image)

Step 2 – Select a language from the *System Prompt Language* dropdown listbox. System prompts and messages are error messages, informational messages, warnings, notices, and prompts that are displayed when you run Appeon Web applications.

**Performance settings**

Appeon recommends that you take advantage of the performance settings to boost the application’s runtime performance. To produce the best performance, perform the following steps:

Step 1 – Click the *Performance Settings* button on the *Misc Settings* tab. The Performance Settings dialog box is displayed, as shown in Figure 4-24.

**Figure 4-24: Performance settings**

![Performance Settings dialog box](image)

Step 2 – Select the performance features based on your needs. Table 4-6 describes how to take advantage of the features.
### Table 4-6: Performance features

<table>
<thead>
<tr>
<th>Performance Feature</th>
<th>Description</th>
<th>When To Use…</th>
</tr>
</thead>
<tbody>
<tr>
<td>10X Web File Compression</td>
<td>Compresses files when they are transferred over the network.</td>
<td>Always</td>
</tr>
<tr>
<td>Merge files</td>
<td>Merges the small files during the application deployment. The small files will be downloaded to the client in one file package at one call, instead of being downloaded one by one at separate calls.</td>
<td>In the production stage.</td>
</tr>
<tr>
<td>Performance Analyzer compliance</td>
<td>Enables the application to be deployed in a method compliant with the Performance Analyzer tool. The application deployed with this option can run with the Performance Analyzer tool at runtime to diagnose the performance issues. For more about the Performance Analyzer tool, see Appeon Performance Tuning Guide. Note: as the data is not encrypted during transmission at runtime, therefore, this option is recommended to use only at the test or development stage.</td>
<td>In the development stage.</td>
</tr>
<tr>
<td>Download ActiveX files in a single thread</td>
<td>Downloads the two ActiveX files in the same thread at runtime. Using the same thread to download the two ActiveX files can speed up the download under particular network conditions.</td>
<td>Always</td>
</tr>
</tbody>
</table>

**Parsing options**

Unsupported identifiers refer to identifiers that are supported in PowerBuilder but unsupported in Appeon, due to the different naming convention between PowerScript and JavaScript/XML. For detailed explanation of such unsupported identifiers, refer to Appeon Features Help.

With this automatic-renaming option enabled, when the Appeon Deployment Wizard deploys an application it automatically detects unsupported identifiers in the application and renames these identifiers to supported ones in the Web files it generates for the application, while keeping the application source code unchanged. After the deployment, you can get the list of all identifiers that are renamed in the deployment log in the Information Manager (Chapter 9: Using Information Manager).

The automatic-renaming feature will not work if the option is disabled. In this case, you must rename the unsupported identifiers by yourself, following the guidance of the UFA report and Appeon Features Help.
Figure 4-25: Parsing options

The log-writing mode determines the content of the log files generated by Appeon Developer, including the deployment logs and feature analysis logs. Select the desired mode from the Appeon Developer Logs box, as shown in Figure 4-26.
The logging options enable you to select the level of information contained in the logs according to your needs:

Table 4-7: Log writing options

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Off mode</td>
<td>Generates no log files. This mode offers the fastest performance since nothing is written to a log file.</td>
</tr>
<tr>
<td>Summary mode</td>
<td>Generates log files with basic execution information. This is useful for tracking errors that have occurred, but inadequate for detailed troubleshooting. Use this mode once the application is stable.</td>
</tr>
<tr>
<td>Debug mode</td>
<td>Generates log files with detailed execution information for troubleshooting obscure and esoteric issues. This is useful for technical support, but performance speed will slow down when using this mode.</td>
</tr>
</tbody>
</table>

You can view or delete these log files. For detailed instructions, refer to Chapter 9: Using Information Manager.
**Web Service Profiles**

A Web service profile contains the settings for the Web service(s) to be invoked in the Appeon deployed application, including the WSDL file, service name and port to be used by the Web service.

After you specify the PBL files of the application in the Basic Settings tab, the Web service proxy objects, WSDL files, service, and port, which are used in the application, are automatically added to the Web Service Profiles tab, as shown in Figure 4-27.

A valid WSDL file must meet the following requirements:

- The WSDL file must be a .wsdl or .asmx file.
- The WSDL file must reside on the local computer. Appeon cannot access the WSDL file in a remote machine.
- WSDL files used in the same application cannot have the same name, even if they reside in different locations.

![Figure 4-27: Web Server Profiles tab](image)

Choose which proxy object(s) will be invoked in the deployed application by selecting the *Used* column of the object. Click *Select All* to select all objects; click *Deselect All* to deselect all objects; or click *Invert Selection* to select all objects which were not selected and deselect the previously selected objects.
If the WSDL file, service and port are not automatically specified for a proxy object, follow the steps below to select them:

Step 1 – Select the proxy object and click *Configure*. The Configure dialog box pops up, as shown in Figure 4-28.

**Figure 4-28: Configure dialog box**

![Configure dialog box](image)

Step 2 – Click *Browse* to select a WSDL file on the local computer, and then select the service and port.

A WSDL file may contain multiple services and each service may contain multiple ports.

Step 3 – Click *OK* to save the configuration and return to the *Web Service Profiles* tab page.

**Runtime Settings**

The Runtime Settings tab (Figure 4-29) specifies the customized information that will be displayed on the Appeon Download Center when the application starts. You can specify the company and application names according to your actual needs.
Summary

The Summary tab page offers a summary view of all configurations of an application profile. It lists the configurations in a treeview structure, and allows you to review and modify their values directly.

Use the following two methods to input or modify the configuration (property values):

- For properties that you can input or select values by using a text field, radio buttons, or dropdown list, such as Mode, Web URL, PBL version etc., you can directly double click their Property Values column and then input, modify or select values.

- For properties that you can add/delete multiple items by using a dialog box, such as application PBLs, INI files etc., you can right click the property name in the Application column and then select “Add…” or “Delete All…” from the popup menu.

The configuration changes you make in this Summary tab page will be immediately synchronized to the other tab pages, but only after you click the OK button, the changes will be saved.
4.3.2 Managing database type profiles

Database type profiles configuration is the easiest way to manage the database types that Appeon Deployment Wizard supports for application deployments. You can configure database type profiles in the DB Type Profiles tab, and apply a database type profile for an application deployment by selecting the profile in the Appeon Developer Configuration Wizard or in the DB Settings tab of the application profile settings.

Database type profiles in Appeon Developer are different from database profiles in PowerBuilder:

1. You only need to configure one database type profile for a given database type. For example, if application A connects to ASA database d1 and application B connects to ASA database d2, you can use the database type profile “Sybase ASA 7/8/9/10/11” for both applications, although the connection information in the profile “Sybase ASA 7/8/9/10/11” sets up connection to ASA database d1.

2. Database type profiles are not used for setting up connections with application databases. Instead, they are mainly used by Appeon Deployment Wizard to call the relevant database driver for converting PowerBuilder SQL statements.

To create or modify a database type profile, take the following steps:

Step 1 – Click the Configure button on the Appeon Developer toolbar, and select the DB Type Profiles tab in the Application Developer Configuration page as shown in Figure 4-30.
The *Database Type* column lists all the database types that are supported by Appeon; and the *Configured* column indicates whether a database type profile has been configured.

**Figure 4-30: DB Type Profiles**

Step 2 – Select a database type (for example, Sybase ASE 12.x/15.x), and click the *Configure* button to configure the database type profile (as shown in Figure 4-31).
Step 3 – Specify the information needed for setting up the database type profile.

The **Outer Join Syntax** setting allows you to specify how Appeon format the SQL syntax for outer joins, you can select corresponding format from the drop-down list in the Database Settings frame. Regardless of the settings in PowerBuilder, it is recommended to use ANSI as the outer join syntax format in Appeon.

**ODBC Interface** – Select this option if using an ODBC interface. Specify the ODBC data source, User ID, and Password.

**Table 4-8: ODBC interface settings**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Source</td>
<td>All the data sources configured in Windows ODBC Administrator are listed. Select a proper data source of the configured database type from the dropdown listbox. It is not necessary to specify the actual databases that the application uses. Click <em>ODBC Administrator</em> to configure an ODBC data source if it is not available.</td>
</tr>
<tr>
<td>User ID</td>
<td>Type in a recognized (set) user name to login to the database. If no user name has been set, leave this field blank.</td>
</tr>
<tr>
<td>Password</td>
<td>Type in the password used to log in to the database. If no password has been set, leave this field blank.</td>
</tr>
</tbody>
</table>

**Native Interface** – Select this option if using a native database interface. Specify the server name/IP, User ID, Password, and the database name.
Note: Not all the listed database types are supported by Appeon for the native database interface. If Sybase IQ, Sybase ASA 7/8/9/10/11 or IBM DB2 is your database type, the Native Interface options are disabled.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Enter the machine name or IP address of the server where the target database resides.</td>
</tr>
<tr>
<td>User ID</td>
<td>Type in a recognized (set) user name to login to the database. If no user name has been set, leave this field blank.</td>
</tr>
<tr>
<td>Password</td>
<td>Type in the password used to login to the database. If no password has been set, leave this field blank.</td>
</tr>
<tr>
<td>Database</td>
<td>Specify the name of the database that will be used by the application.</td>
</tr>
<tr>
<td>Release</td>
<td>(ASE only) Specify the version number of ASE database.</td>
</tr>
</tbody>
</table>

Step 4 – Specify the “Enclose table and column names in double quotes” option. It determines whether Appeon Developer encloses the names of tables, columns, indexes, and constraints in double quotes when it generates the database syntax during deployment.

Notes:
1) When this option is selected, verify that the quoted_identifier setting is set to ON in the database server.
2) Clear the check box of this option when the updated table name of DataWindow contains the owner name.

Step 5 – Click Test Connection to test the connection to the database.

4.3.3 Managing server profiles

A server profile is a set of parameters for Appeon Developer to connect to, and deploy applications to, a particular Appeon Server or Web server. Creating and using server profiles is the easiest way to manage your application deployments because you can:

- Select one or more Appeon Server profiles and Web Server profiles to set up a deployment profile.
- Easily add, edit, and remove server profiles.
- Test connections in server profiles. If the connection is successful, it means that the application can be deployed to the server configured in the server profile.
- Manage which applications are accessible from the Appeon Developer shortcut Run button. The shortcut lists the applications that are deployed to the default server profiles.

4.3.3.a Server Profiles tab page

There are two main parts in the Server Profiles tab page, as shown in Figure 4-32. The upper part is for management of Appeon Server profiles, while the lower part is for management of Web server profiles.
Figure 4-32: Server Profiles

You can select an Appeon Server profile or a Web Server profile as the DEFAULT server profile in the Server Profiles tab page. Appeon Developer’s Run shortcut reads the default server profiles and lists the applications that are deployed to default server profiles, enabling you to access them quickly. Applications that are deployed to non-default server profiles are not available in the shortcut. For details about the shortcut, refer to Section 8.3.2: Running Web applications.

Edit, add or delete server profiles

Two sets of Edit, Add, and Delete buttons are available in the Server Profiles tab page for you to modify, create, or remove Appeon Server or Web Server profiles. Sections 4.3.3.b and 4.3.3.c provide instructions on how to configure the properties for an Appeon Server or Web Server profile when you modify or create a new server profile.

4.3.3.b Appeon Server profile settings

Before you edit or add an Appeon Server profile, make sure the Appeon Server specified in the profile is running, and that the Appeon Developer computer can successfully connect to the Appeon Server computer.

If you want to deploy to an Appeon Server cluster, you must create profiles for each participating Appeon Server and deploy the application to all Appeon Servers in the cluster.

Figure 4-33 shows the Appeon Server Profile Configuration window that displays when you click the Edit or Add button in the Appeon Server group box of the Server Profiles tab page.
Table 4-10 lists detailed instructions for how to specify the properties for an Appeon Server profile.

Table 4-10: Instructions for creating an Appeon Server profile

<table>
<thead>
<tr>
<th>Property</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Profile Settings</strong></td>
<td></td>
</tr>
<tr>
<td>Profile Name</td>
<td>Assign a name to the Appeon Server profile. You should use names that are easy to remember and identify (for example “Appeon Server for Test” or “Remote Appeon Server”).</td>
</tr>
<tr>
<td>** Appeon Server Settings**</td>
<td></td>
</tr>
<tr>
<td>Server Type</td>
<td>Select J2EE if Appeon Server is installed to EAServer, WebLogic, WebSphere, JEUS, or JBoss; select .NET if Appeon Server is installed to Microsoft .NET Framework.</td>
</tr>
<tr>
<td>Server</td>
<td>Enter the IP address or the machine name of Appeon Server.</td>
</tr>
<tr>
<td>Server Port (http)</td>
<td>Enter the HTTP port number used by Appeon Server. Note: Do not enter an HTTPS port here, Appeon Developer must use an HTTP port to connect with Appeon Server, though the Appeon Web application can be connected with an HTTPS port.</td>
</tr>
</tbody>
</table>
### AEM Settings

<table>
<thead>
<tr>
<th>AEM Settings</th>
<th>AEM URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>The URL for Appeon Enterprise Manager (AEM) will be automatically generated after Server and Server Port are specified. The URL will be in the following format: <a href="http://server:port/AEM">http://server:port/AEM</a>, (for example, <a href="http://localhost:9988/AEM">http://localhost:9988/AEM</a>).</td>
<td></td>
</tr>
</tbody>
</table>

**Note:**
1) If you use EAServer as Appeon Server, the Appeon installation program creates an HTTP listener (localhost: 9988) for Appeon Server. If the Appeon Server configured is a local computer, the following URL should always work: http://localhost:9988/AEM.
2) DO NOT use a “localhost” listener in a production environment.

<table>
<thead>
<tr>
<th>Connection method</th>
<th>Select HTTP or HTTPS to connect to AEM.</th>
</tr>
</thead>
<tbody>
<tr>
<td>If the Web Server is configured as an SSL Web Server, check the HTTPS (secure) option. Otherwise, check the HTTP (insecure) option.</td>
<td></td>
</tr>
</tbody>
</table>

### Deployment Security Settings

<table>
<thead>
<tr>
<th>Deployment Security</th>
<th>Deployment Security Settings in AEM. If Deployment Security is set to ON, the state will be “Enabled”, and you must specify the deployment user name and password. By default, the Deployment Security in AEM is OFF, so the state is “Disabled”.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>Enter the username used to deploy the application. The username and password for the deployment security feature are configured in the AEM. The username can be left blank if the deployment security feature is turned off in AEM.</td>
</tr>
<tr>
<td>Password</td>
<td>Enter the password used to deploy the application. The username and password for the deployment security feature are configured in the AEM. The password can be left blank if the deployment security feature is turned off in AEM.</td>
</tr>
</tbody>
</table>

After profile configuration, perform the following steps to make sure the Appeon Server profile can be successfully used for application deployments:

- Check whether the deployment security settings are configured correctly. The username and password in the deployment security settings must be the same as those configured in AEM. Make sure you retrieve the correct deployment username and password from the AEM administrator.
- Test Appeon Server settings by clicking the Test Appeon Server Settings button. DO NOT proceed to the next step until the testing succeeds.

### 4.3.3.c Web Server profile settings

Before you edit or add a Web Server profile, make sure the Web Server specified in the profile is running so the Appeon Developer computer can successfully connect to the Web Server computer.

Figure 4-34 shows the Web Server Profile Configuration window that displays when you click the Edit or Add button in the Web Server group box of the Server Profiles tab page.
Figure 4-34: Web Server profile configuration window

Table 4-11 lists detailed instructions for how to specify the properties for a Web Server profile.

**Table 4-11: Instructions for creating a Web Server profile**

<table>
<thead>
<tr>
<th>Property</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Profile Settings</strong></td>
<td></td>
</tr>
<tr>
<td>Profile Name</td>
<td>Assign a name to the Web Server profile. You should use names that are easy to remember and identify such as “Web Server for Test” or “Production Web Server”.</td>
</tr>
<tr>
<td>File Transfer Type</td>
<td>Use <a href="#">Local Server</a> if the Web Server is a local computer. Use <a href="#">Remote Server</a> if the Web Server locates remotely.</td>
</tr>
<tr>
<td><strong>Web Server Settings</strong></td>
<td></td>
</tr>
<tr>
<td>Server Type</td>
<td>Select the Web server type. Appeon supports the following Web Server types: Apache, EAServer, Microsoft IIS, WebLogic, WebSphere, and JBoss.</td>
</tr>
<tr>
<td>HTTP Server</td>
<td>Enter the IP address or the machine name of Web Server.</td>
</tr>
<tr>
<td>HTTP Port</td>
<td>Enter the Web Server port number. Note: Do not enter an HTTPS port here, even though the Web server is configured as an SSL Web server, because Appeon Developer must use an HTTP port to connect with Web server. However, you can access the deployed application with the HTTPS port.</td>
</tr>
</tbody>
</table>
Local Server Settings

| Web Root Path | Enter the home directory of the Web server (for example: for EAServer, the home directory is C:\Program Files\Sybase\EAServer6\html; for IIS, the home directory is C:\inetpub\wwwroot; for Weblogic 8, the home directory is C:\bea\user_projects\domains\appeon_domain\applications; for Weblogic 9/10, the home directory is C:\bea\user_projects\domains\appeon_domain\autodeploy; for WebSphere, the home directory is C:\IBM\WebSphere\AppServer\installableApps). Click the Browse button to select the home directory of the Web server. |

FTP File Transfer Settings

| FTP Port | Enter the port number of the FTP server. The typical FTP port is port 21. If you use a remote Web Server, Appeon Developer will upload the files to Web Server using FTP protocol. Note that Appeon Developer works as a passive FTP client when uploading files. So you need to make sure that you are using passive FTP and the FTP data port and command port are configured properly in the firewall. |

| FTP Username | Enter the username for FTP login. If the FTP server offers “anonymous” access then the username should be “anonymous”. |

| FTP Password | Enter the password for FTP login. If no password is set for the FTP server or the FTP Username is anonymous, leave this field blank. |

File Compression Settings for War/Jar

| Appeon Compression | Select to use the Appeon compression method to compress the Web files into War or Jar packages when deployed to JBoss, WebSphere, or WebLogic Web server. |

| JDK Compression | Select to use the JDK compression method to compress the Web files into War or Jar packages when deployed to JBoss, WebSphere, or WebLogic Web server. Click Browse to specify the location of JDK 1.4 or above, for example, C:\j2sdk1.4.2_04. To ensure the most compatibility, JDK compression is recommended for J2EE servers such as JBoss, WebSphere, and WebLogic. |

After the profile configuration, perform the following steps to make sure the Web Server profile can be successfully used for application deployments:

- Test Web Server settings by clicking the Test Web Server Settings button. DO NOT proceed to the next step until the testing succeeds.

  If the Web Server is a remote server, refer to the Two requirements for FTP settings section to make sure the configuration for the Web Server profile is successful.

  If the Web Server is an SSL Web server, refer to the If the Web Server is an SSL Web Server section to make sure the configuration for the Web Server profile is successful.
Two requirements for FTP settings

If you configure a Web Server profile for a remote Web Server, make sure the FTP settings in the Web Server profile meet the following two requirements:

1. The user name and password for accessing the FTP server should have permission to read and write files to the FTP server of the Web Server.

2. The FTP home directory should be mapped to the Web Server’s Web root.

The following steps use the Microsoft IIS FTP service as an example to show you how to fulfill the requirements. You should find that the settings for other FTP types are similar to the settings for Microsoft IIS FTP.

Step 1 – On the FTP server (Web Server), open the Internet Services Manager in Administrative Tools, as shown in Figure 4-35.

Figure 4-35: FTP configuration

Step 2 – Right click on Default FTP Site and select Properties in the popup menu. The Default FTP Site Properties Window is displayed.

Step 3 – Go to the Home Directory tab and verify that:

- The Local Path is the full path to the Web Server document root.
  
  If the Web Server is an Appeon Server Web Server running on Windows, the local path is the full path to the EAServer document root: %JAGUAR%\html (Figure 4-36), if the Web Server is an IIS server, the local path is C:\inetpub\wwwroot.
  
  %JAGUAR% is the EAServer installation directory (for example: C:\Program Files\Sybase\EAServer).

- The Write property of the FTP Site Directory is enabled.
Step 4 – Go to the Directory Security tab and verify that the Granted Access option is checked, as shown in Figure 4-37.
If the Web Server is an SSL or secure Web Server, you can configure the Web Server profile following the configuration instructions provided in Table 4-11, except for the following two points:

1. You must configure an HTTPS listener and port number for the Web Server.
2. You must specify an HTTP (not HTTPS) listener and its port number in the HTTP server settings of the Web Server profile, because Appeon Developer must use the HTTP protocol when deploying Web files to the server. After the application is deployed to the Web server, you can input “https” instead of “http” and the HTTPS port in the URL to access the application, (for example, https://192.0.0.80:8181/appeondemo/).

4.3.4 Managing deployment profiles

A deployment profile associates specified Web server(s) and Appeon Server(s) as a group used for Web deployment. You may create several deployment profiles (e.g. local deployment, test deployment, production deployment, and so on), and set the most commonly used profile as the default profile.

A deployment profile is based on server profiles. Before configuring the deployment profiles, make sure you have set up one server profile for each of the Appeon Servers and Web servers to be used for Web conversion. You can add as many deployment profiles as you need, but there can be only one default deployment profile.
### 4.3.4.a Deployment Profiles tab page

On the *Deployment Profiles* tab page, you can edit, add, or delete deployment profiles and specify a default deployment profile.

![Deployment Profile tab page](image)

Figure 4-38: Deployment Profile tab page

Table 4-12 provides instructions for how to use the Deployment Profile tab page.

<table>
<thead>
<tr>
<th>Use this button</th>
<th>To…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default radio button</td>
<td>Specify the default deployment profile used for Web deployment.</td>
</tr>
<tr>
<td>Edit button</td>
<td>Modify an existing deployment profile. This will open the Deployment Profile Configuration dialog box.</td>
</tr>
<tr>
<td>Add button</td>
<td>Create a new deployment profile. This will open the Deployment Profile Configuration dialog box.</td>
</tr>
<tr>
<td>Delete button</td>
<td>Remove a deployment profile.</td>
</tr>
</tbody>
</table>

### 4.3.4.b Deployment profile settings

In the Deployment Profile Configuration dialog box, set up the necessary configuration, as shown in Figure 4-39.
Table 4-13 provides detailed instructions to specify the deployment profile settings.

### Table 4-13: Deployment Profile settings

<table>
<thead>
<tr>
<th>In this field</th>
<th>You can…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile Name</td>
<td>Type the deployment profile name.</td>
</tr>
<tr>
<td></td>
<td>You can use names that are easy to remember and identify such as “Test</td>
</tr>
<tr>
<td></td>
<td>Deployment”, “Remote Deployment”, and so on.</td>
</tr>
<tr>
<td>Appeon Server</td>
<td>Select a check box to include the Appeon Server in the deployment profile.</td>
</tr>
<tr>
<td></td>
<td>All the Appeon Server profiles you have created are listed. More than one</td>
</tr>
<tr>
<td></td>
<td>Appeon Server can be selected.</td>
</tr>
<tr>
<td>Web Server</td>
<td>Select a check box to include the Web Server in the deployment profile.</td>
</tr>
<tr>
<td></td>
<td>All the Web server profiles are listed. More than one Web server can be</td>
</tr>
<tr>
<td></td>
<td>selected.</td>
</tr>
</tbody>
</table>

Click OK to save the new settings and return to the Deployment Profiles tab.

### 4.3.5 Managing connection cache profiles

Note: This Connection Cache Profiles tool works for EAServer 5.x, but does not work for EAServer 6.x, JBoss, WebLogic, and WebSphere.

To create connection cache data source for EAServer 6.x, JBoss, WebLogic, and WebSphere, refer to the Database Connection Setup chapter in the Appeon Server Configuration Guide.
Appeon Web applications use JDBC connection caches created in Appeon Server to connect to the database. “Connection Cache Profile” tab (Figure 4-40) allows you to maintain a storage of connection caches which you can specify to use for Appeon applications. Its function is the same as the one provided in AEM, and changes made in the tab or AEM are automatically synchronized.

**Figure 4-40: Connection cache profile**

Follow the instructions below to create a connection cache on the Connection Cache Profile tab:

Step 1 – Select an Appeon Server from the Appeon Server list box. Only the configured Appeon Server profiles are listed.

Step 2 – When it is EAServer 5.x, input the correct user name and password to log in to the EAServer application server, as shown in Figure 4-41.

**Figure 4-41: Login dialog box**

The connection caches existing on the application server will be displayed, as shown in Figure 4-42.
Figure 4-42: Connection caches existing on the application server

Step 3 – Click the Add button to create a connection cache in the application server. The Add dialog box is displayed, as shown in Figure 4-43.
Figure 4-43: Add a connection cache

Table 4-14 provides instructions for how to specify the connection cache settings.

<table>
<thead>
<tr>
<th>Settings</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type the name of the connection cache. The connection cache name can contain a combination of letters, underscores (“_”), dot (“.”), and numbers. Do not use double-byte characters (such as Chinese, Korean or Japanese characters) or leave a leading or trailing space.</td>
</tr>
<tr>
<td>Driver</td>
<td>Select the driver type for the connection cache.</td>
</tr>
<tr>
<td>Host or IBM Informix Server</td>
<td>Type the database host.</td>
</tr>
<tr>
<td>Port</td>
<td>Type the database port.</td>
</tr>
<tr>
<td>Database Name</td>
<td>Type the database name.</td>
</tr>
<tr>
<td>ODBC Data Source</td>
<td>Type the data source name only when the driver type is ODBC-JDBC Bridge.</td>
</tr>
<tr>
<td>User Name</td>
<td>Type the database login username. The username is set on the database server.</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Password</td>
<td>Type the database login password. The password is set on the database server.</td>
</tr>
<tr>
<td>Maximum Connection Pool Size</td>
<td>Specify the maximum number of connections Appeon Server opens and pools on startup.</td>
</tr>
<tr>
<td>Minimum Connection Pool Size</td>
<td>Specify the minimum number of connections Appeon Server opens and pools on startup.</td>
</tr>
<tr>
<td>Connection Timeout</td>
<td>Specify the timeout period for the connection.</td>
</tr>
<tr>
<td>Command Timeout</td>
<td>Specify the timeout period for the commands.</td>
</tr>
<tr>
<td>Connection Lifetime</td>
<td>Specify the lifetime period for the pooled connection.</td>
</tr>
<tr>
<td></td>
<td>To be specific, you can specify the timeout period from the time a connection is created to the time when the connection returns to the pool. If the actual lifetime of a connection exceeds that specified in Connection Lifetime, the connection will be terminated. It is recommended to be used in cluster configuration when forcing the load balancing between a running server and a server just brought online. If the value is set to 0, the pooled connection allows the maximum connection timeout.</td>
</tr>
<tr>
<td>Other Options</td>
<td>Specify the connection_authentication property for ASA database. The value should be “authentic_code=###”. “authentic_code” is the keyword, text after “=” should be the authorization code. Appeon Server will automatically set the connection_authentication property when connecting with the ASA database. This setting is effective to ASA database only.</td>
</tr>
<tr>
<td>Dynamic Database Connection</td>
<td>Select whether to enable dynamic database connection. When it is on, the LogID and LogPass of the Transaction object will be used to connect to the database; when it is off, the user name and password specified in the Connection Cache will be used to connect to the database.</td>
</tr>
<tr>
<td>Pooling</td>
<td>Select whether to use the connection pool.</td>
</tr>
</tbody>
</table>

Refer to the Appearon Server Configuration Guide for how to specify the settings for different JDBC drivers and database types.
5 Using UFA Tool

5.1 Overview

Unsupported Features Analysis (UFA) tool helps you analyze an application for unsupported features and heavy windows. Correspondingly, two reports will be generated: one is the Unsupported Features Analysis report (UFA Report) and the other is the Heavy Window report.

Table 5-1 gives brief descriptions of both reports.

<table>
<thead>
<tr>
<th>Purpose of the report</th>
<th>UFA Report</th>
<th>Heavy Window Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enables you to work around or remove unsupported features in order to make the PowerBuilder application suitable for Appeon Web migration.</td>
<td>Gives you an idea of which windows to tackle for performance tuning and what PowerBuilder features to thin out. The general principle is: the lighter the window is, the better the window will perform in the Web application.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>What’s reported</th>
<th>Unsupported PowerBuilder coding features.</th>
<th>Complex windows that most affect the performance of the deployed Web application.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>How to generate the report</th>
<th>Use the Unsupported Features Analysis tool, or perform the first two tasks in Appeon Deployment Wizard.</th>
<th>Use the Unsupported Features Analysis tool, or perform the first two tasks in Appeon Deployment Wizard. Requirement: the Generate Heavy Window Report option must be enabled in the Appeon Deployment Wizard.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Features</th>
<th>These features apply to both reports:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automatic incremental analysis. After the first analysis, the subsequent analysis is automatically incremental to save time, but the report generated still contains all the information as if it were a full analysis.</td>
<td></td>
</tr>
<tr>
<td>Analysis at application, PBL or object level. You can choose to generate UFA Report or Heavy Window report for an application, PBL(s) in the application, or object(s) in the application.</td>
<td></td>
</tr>
</tbody>
</table>

This chapter describes:

- How to use the UFA tool to analyze an application and generate the UFA Report.
- How to use the UFA tool to work around unsupported features effectively.
- How to use the UFA tool to manipulate the UFA Report. For example, you can adjust the view of the UFA Report so it suits your preferences.
- Undetected unsupported features. There are a small number of unsupported features that the UFA Report cannot detect.
- The usage of Heavy Window Report.
5.2 Analyzing an application

5.2.1 Tasks required before you perform feature analysis

Step 1 – Make sure the computer is powerful enough to support feature analysis.

Unsupported Features Analysis is CPU and memory intensive. A faster CPU can speed up the process. Additional memory is useful for ensuring that RAM will not be exhausted when analyzing a large application. Regarding system requirements for the Developer machine, refer to the System Requirements section in the Appeon Installation Guide.

Step 2 – Close any unnecessary running programs or Windows services in order to have the maximum amount of memory available for Features Analysis.

Step 3 – Perform an incremental build or object regeneration for the PowerBuilder application. Right-click on the application target and select a build option (for example, “Incremental Build”) as shown in Figure 5-1.

![Figure 5-1: Incremental Build](image)

Table 5-2 shows you which type of build options you should select for the application.

Table 5-2: Recommended build options

<table>
<thead>
<tr>
<th>Build Option</th>
<th>What It Does</th>
<th>Recommended For</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regenerate</td>
<td>Refreshes the timestamp for a PowerBuilder object.</td>
<td>Object-level Features Analysis. If the object is regenerated, the object will be analyzed even if no change has made to it.</td>
</tr>
<tr>
<td>Incremental</td>
<td>Refreshes the objects that have been changed.</td>
<td>Application-level Features Analysis.</td>
</tr>
<tr>
<td>Full</td>
<td>Refreshes all objects.</td>
<td>Not recommended.</td>
</tr>
</tbody>
</table>

Step 4 – Perform PBL optimization for the PowerBuilder application.
The purpose of PBL optimization is to remove gaps from libraries and defrag the storage of objects to get rid of potential problems in PBL files.

Step 5 – (Required only if you want to generate a Heavy Window Report) Check the Generate Heavy Window Report option in the Application Deployment Wizard, as shown in Figure 5-2.

**Figure 5-2: Enable Heavy Window Report**

![Figure 5-2: Enable Heavy Window Report](image)

5.2.2 Accessing the UFA tool

Click the Analyze button ( ) in the Appeon Developer toolbar to launch the UFA tool. The Unsupported Feature Analysis Report window (UFA Report window) will be displayed.
Using UFA Tool

Figure 5-3: UFA Report Window

![UFA Report Window](image)

UFA Report window provides a variety of menus to help you manage unsupported features. Table 5-3 gives a brief description of these menus. Some of them are also listed as shortcut menus on the toolbar.

Table 5-3: UFA Report Window

<table>
<thead>
<tr>
<th>Menu</th>
<th>Description</th>
</tr>
</thead>
</table>
| File | Generates a UFA report. See section 5.2.3.  
Opens or saves a UFA report. See Section 5.3.2.a. |
| Edit | Opens and modifies the unsupported source code in the PowerBuilder painter. See Section 5.3.1.  
Searches for the unsupported features. See Section 5.3.2.c.  
Filters the unsupported features according to objects, unsupported feature types or priorities. See Section 5.3.2.d.  
Expands or collapses treeview.  
Displays the report in the specified level. See Section 5.3.2.e. |
| View | Selects a display mode of the objects in the Unsupported Feature List treeview. See Section 5.3.2.b. |
### 5.2.3 Performing feature analysis

There are two ways you can perform a feature analysis of an application:

- Use the Unsupported Features Analysis tool. This option enables you to do a feature analysis of the whole application, or PBL(s) in the application, or objects in the application.

- Use the first two tasks in Appeon Deployment Wizard. When you use Appeon Deployment Wizard to deploy an application, the first task the wizard performs is a feature analysis of the whole application.

This section mainly gives instructions on the first option (using the Unsupported Features Analysis tool). For more information about the second option, refer to Section 6.3: Deployment process.

Step 1 – Select the File | New Report menu in the UFA Report window. The New UFA Report dialog box will appear, as shown in Figure 5-4.
Figure 5-4: New UFA Report dialog box

Table 5-4 gives a brief description of the elements in the New UFA Report window.

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application profile</td>
<td>Provides a list of application profiles for you to select and analyze.</td>
</tr>
<tr>
<td>dropdown list</td>
<td>Only application profiles configured in the Appeon Developer</td>
</tr>
<tr>
<td></td>
<td>Configuration window will be listed.</td>
</tr>
<tr>
<td>Search field</td>
<td>Searches for PBLs or objects that contain the text you enter.</td>
</tr>
<tr>
<td>Treeview of objects</td>
<td>Gives the treeviews of PBLs and objects for the application selected in the</td>
</tr>
<tr>
<td></td>
<td>Application dropdown list and allows you to select the whole application or</td>
</tr>
<tr>
<td></td>
<td>only some objects to analyze.</td>
</tr>
<tr>
<td>Display Full Path option</td>
<td>Gives you options to display or hide the full path of the PBLs and objects</td>
</tr>
<tr>
<td></td>
<td>in the treeview.</td>
</tr>
<tr>
<td>Refresh button</td>
<td>Refreshes the PBL and object list in the treeview.</td>
</tr>
</tbody>
</table>

Step 2 – Select the application that you want to analyze from the Application profile dropdown list.

The default target to analyze is the default application profile in the Appeon Developer Configuration window.
Step 3 – Check the PBLs and/or objects in the treeview for which you want to generate the UFA Report.

The treeview lists all the PBLs and objects contained in the selected target. By default, all objects are selected.

Step 4 – Click OK to start the feature analysis.

The feature analysis of the selected objects begins, as shown in Figure 5-5.

**Figure 5-5: Unsupported Features Analysis Process dialog**

![Unsupported Features Analysis Process dialog](image)

Step 5 – Click Close when the analysis is completed. The UFA Report is loaded into the UFA Report Window.

### 5.3 Working with UFA Report

This section describes:

- How to modify unsupported features. You can use the UFA Report to effectively remove unsupported features in the application.
- How to manipulate the UFA Report in the UFA Report Window. You can adjust the view of the UFA Report so it suits your preferences.
- Undetected unsupported features. There are a small number of unsupported features that the UFA Report cannot detect.

#### 5.3.1 Modifying unsupported features

When the UFA Report is loaded, you can view all the unsupported features in the left treeview and modify them one by one.

Step 1 – Expand the unsupported feature list treeview and select the unsupported item.

The detailed analysis of the selected item will be displayed on the right of the window, as shown in Figure 5-6.
Figure 5-6: Unsupported feature details

The right text box provides a detailed diagnosis on the unsupported code in seven aspects: the Unsupported Code, Line Number, Location, Unsupported Item, Type, Reason, and Priority.

Step 2 – View the workaround for the unsupported item by right-clicking the unsupported item in the unsupported feature list treeview and selecting Workarounds solutions.

- Go To Workarounds Online – Opens the online Appeon Workarounds Guide at URL: http://www.appeon.com/support/documents/workarounds/6.0/.
- Go To Local Workarounds – Opens the local Appeon Workarounds Guide installed with Appeon Help.
Step 3 – Modify the unsupported source code in the PowerBuilder painter according to the solution provided in the *Appeon Workarounds Guide*.

Right click the unsupported item in the unsupported feature list treeview, and select *Edit* or *Edit Source* to open the source code in the PowerBuilder Script view.

- **Edit** – Displays the source code of an event or function, which contains the unsupported feature. This button is available only if the unsupported code resides in an event or function and it is not DataWindow-related.
- **Edit Source** – Displays the source code of the object, which contains the unsupported feature. This button is always available.

The PowerBuilder Script view automatically opens, as shown in Figure 5-8.
5.3.2 Manipulating the UFA Report

5.3.2.a Opening or saving a UFA Report

You can open or save a UFA Report using the File menu in the UFA Report Window.

- To open an existing UFA Report, select the File | Open menu. The Open UFA Report dialog box is displayed. If there is no UFA report available, select New Analysis and click OK to analyze an application.

- To save a UFA Report, select the File | Save menu.
- To save a UFA Report with a new name, select the File | Save As menu.
5.3.2.b Selecting report view mode

You can view the UFA Report in four different modes. Go to the View menu in the UFA Report Window and select the desired mode:

- **Category** – Enables you to view the unsupported features in the following categories: Unsupported Objects, Unsupported PowerScript, Unsupported Embedded SQL and Others.

- **Hierarchy** – Enables you to view the unsupported features in hierarchical order: PBL, Object, Control, Method (Event/Function).

- **Priority** – Enables you to view the unsupported features in priorities (you can go to the Tools | Define Priority menu to define the priorities of the PowerBuilder features): Have to modify, Suggest to modify, Can be ignored.

- **Summary** – Enables you to view summary information of each unsupported feature.

You can set the default view mode in the Tools | Options menu. For detailed descriptions, refer to Section 5.3.2.g.

5.3.2.c Searching for UFA Report items

To do a quick search, you can directly enter the text that you want to search in the search field of the UFA Report window. Or follow steps below to use the standard Find Text dialog box:

1. **Step 1** – Select the Edit | Find menu. Or right click in the unsupported feature list treeview and select Find.

2. **Step 2** – In the Find Text box, enter the text that you want to search.

3. **Step 3** – Select any other options that you want.

4. **Step 4** – Click Find Next.

**Figure 5-10: Find Text**

5.3.2.d Filtering UFA Report items

Filtering helps you restrict the number of the unsupported features displayed in the UFA Report and quickly find the target unsupported features, especially when there are a large number of unsupported features in the application.

You can filter the unsupported features by the objects they reside in, or by the feature types, or by the priorities they are specified to.
**Filtering by objects**

Step 1 – Select the Edit | Filter by Object menu. Or right click in the unsupported feature list treeview and select *Filter by Object*.

The Filter by Object dialog box is displayed, as shown in Figure 5-11.

**Figure 5-11: Filter by object**

![Filter by Object dialog box](image)

Step 2 – Make sure the unsupported objects you want to display in the report are on the right listbox, whereas the unsupported objects you want to hide are on the left listbox. Click *OK*.

To quickly select all objects of the same type, right click the list and select an object type. The UFA Report hides the unsupported objects that are on the left listbox in the Filter by Object window.

**Filtering by categories**

Step 1 – Select the Edit | Filter by Category menu. Or right click in the unsupported feature list treeview and select *Filter by Category*.

The Filter by Category dialog box is displayed, as shown in Figure 5-12.
Figure 5-12: Filter by Category

Step 2 – Make sure the categories of the unsupported features you want to display in the report are on the right listbox, whereas the categories of the unsupported features you want to hide are on the left listbox. Click **OK**.

The UFA Report hides unsupported features that belong to the categories that are on the left in the Filter by Category window.

**Filtering by priorities**

Step 1 – Select the Edit | Filter by Priority menu. Or right click in the unsupported feature list treeview and select **Filter by Priority**.

The Filter by Priority dialog box is displayed, as shown in Figure 5-13.
Step 2 – Uncheck the priorities of the unsupported features which you want to hide. For example, if you want to hide the features that “can be ignored”, uncheck the Can be ignored option.

Step 3 – Click OK.

The UFA Report only displays unsupported features of the specified priorities.

5.3.2.e Specifying report display level

You can quickly expand or collapse a node of the unsupported feature list treeview by selecting Expand Sub-node or Collapse Sub-node from the Edit menu or from the popup menu on any node.

You can also specify the display level using the following steps:

Step 1 – Select the Edit | Display Level menu.

Step 2 – In the Display Level box, enter the number of the level at which the unsupported feature list treeview will display.

Step 3 – Click OK.

The UFA Report only displays unsupported features of the specified level.

5.3.2.f Defining the priority settings of unsupported features

By default, Appeon divides all unsupported features into three priority levels:
- **Have to modify** – The features flagged with “Have to modify” are important in the application and will cause significant functionality loss if they are not modified or worked around.

- **Suggest to modify** – The features flagged with “Suggest to modify” will not necessarily cause functionality loss if they are not modified or worked around. You can decide whether to modify them according to their functionalities in the application.

- **Can be ignored** – The features flagged with “Can be ignored” are small or trivial features which will not compromise application functionality even if they are not modified or worked around.

To customize or change priority levels for unsupported features, take the following steps:

**Step 1** – Select the Tools | Define Priority menu. The Define Priority window is displayed, as shown in Figure 5-15.

**Figure 5-15: Define Priority dialog box**

Step 2 – Add, modify or delete the system default priorities in the left list box.

Step 3 – Change priority levels of unsupported feature types by selecting a priority level in the **Priority** column on the right box.

Step 4 – Click **Save** to save the priority settings.

Or click **Reset to Default** to restore the system default priority settings.

**5.3.2.g Customizing the general settings of the UFA Report**

You can define your preference settings of the UFA Report window. These settings include default report view mode, display formats of PBLs, feature count, unsupported source code edition, and filtering. The new settings will be saved as the default settings.
Select the Tools | Options menu in the UFA Report Window. The General Settings window is displayed, as shown in Figure 5-16.

**Figure 5-16: General Settings dialog box**

- **Default Report View** group box – Select the default view mode of the UFA Report. For detailed description of each view mode, refer to Section 5.3.2.b: *Selecting report view mode*.

- **Display Full Path of PBL Files** – Displays the full path and the file name of the PBLs in the UFA Report.

- **Display Unsupported Feature Count** – Displays the total number of the unsupported features in the UFA Report.

- **Auto start PowerBuilder and load target application** – Enables PowerBuilder to automatically start and load the target application when you click the Edit or Edit Source button in the UFA Report window.

- **Filter Previous Results** – Allows you to filter only the results of a previous filter. It narrows a filter that resulted in too much data the first time. If you want to search all data of the report, leave this option as unselected.

### 5.3.3 Undetected unsupported features

Appeon Unsupported Features Analysis is capable of highlighting the majority of unsupported features contained in the PowerBuilder application. There are still a small number of features that Unsupported Features Analysis will not detect, so they will not be listed in the UFA Report.

An application containing unsupported features can still be converted to the Web and may work correctly, depending on the type and number of unsupported features. If the Web application does not work correctly and the UFA Report does not indicate any unsupported
features that are likely to cause such problems, the problems may be caused by unsupported features that have not been detected. In this situation, refer to Undetected Features in Appeon Features Help, which documents the types of unsupported features that may not be detected.

5.4 Working with Heavy Window Report

You can access the Heavy Window Report from the Information Manager if the Generate Heavy Window Report option is enabled in the Appeon Deployment Wizard before the feature analysis or the application deployment is performed.

5.4.1 How a heavy window is measured

Two factors determine the heaviness of a window:

- The number of controls in the window.
- The complexity of the controls in the window (for example, a DataWindow with several DropDownDataWindows makes the window significantly heavier than a DataWindow without DropDownDataWindows).

Appeon assigns different weights to different types of controls. (Example: the weight of a DataWindow is 1.0, the weight of a computed field is 0.06, and the weight of a tab page is 0.15.) The weight of a window is determined by summing up the total weight of all the controls contained within it.

The time required to open a window increases proportionally with the weight of the window. As our tests show, it takes 0.5 of a second to open a window that weighs 8.0 using a computer with the following specifications: P4 1.8 GHz 512 K cache CPU and 256 MB RAM. It takes approximately 0.6 second to open a window that weighs 10.0. In different environments the time it takes to open a window can vary. Note that the elapsed time does not include the time used to connect to the database and retrieve data.

By default, the Heavy Window Report flags all windows that weigh 8.0 (default minimum load value) or more.

The weight of the different control types, as well as the default minimum load value, are all stored in the PerformanceStatConfig.xml file in the %Appeon%\Developer6.5 path. You can change the settings in the XML file based on actual requirements. (Example: change the minimum load value to 6.0 to get a report of all the windows that take about 0.4 second or more to open). However, Appeon strongly recommends that you do not change the default value of each type of control because each default value is a fixed value based on multiple testing of each control’s complexity. Most controls will significantly increase the loading time of a window as their number increases. If you change the default value of the controls, the minimum load value may not compute correctly. For example, the minimum load value adds up to 10.0, but the actual loading time may be much more or less than 0.6 of a second.

5.4.2 Reading Heavy Window Report

By using the Heavy Window Report, you can grasp a general idea of how many heavy windows are contained in the application. This report lists the oversize JavaScript files generated for the application and the heavy windows, including the number of complex controls and the minimum load value of each window, as shown in Figure 5-17. This helps you measure each window’s performance.
5.4.3 Making heavy windows lighter

For instructions on how to make heavy windows lighter, refer to the *Appeon Performance Tuning Guide*. 
6 Deploying PowerBuilder Applications

6.1 Overview

Appeon Deployment Wizard deploys PowerBuilder applications to Web applications. The deployment should take place only after the application has undergone careful analysis and is in compliance with the Basic and Architectural Requirements defined in Appeon Features Help.

The Appeon Deployment Wizard handles Web deployment in five steps:

1. Exports the PowerBuilder application source code.
2. Analyzes the PowerBuilder application to verify that it does not contain unsupported features.
3. Parses the PowerBuilder source code (PBL files) into Web application files that are stored on the local machine.
4. Uploads the DataWindow syntax to Appeon Server.
5. Transfers local Web files to the Web server by either file copy or FTP.

Notes:

1) Deploying to an SSL Web Server is no different than deploying to a standard Web Server. You only need to make sure that the SSL Web Server is set up correctly and is fully functional. For detailed instructions on configuring a Web server as an SSL Web Server, refer to the corresponding documents provided by the Web server vendor.

2) If you want to re-deploy an application which is now running online:
   - You can incrementally deploy the application while keeping the application running online, if non-database related source code and image files are changed.
   - You must shut down the running Web application and then full or incrementally deploy the application if the INI files or database-related source code are changed. Otherwise, the application may have runtime problems such as database connection failure or missing tables or fields.

6.2 Deployment performance

Web deployment is a process that can be very CPU and memory intensive, so we recommend that you use a developer computer that conforms to the recommended specifications outlined in the System Requirements section of the Appeon Installation Guide for a Developer PC to run Appeon Developer.

For more information regarding the deployment performance of different types of applications, refer to the Appeon Performance Tuning Guide.

Be sure to close any unnecessary programs and Windows services in order to have an optimal amount of memory available for the deployment process.

6.2.1 Speed of Web deployment process

The table below benchmarks the time taken or speed to compile various applications in PowerBuilder and deploy to the Web with Appeon. In general, the results indicate that the
speed of the Appeon Web deployment process is similar to the speed of the PowerBuilder compilation/build process.

Table 6-1: Appeon Web Deployment Time Benchmarks

<table>
<thead>
<tr>
<th>Application</th>
<th>PB EXE w/ PBDs</th>
<th>PB EXE w/DLLs</th>
<th>Appeon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales Application Demo (5.55M)</td>
<td>20 seconds</td>
<td>50 seconds</td>
<td>22 seconds</td>
</tr>
<tr>
<td>Appeon Code Example Demo (20.5M)</td>
<td>1 minutes 3 seconds</td>
<td>12 minutes 37 seconds</td>
<td>1 minutes 12 seconds</td>
</tr>
<tr>
<td>ACF Demo (21.5M)</td>
<td>2 minutes 5 seconds</td>
<td>24 minutes 46 seconds</td>
<td>2 minutes 12 seconds</td>
</tr>
</tbody>
</table>

As a general rule of thumb, each additional megabyte of PBLs will add approximately 8 seconds to the full deployment process. For example, it would take 4 minutes to deploy a 30MB PFC application. This would only apply for the initial or full deployment. All subsequent deployments should take no more than a few minutes or even possibly as little as a few seconds.

This rule of thumb applies to hardware configurations that meet the minimum Developer PC requirements stated in the Appeon Installation Guide. Machines with a slower clock-speed will take longer to perform the deployment.

6.2.2 Deployment duration for full deployments

Remember the following main points regarding the deployment time:

- The process is very CPU and memory intensive.
- The size and complexity of the application affect the deployment time.

The following is an example of how long it takes to deploy an application with 3.69 MB in PBLs and images using full deployment:

Table 6-2: Deployment duration

<table>
<thead>
<tr>
<th>Deployment Task</th>
<th>Elapsed Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1: Application Source Code Export</td>
<td>8 seconds</td>
</tr>
<tr>
<td>Task 2: Web File Generation</td>
<td>27 seconds</td>
</tr>
<tr>
<td>Task 3: Web Deployment</td>
<td>15 seconds (for a local deployment)</td>
</tr>
</tbody>
</table>

Test environment: a single CPU Intel P4 1.8 GHz with 256 MB RAM and 60 GB IDE hard drive

All generated Web files deployed to the server are a combined 1.25 MB in size.

6.2.3 Deployment duration for incremental deployments

Since the initial full deployment can be time-consuming, Appeon provides an Incremental Deployment for maintaining or upgrading an already-deployed application. This feature only re-deploys to the Web the incremental changes that have been made in the application. Appeon can detect whether an object has been modified and re-deploy only the changed objects.
6.3 Deployment process

6.3.1 Preparing the PowerBuilder application

Appeon recommends that you perform a build and PBL optimization on the PowerBuilder application before performing an Appeon Web Application Deployment. Table 6-3 shows you which type of build options you should select for the application.

Table 6-3: Build options

<table>
<thead>
<tr>
<th>Build Option</th>
<th>What It Does</th>
<th>When To Perform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incremental build</td>
<td>Refreshes the objects that have been changed.</td>
<td>Before a full or incremental deployment.</td>
</tr>
<tr>
<td>Full build</td>
<td>Refreshes all objects.</td>
<td>Not recommended.</td>
</tr>
<tr>
<td>Optimize PBLs</td>
<td>Removes unused space in the PBLs as stored on disk, and reduces object fragmentation.</td>
<td>After building the application but prior to deployment.</td>
</tr>
</tbody>
</table>

Performing an incremental build in PowerBuilder is necessary because when modifications are made to a parent class, the child class does not reflect the changes unless one of three things occurs:

1. The child class is opened and saved.
2. An incremental build is performed.
3. A full build is performed.

6.3.2 Specifying the deployment settings

Click the Deploy button (部署) in the Appeon Developer toolbar. The Appeon Deployment Wizard is displayed.
Table 6-4 describes how to specify the deployment settings for an application:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application to be deployed list box</td>
<td>Select an application to be deployed. The default application is listed by default. The application chosen for deployment will become the default application under Application Profiles in Appeon Developer Configuration.</td>
</tr>
<tr>
<td>Deployment Profile list box</td>
<td>Select a deployment profile to be used in Web deployment. The selected deployment profile determines which Appeon Server(s) and Web Server(s) the application will be deployed to. You must make sure the Appeon Server(s) and Web Server(s) are started before you perform the deployment.</td>
</tr>
<tr>
<td>Deployment Mode list box</td>
<td>Select one of the three available deployment modes: Full Application Deployment, Incremental Application Deployment, Deploy Already Generated Application</td>
</tr>
<tr>
<td>Unattended Deployment check box</td>
<td>Specify whether the whole deployment process will automatically proceed without displaying the Next button or waiting for interaction. If you want to view the messages or reports generated at each step, you can set the unattended deployment to false; the wizard will require you to click the Next button to start the next task until the whole conversion is complete.</td>
</tr>
</tbody>
</table>
Advanced button

Select the types of objects or files that will be uploaded to servers: DataWindow objects, other objects, INI files, external DLL/OCX files, or image files. By default, all objects and files will be uploaded.

This button is effective only when the Deploy Already Generated Application option is selected, therefore, make sure you have performed a full deployment on the application.

When using this function, verify that the target servers are the same ones used in the application deployments.

Generate Object Flags for Appeon Debugger

Specify whether to flag the objects for debug with Appeon Debugger. This option is available for the Debug PS/JS and Debug JS modes. De-selecting it will have the following two effects:

1) The deployment process will be faster;
2) The objects will not be able to be debugged in Appeon Debugger, though they can still be debugged in other debugging tools, such as Microsoft Visual InterDev.

Generate Unsupported Features Analysis Report check box

Specify whether to generate an unsupported feature analysis report during deployment. This report lists all unsupported PowerBuilder coding features in the application. You can use this report to remove or work around the unsupported features. For detailed instructions, refer to Section 5.3: Working with UFA Report.

Generate Heavy Window Report check box

Specify whether to generate a Heavy Window Report during deployment.

### 6.3.2.a Selecting the deployment mode

Table 6-5 describes the three different deployment modes and guides you in choosing the proper deployment mode.

<table>
<thead>
<tr>
<th>Deployment Mode</th>
<th>What It Does</th>
<th>Elapsed Time</th>
<th>When To Use It</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full deployment</td>
<td>Exports all PowerBuilder application objects and code, generates the corresponding Web files for the application, and deploys all Web files to the server.</td>
<td>Depends on the size and complexity of the PowerBuilder application.</td>
<td>1. The first time an application is deployed to the Web.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. After making changes to any ESQL.</td>
</tr>
<tr>
<td>Incremental deployment</td>
<td>Exports objects, re-generates and re-deploys Web files for the changed objects only. The Deployment Wizard continues even if it finds unsupported features during this process.</td>
<td>Up to 80% less time than a full deployment.</td>
<td>After making changes to an already fully-deployed application’s source code or features in the application profile.</td>
</tr>
</tbody>
</table>
Deploying Already Generated Application

| Bypasses the object exporting and file generation tasks and only deploys the Web files for the changed objects or for the entire application. | Much less time than a Full or Incremental Deployment. | When you intend to:
1) deploy the Web files for the changed objects to the server after an Incremental Deployment has been cancelled during Web deployment (Task 3).
2) refresh the set of Web files on the server after a Full Deployment has been cancelled during Web deployment (Task 3).
3) deploy the application to a different server. |

Appeon Developer will automatically set the default deployment mode by detecting whether a folder named after the application exists in the %Appeon%\Developer6.5\Projects directory on the Developer machine. If the folder is detected, the application will be regarded as an already deployed application and incremental deployment mode is selected; otherwise, full deployment mode is selected. This detection can mislead Developer into selecting incremental deployment mode, because the folder will exist in the following situations:

Table 6-6: Situations when the folder exits

<table>
<thead>
<tr>
<th>The folder will exist if you have …</th>
<th>In this case, choose …</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performed an unsupported feature analysis on the application before it has undergone a full deployment. The folder is created during analysis.</td>
<td>Incremental deployment mode.</td>
</tr>
<tr>
<td>Undeployed an Appeon deployed application from servers using the Application Undeployment Wizard. The folder is preserved.</td>
<td>Full deployment mode.</td>
</tr>
<tr>
<td>Deployed the application using a previous version of Appeon Developer. The folder is preserved unless you delete it manually. This legacy folder can be detected by the new Appeon Developer.</td>
<td>Full deployment mode.</td>
</tr>
</tbody>
</table>

6.3.3 Deploying the PowerBuilder application

The deployment process of an application contains three major deployment tasks. Different deployment modes perform different tasks. Refer to Table 6-7 for the tasks performed during each deployment mode.
Table 6-7: Different tasks performed during each deployment mode

<table>
<thead>
<tr>
<th>Deployment Task</th>
<th>Full Deployment</th>
<th>Incremental Deployment</th>
<th>Deploy Already Generated Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1: Application Source Code Export</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Exports the source code of the original PowerBuilder application.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 2: Web File Generation</td>
<td>Y</td>
<td>Y (Partial)</td>
<td>N</td>
</tr>
<tr>
<td>Analyzes unsupported features and generates Web files.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task 3: Web Deployment</td>
<td>Y</td>
<td>Y (Partial)</td>
<td>Y</td>
</tr>
<tr>
<td>Deploys generated Web files to Appeon Server and Web server.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6-2: Full application deployment

As shown in Figure 6-2, the application to be deployed is the `sales_application_demo_ax` and the deployment mode is Full Application Deployment.

The following are the complete steps in the deployment process:

Step 1 – Click Next. The Appeon Deployment Wizard begins Task 1, Application Source Code Export, as shown in Figure 6-3.
Figure 6-3: Task 1: Application Source Code Export

After Task 1 is completed, Task 2, Web File Generation, will automatically start (Figure 6-4) if the Unattended Deployment option is selected, otherwise, you must click Next at the bottom of the Appeon Deployment Wizard to begin Task 2.
After Task 2 is completed, Task 3, Web Deployment, will automatically start (Figure 6-5), or click Next at the bottom of the Appeon Deployment Wizard to begin Task 3 if the Unattended Deployment option is not selected.

Confirm that the Web Server and Appeon Server have been started and correctly configured in Appeon Developer Configuration; otherwise, deployment will fail. If deployment fails, a Retry button appears so you can retry Task 3 without running through a full deployment again.
Click the Cancel button or press the Esc key to terminate Task 3. If you cancel Task 3 during a Full Deployment, only some of the Web files will have been uploaded to the server and the application will not run correctly. To correct this, select Incremental Deployment or Deploy Already Generated Application as the deployment mode the next time you attempt to deploy the application.

Step 2 – After Task 3 has been completed, click the Next button at the bottom of the Appeon Deployment Wizard page to display the Web deployment report page, as shown in Figure 6-6.
Figure 6-6: Appeon Deployment Wizard

Deployment Information
- Application deployed: sales_application_demo
- Deployment mode: Full Application Deployment
- Deployment profile used: Local Deployment

Deployment Task Summary

<table>
<thead>
<tr>
<th>Task</th>
<th>Status</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1: Application Source Code Export</td>
<td>Completed</td>
<td>&lt; 1 Minutes</td>
</tr>
<tr>
<td>Task 2: Web File Generation</td>
<td>Completed</td>
<td>&lt; 1 Minutes</td>
</tr>
<tr>
<td>Task 3: Web Deployment</td>
<td>Completed</td>
<td>&lt; 1 Minutes</td>
</tr>
</tbody>
</table>

Total Time Elapsed for Automatic Web Conversion: < 3 Minutes

Log Information
- Warnings and Errors Summary: Error: 0, Warnings: 0
- View Log

Reports
- Features Analysis report available: Analysis Report
- Heavy window analysis report available: Heavy Window Report

Finish"
### Table 6-8: Deployment report page

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment Information</td>
<td>Displays the name of the application deployed, the deployment mode used, and the deployment profile used.</td>
</tr>
<tr>
<td>Deployment Task Summary</td>
<td>Displays the status and time of each of the three deployment tasks, as well as the total time for the entire PowerBuilder-to-Web conversion process.</td>
</tr>
<tr>
<td>Log Information</td>
<td>Displays the number of errors and warnings that occurred during the deployment process. Click the View Log button to view the errors and warnings generated during the deployment process. The log file can be managed in the Information Manager. For detailed instructions, refer to Chapter 9: <em>Using Information Manager</em>.</td>
</tr>
<tr>
<td>Reports</td>
<td>Displays the reports (Analysis Report and Heavy Window Report) generated during the deployment process (mainly Task 2):</td>
</tr>
<tr>
<td></td>
<td>• <strong>Analysis Report</strong> button – This report is available if the Generate Unsupported Feature Analysis Report option on the Appeon Deployment Wizard is selected before deployment. After you click this button, the UFA Report Window will be displayed. For detailed instructions, refer to Section 5.3: <em>Working with UFA Report</em>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Heavy Window Report</strong> button – This report is available if the Generate Heavy Window Report option on the Appeon Deployment Wizard is selected before deployment. After you click this button, the Heavy Window Report will be displayed. For detailed instructions, refer to Section 5.4: <em>Working with Heavy Window Report</em>.</td>
</tr>
</tbody>
</table>

Step 3 – Click *Finish* to close the Appeon Deployment Wizard, or click *Finish and Debug* to close the Appeon Deployment Wizard and open Appeon Debugger to debug the application, or click *Finish and Run* to close the Appeon Deployment Wizard and automatically launch the newly deployed Web application.
# 7 Debugging Appeon Web Applications

## 7.1 Overview

Sometimes a deployed application does not behave well although the original application does. In these situations, you can examine the execution results of the JavaScript code in the deployed application using Appeon Debugger.

The usage of Appeon Debugger is similar to that of PowerBuilder Debugger. You set breakpoints or monitor variables in the PowerBuilder source code, same as in PowerBuilder Debugger. The main difference is that Appeon Debugger runs the generated JavaScript code but not the PowerBuilder source code, and shows the execution results of the JavaScript code.

This chapter assumes that you are familiar with the functions of PowerBuilder Debugger; therefore it does not provide descriptions of common debugging features, but focuses on introducing the specialties of Appeon Debugger. For common debugging features, refer to the *Debugging an application* section in the *PowerBuilder User’s Guide*.

## 7.2 Important requirements

To start and use Appeon Debugger successfully, be aware of the following important prerequisites:

1. Microsoft Script Debugger must have been correctly installed and registered on the same machine and script debugging is enabled in Internet Explorer. Otherwise, you may encounter problems in starting and using Appeon Debugger. For detailed instructions and solutions, refer to the *Appeon Debugger* section in the *Appeon Troubleshooting Guide*.

2. A script in the deployed application must meet all of the following requirements to be debugged; otherwise Appeon Debugger will simply ignore it:

   - The script is deployed in the Debug JS or Debug PS/JS mode, using the local Appeon Developer, and having the Generate Object Flags for Appeon Debugger option enabled in the Appeon Deployment Wizard.
     
     If you have changed the application PBLs, or the local Web files and the remote Web files are not the same, you must re-deploy the application with the Generate Object Flags for Appeon Debugger option enabled. Or the newly-added code will not be selectable in Appeon Debugger.

   - The original PowerScript code of the script is supported by Appeon.
     
     If the code is unsupported, it is marked with a “⚠️” icon in the Appeon Debugger Source view, and cannot be set as breakpoint, nor be located with the Run To Cursor or Set Next Statement commands in the debugger.

## 7.3 Introduction to the debugging procedure

Although Appeon Debugger is used for debugging the JavaScript in the deployed application, breakpoints are not set in the JavaScript, but directly in the PowerBuilder code loaded in the Appeon Debugger Source view. The generated JavaScript is invisible throughout the entire process, however the results shown in the Watch view are the execution results of the JavaScript.
The debugger procedure with Appeon Debugger is the typical 5-step procedure as with PowerBuilder Debugger:
Step 1 – Start the debugger;
Step 2 – Set breakpoints;
Step 3 – Run the application in debug mode;
Step 4 – Examine the application at the breakpoints by monitoring variables, expressions, and call stacks. You may also step through the application after a breakpoint to closely examine the effects of each code.
Step 5 – Fix the code.

The actions performed in each step are not very different from those required in PowerBuilder Debugger. However, there are some special requirements related with the actions, and the action results are from the running of JavaScript. Continue reading the following sections for detailed descriptions of these debugging steps.

7.4 Starting Appeon Debugger

To start Appeon Debugger:
Step 1 – Close the other Appeon Developer tools.
Step 2 – Start Appeon Server and the Web server that host the Web application.
Step 3 – Click Appeon Debugger ((GUI) on the Appeon Developer toolbar.
The “Select a target” window displays, as shown in Figure 7-1.

Figure 7-1: Select an application

Step 4 – Select an application from the list and click OK.

Appeon Debugger starts and the selected application is loaded in the Appeon Debugger (Figure 7-2).

The menu bar of the debugger provides three simple menus, File, View and Help. The toolbar is almost the same as that in PowerBuilder Debugger, except that the “Start Remote” button is currently unavailable.
7.4.1 Views in Appeon Debugger

Appeon Debugger contains all the views provided by PowerBuilder Debugger except for Instances View and Objects in Memory view. Each Appeon Debugger view shows almost the same information as its counterpart in PowerBuilder Debugger. Table 7-1 summarizes the general differences between the two.

Table 7-1: Comparing the views in Appeon Debugger and PowerBuilder Debugger

<table>
<thead>
<tr>
<th>Views</th>
<th>Unique to Appeon Debugger (Unavailable in PowerBuilder Debugger)</th>
<th>Unique to PowerBuilder Debugger (Unavailable in Appeon Debugger)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>Identifies Appeon unsupported features with the icon (xico)</td>
<td>• Ancestor Script popup menu item</td>
</tr>
<tr>
<td>Source Browser</td>
<td>None</td>
<td>• Descendant Script popup menu item</td>
</tr>
<tr>
<td>Source History</td>
<td>None</td>
<td>List of system events, or the events and functions defined in parent classes, or some special events (such as Menu OnCreate event)</td>
</tr>
<tr>
<td>Variables</td>
<td>The values of variables unsupported by Appeon are marked as “unsupported”. The Global view displays all global variables used in the application, while PowerBuilder Debugger only displays instantiated global variables.</td>
<td>• Parent and Shared views</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Parent popup menu item</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Shared popup menu item</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Break on changes popup menu item</td>
</tr>
</tbody>
</table>
7.5 Setting breakpoints

7.5.1 Code lines that can be set as breakpoints

Pay attention to the following do’s and don’ts before setting breakpoints:

1. Do set breakpoints to the code lines supported by Appeon.

2. Do set simple breakpoints to script, but don’t set special breakpoints, such as breakpoints triggered when a statement has been executed a specified number of times (an occasional breakpoint), when a specified expression is true (a conditional breakpoint), or when the value of a variable changes.

3. To set a SQL statement that spans multiple lines as a breakpoint, do set the breakpoint at the first line, but don’t set breakpoints at the other lines. If you try to set a breakpoint at the other lines, the breakpoint will be set to the line that follows the SQL statement.

4. To set a PowerScript non-SQL statement, which spans multiple lines with continuation characters (&) as a breakpoint, do set the breakpoint at the last line of the statement, and if it is set at the other lines, the breakpoint icon is still added to the last line.

Note: you can set a breakpoint to the FOR statement, but the application will be suspended for only once when this statement is executed.

7.5.2 Methods for setting breakpoints

You can set breakpoints in the Source, Breakpoints, Variables, or Watch view, before or after the application starts to run in debug mode.

Method #1: Setting a breakpoint in the Source view:

Step 1 – Select the script to display in the Source View.

Select a function or event in the Source Browser view to display its script in the Source View, or right click on the Source view and select Select Script from the popup menu to select the script to be displayed.

Step 2 – Right click a line in the Source view and select Insert Breakpoints from the popup menu.

Once a breakpoint is set, a red circle displays at the beginning of the line (Figure 7-3).
Method #2: Setting a breakpoint in the Breakpoints, Variables, or Watch view:
Step 1 – Right click the Breakpoints, Variables, or Watch view;
Step 2 – Select Breakpoints from the popup menu to add a breakpoint using the Edit Breakpoints dialog.

Method #3: Removing or adding a breakpoint at runtime
When the application is already running in debug mode, you can still change the way the debug works by removing or adding breakpoints at runtime:
Double-click a line in the Source view, or select Insert Breakpoint from the popup menu in the Source view.

7.6 Running the application in debug mode
Click Start ( ) on the Appeon Debugger toolbar. The application starts to run in the Internet Explorer. It works normally until it reaches a statement containing a breakpoint. At this point it stops so that you can examine the application in the Appeon Debugger.
7.7 Examining an application at a breakpoint

Appeon Debugger provides a Quick Watch function, and the Variables, Watch and Call Stack views that work the same as PowerBuilder Debugger for examining the state of the application. It does not provide Break on Changes and TipWatch functions, nor the Object in Memory view.

7.7.1 Special variable and expression handlings

Appeon Debugger watches the variable and expression results out of JavaScript code but not PowerBuilder code. This is the main difference with PowerBuilder Debugger. Despite the difference, the results shall be mostly the same as in PowerBuilder Debugger. If the results are different, there are two possible reasons:

(1) The deployment of the application contains an error. In this case, determine the cause of the error and correct the problem.

(2) Appeon Debugger handles some variables and expressions differently from PowerBuilder Debugger. Such differences are introduced below. Be aware of them, so not be false alarmed and conduct unnecessary complex examination work.

**Appeon Debugger does NOT support the following when examining variables:**

- Expanding the values of ClassDefinition, PowerObject, and Any variables;
- Expanding the runtime values of System variables. Instead, it just expands the inherent properties of such variables.
For example, with a Window type variable w_1, PowerBuilder Debugger shows all the values of all the controls and variables contained in w_1, while Appeon Debugger only shows the values of the Window properties of w_1.

**Appeon Debugger gives different results when evaluating the following expressions:**

- It reports errors for expressions that contain unsupported code;
- If the same expression is added multiple times - in different context - into the Watch, Appeon Debugger regards it as one expression, and evaluates its value according to the context of the breakpoint;
- Appeon Debugger does not validate expressions. Therefore, if an expression does not pass validation in PowerBuilder, PowerBuilder Debugger does not give evaluation results, however Appeon Debugger may give a result, sometimes even an incomprehensible result, such as “[object]+3” for the expression “this + 3”.
- If two expressions are interconnected, because the evaluation results of the expressions are dependent on the orders they are executed, and the ordering may follow different rules in PowerBuilder Debugger and Appeon Debugger, Appeon Debugger may produce different results from PowerBuilder Debugger.

### 7.7.2 Adding variables or expressions to Watch view

**Method #1: Adding variables or expressions to Watch with the Quick Watch**

Step 1 – Select a variable or expression or place the cursor to a variable or expression in the Source view, right click and select *Quick Watch* from the popup menu or press Shift+F9.

The variable or the expression will be added to the Expression box on the Quick Watch dialog (Figure 7-5).

**Figure 7-5: Quick Watch**

Step 2 – Click the *Add Watch* button to add the variable or expression to the Watch view.

**Method #2: Adding variables or expressions directly in the Watch**
Step 1 – Right click the Watch view and select Insert from the popup menu.
Step 2 – Type valid PowerScript expression in the New Expression dialog box, and click OK to add it to the Watch view.

7.7.3 Changing the value of a variable or expression

To change the value of a variable in the Watch view:

Step 1 – Double click a variable or right click a variable and select Edit Variable. The Modify Value dialog box will be displayed.
Step 2 – In the Modify Value dialog box (Figure 7-6), type a new value for the variable in the New Value box, click OK to close the Modify Value dialog box and continue debugging the application with the variable set to the new value.

Figure 7-6: Modify value

To modify an expression in the Watch view:

Step 1 – Double-click an expression or right click an expression and select Edit Expression. The Modify Expression dialog box will be displayed.
Step 2 – In the Modify Expression dialog box, type the new expression in the New Value box and click OK.
Notice that it is unsupported to select the Null check box in the Modify Value or Modify Expression dialog to set the value of the variable or expression to null.

7.7.4 Evaluating an expression

To evaluate an expression in the Quick Watch dialog box and add it to the Watch view:

Step 1 – Change the variable or expression in the Expression box.
Step 2 – Click Reevaluate to display the new value in the tree view.
Step 3 – (Optional) Click Add Watch to add the expression to the Watch view.

7.7.5 Examining context in Call Stack view

You can examine the context of the application by clicking the code lines in the call stack.

To show a different context from the Call Stack view:

- Double-click a line or right click a line and select Set Context

Once a function is selected in the Call Stack view, the Variable views, the Source view, and the Watch view will all be updated.
7.7.6 Stepping through the application

You can use the Step In, Step Over, Step Out, Run To Cursor, and Set Next Statement commands to step through an application. Using these commands in Appeon Debugger is no different from using them in PowerBuilder Debugger.

7.8 Fixing the code/stopping the debug procedure

Once you locate the cause of the erroneous execution results, you can stop the debugging procedure, return to the PowerBuilder application to fix the code, and re-deploy the application to verify that the error has been corrected.

To stop the debugging process: close Internet Explorer where the application is running, or select Stop on the toolbar. The breakpoints you set to the application will be saved when Appeon Debugger is closed, and will be available when Appeon Debugger is re-opened.

When fixing the code, you must return to the PowerBuilder IDE and make code changes in the original PowerBuilder application. Redeploy the application and run the application in Internet Explorer to verify if the problem has been corrected. If the problem still exists and you need to return to Appeon Debugger, make sure you redeploy the application with the Generate Object Flags for Appeon Debugger option enabled. Otherwise, the latest code changes will not be reflected in Appeon Debugger.
8 Running Web Applications

8.1 Overview

Appeon Web applications are accessed via Microsoft Internet Explorer. Confirm that the Web Server(s) and Appeon Server(s) hosting the Web applications are running, and verify that the Web browser meets the necessary requirements outlined in the Appeon Installation Guide.

8.2 Requirements

8.2.1 Windows account privileges

When you run an Appeon Web application for the first time, the Xcelerator plug-in (ActiveX controls for improving the runtime performance of Web applications) must be downloaded to the client.

Different Internet Explorer versions require different privileges for installing the ActiveX control. For example, if using IE 7.0 or earlier, you must have administrative privileges while installing the ActiveX control; however, if using IE 8.0, you do not need administrative privileges.

If you are using IE 7.0 or earlier, ask your administrator to assign the current login account to the Administrators group, so that it can successfully download and register the Xcelerator plug-in. After that, you can use any account to log into Windows and run the application.

8.2.2 Internet Explorer settings

Settings in Internet Explorer can affect the deployed Web application. Sometimes outdated files that are cached in the browser can interfere with how a Web application functions and can cause errors when they are re-deployed many times. Internet Explorer’s settings may also block proxy connections or ActiveX control downloads when the Web application is opened. Therefore, you must make sure you configure Internet Explorer’s settings accordingly. Note that you may need to have the administrator rights to modify certain settings.

8.2.2.a Settings for temporary Internet files

Sometimes cached files in Internet Explorer can cause problems when accessing the converted Web application after deployment. In order to avoid the risk of providing “stale” data, it is imperative that Internet Explorer has the cache options set to Automatically and that there is sufficient disk space on the Client to enable the Web files to be cached.

To set the cache options:

Step 1 – Open Internet Explorer and select Tools | Internet Options, as shown in Figure 8-1.
Step 2 – Click the *Settings* button in the *Temporary Internet Files* group box.

Step 3 – Select the radio button to *Automatically* check for newer versions of stored pages, as shown in Figure 8-2.
Figure 8-2: Temporary Internet File Settings

Step 4 – Verify that the Disk space to use scroll box is set to at least 200 MB. Click OK in the Settings window.

8.2.2.b Advanced settings for temporary Internet Explorer files folder

During development, the “Temporary Internet Files folder” should be set to automatically empty each time Internet Explorer is closed. This will force Internet Explorer to re-download the entire application every time it is opened, to ensure the latest version of the application is present, and to prevent interference from outdated cached files.

Here are the necessary steps required to configure Internet Explorer to flush its temporary files each time the browser is closed:

Step 1 – Click the Advanced tab in the Internet Options window, as shown in Figure 8-3.

Step 2 – Check the Empty Temporary Internet files folder when browser is closed check box under the Security section, as shown in Figure 8-3.
Figure 8-3: Empty Temporary Internet Files folder when browser is closed

Step 3 – Click Apply in the Advanced tab and then OK in the Internet Options window to allow the settings to take effect.

Allowing the temporary Internet files folder to be emptied each time the browser is closed, is recommended only during the development stage. Once your Web application is ready for production deployment, and frequent changes are not being made to the application, this setting can be disabled once again (unchecked). This setting should be reset once development is complete so the Web application can be cached in each Client PC for better Client-side performance. Using this set of guidelines, caching files should be enabled when they are needed for performance improvement, and disabled during the development and debugging stages.

8.2.2.c Advanced settings for proxy server

If the Client visits the Web application through a proxy server, you must select the Use HTTP 1.1 through proxy connections option under the HTTP 1.1 settings section, as shown in Figure 8-4.

The following steps detail the configuration needed in Internet Explorer:

Step 1 – Click the Advanced tab in the Internet Options window, as shown in Figure 8-4.

Step 2 – Check the Use HTTP 1.1 through proxy connections check box under the HTTP 1.1 settings section, as shown in Figure 8-4.
Figure 8-4: Proxy connection settings

Step 3 – Click Apply in the Advanced tab and then OK in the Internet Options window to allow the settings to take effect.

8.2.2.d Security settings for ActiveX controls and Appeon Xcelerator plug-in download

If the Web application calls OLE objects, DLL files, or runs executable programs on the Client, then when the Web application is run for the first time on the Client the user will need to download an ActiveX control (for executing functions that involve client-side integration) and the Appeon Xcelerator plug-in (for improving the runtime performance of Web applications) from the Web Server. To ensure that Internet Explorer will not block the download, configure Internet Explorer’s settings, using the following steps:

Step 1 – Open Internet Explorer and select Tools | Internet Options | Security tab.

Step 2 – Verify that you selected an appropriate Web content zone. Appeon recommends you select the Trusted sites zone and add the Web application URL to the Trusted sites.
Step 3 – Click the **Custom Level**... button in the **Security** tab. The Security Settings page is displayed, as shown in Figure 8-6.
Figure 8-6: Security Settings page

Step 4 – Check the Enable or the Prompt (recommended) radio button for the following options:

- Download signed ActiveX controls and plug-ins (and Download unsigned ActiveX controls in Windows XP)
- Initialize and script ActiveX controls not marked as safe
- Run ActiveX controls and plug-ins
- Script ActiveX controls marked safe for scripting
- File download
- Active scripting
- Allow active content to run in files on My Computer (If the version of Internet Explorer is 6.0 SP2)

If Enable is selected, the ActiveX control will download automatically without prompting. If Prompt is selected, your approval will be required before downloading takes place.

Note: If running the Appeon Web application for the first time in Internet Explorer 8 on Windows XP, be sure to set “Download signed ActiveX control” to Enable (Do not set to Prompt or Disable), otherwise, Internet Explorer will crash.
8.2.3 Language setting requirements

The language of the data in the database used by the application should match the operating system language settings on the machine where Appeon Server is installed. If the language settings conflict, problems will occur when the Web application is run.

Verify that the language of the data in the database used by the Appeon-deployed Web application matches the language settings of the client operating system:

Step 1 – Select Windows Start Menu | Settings | Control Panel.
Step 2 – Click Regional Options, as shown in Figure 8-7.

Figure 8-7: Regional Options

Step 3 – Verify that the language selected in the “Your locale (location)” list box and the language checked as default in “Language settings for the system” settings are the same. The language can be from different regions as long as it is the same language.

This configuration is not required for English language applications.

8.2.4 Disabling anti pop-up software

If anti-popup software (example: Popup Stopper by Panicware) is installed and is active on the Client machine, it will not allow any browser pop-ups to occur. This can cause certain
operations to become disabled in the converted Web applications such as the enhanced Web features detailed below, or other operations that automatically load an Internet Explorer window or popup. To prevent any problems, it is recommended that users disable any anti-popup software while using the converted Web applications.

8.3 Accessing Web applications

8.3.1 URLs for Web applications

A typical URL for a Web application is similar to one of the following:
http://192.0.0.80:8080/appeondemo/
http://192.0.0.80:8080/appeondemo/x32_index.htm
http://192.0.0.80:8080/appeondemo/x64_index.htm

A complete Web application URL consists of four parts:

Table 8-1: Web URL

<table>
<thead>
<tr>
<th>Web URL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>Uses “http://” typically. If your Web Server is configured as an SSL Web Server, use “https://”</td>
</tr>
<tr>
<td>IP address or</td>
<td>Uses the IP address or domain name of the Web Server. Using “localhost” listeners in a production environment is not recommended, although the Appeon installation program has created an HTTP listener (localhost:9988) for EAServer\Appeon Server.</td>
</tr>
<tr>
<td>domain name</td>
<td></td>
</tr>
<tr>
<td>HTTP port</td>
<td>(Optional) Specifies the port number that your Web Server accepts for HTTP connections. If it is not specified, the default port is 80. If several Web Servers are running on the same machine, make sure that the port specified is a port of the Appeon Web Server.</td>
</tr>
<tr>
<td>Web file path</td>
<td>Specifies the path of the entry page (x32_index.htm for 32-bit system, or x64_index.htm for 64-bit system) of the Web application, relative to the home directory of the Web server. It is the path set for the Web application URL in Application Profiles, for example, “demo/” or “demo/x32_index.htm” (x32_index.htm can be omitted).</td>
</tr>
</tbody>
</table>

8.3.2 Running Web applications

There are two methods to run a Web application:

- **Typing the URL of the Web application in Internet Explorer address bar**: You can always use this method to access the application, on the condition that the computer meets the requirements in Section 8.2: Requirements.

- **Launching the Web application from the Appeon Developer Run button**: Run is a shortcut to Appeon applications that are deployed to the default Appeon Server and Web Server.

This section explains how to launch a Web application from the Appeon Developer Run button.
8.3.2.a Launching Web applications from Run button

You can launch the Web application by clicking the Run button in the Appeon Developer toolbar. Appeon Developer automatically remembers the URLs of the Web applications that are deployed to the default Appeon Server and Web Server.

To run a Web application from the Appeon Developer toolbar:

Step 1 – Verify that the default Appeon Server and Web Server in the Server Profiles settings have been started.

Step 2 – Click the Run button ( ) on the Appeon Developer toolbar.

If more than one application is deployed, the Run Web Application window is displayed as shown in Figure 8-8, prompting you to select which application to run.

Figure 8-8: Run Web Application window

Only applications deployed to the Default Appeon Server and Default Web Server are listed in the Run Web Application window.

If you have deployed an application to several different Web file folders on the same Web Server, Appeon Developer will only remember the URL used in the last deployment, although the URLs to access the other Web file folders used in the previous deployments will also be valid.

Step 3 – Select a Web application from the list and click Run.

8.3.2.b Selecting a run mode

When you run a Web application for the first time, a run mode selection page will be displayed, if you have enabled the “Allow user to select run mode” option in AEM (see Appeon Server Configuration Guide for details).
On the run mode page, you can select one of the following run modes (Figure 8-9):

- Run Now – The application runs immediately and files will be downloaded only when used. This mode is recommended for high speed network.
- Run After Download – Files will be downloaded before the application runs. This mode is recommended for low speed network.

You can also set a run mode to be the default mode in AEM (see Appeon Server Configuration Guide for details).

Figure 8-9: Run Mode window

8.4 Appeon DataWindow menu

Appeon provides a Web enhancement, the Appeon DataWindow menu, for all DataWindows deployed with Appeon. When right-clicking on a Web DataWindow, you can access the Appeon DataWindow popup menu: Find, Find Next, Find Previous, Sort, Filter, Print, and Save As File..., as shown in the following figure.
8.4.1 Enabling Appeon DataWindow menu

To use the Appeon DataWindow menu, you must enable it by using two PowerBuilder global functions: `appeonpopmenu` and `appeonpopmenuon` that are defined in `appeon_workarounds.pbl`. For detailed instructions on how to use `appeonpopmenu` and `appeonpopmenuon` functions in a DataWindow, refer to *Appeon Workarounds Guide* | *Appeon Workarounds PBL Reference* | *Appeon Client Functions*.

8.4.2 Using Appeon DataWindow menu

Appeon DataWindows menu have the following features: Find, Find Next, Find Previous, Sort, Filter, Print, and Save As File. Perform the following steps to use the menu in the Web DataWindows.
8.4.2.a Find

The user can search for data using the Find, Find Next and Find Previous functions.

Step 1 – Right-click the Web DataWindow and select *Find* from the popup menu, as shown in Figure 8-11.

**Figure 8-11: Appeon DataWindow Menu**

![DataWindow Menu](image)

Step 2 – Select the column and search scope and input criteria for the search, as shown in Figure 8-12.

**Figure 8-12: Find**

![Find](image)

Step 3 – The first row that matches the search criteria will be highlighted.

Step 4 – Right-click and select *Find Previous* or *Find Next* to highlight the rows (one at a time) that match the criteria.

8.4.2.b Sort and filter

The user can sort data by the column in ascending or descending order, or filter data using a number of functions.
To sort data:
Step 1 – Right-click the Web DataWindow and select *Sort* from the popup menu, as shown in Figure 8-13.

**Figure 8-13: Sort**

![DataWindow](image)

Step 2 – Specify the column used to sort the data in the DataWindow by double clicking on the column name in the left list box. The column name appears on the right, as shown in Figure 8-14.

**Figure 8-14: Specify sort column**

![Specify Sort Columns](image)

Step 3 – Check the *Ascending* check box to sort the data in ascending order or uncheck the *Ascending* check box to sort the data in descending order. Click *OK* to accept the changes. The DataWindow automatically refreshes and the data is sorted.
If you want to edit the selected column, double click it to open the “Modify Expression” dialog, as shown in Figure 8-15.

**Figure 8-15: Modify Expression dialog**

To filter data:

Step 1 – Right-click the Web DataWindow and select *Filter* from the popup menu, as shown in Figure 8-16.

**Figure 8-16: Filter**

Step 2 – Specify the filter by selecting the functions and columns. Click *Verify* to test whether the filter expression is valid. Click *OK* to accept the filter, as shown in Figure 8-17. The DataWindow automatically refreshes and the data is filtered.
8.4.2.c DataWindow printing

The Print menu provides user ability to print DataWindows directly on printers connected to the Client.

Step 1 – Right-click the DataWindow and select “Print” from the popup menu, as shown in Figure 8-18.

Figure 8-18: Print file with physical printers

Step 2 – Specify the printer settings and paper settings in the Print Setup window, as shown in Figure 8-19. Click OK.
The DataWindow will be printed directly to the local printer.

**Figure 8-19: Print setup window**

![Print Setup Window](image)

### 8.4.2.d DataWindow SaveAs

The user can save the contents of a DataWindow in different file formats.

Step 1 – Right-click on a DataWindow and select “Save As File…” from the popup menu, as shown in Figure 8-20.

**Figure 8-20: Save As File…**

![Save As File Window](image)

Step 2 – Specify the file location, file name and file type in the Save As window, as shown in Figure 8-21.
8.4.2.e Additional enhanced features

In addition to the popup menu (Find, Find Next, Find Previous, Sort, Filter, Print, Save As File), Appeon offers more features for users to manage DataWindows on the Web. Users can place DataWindow columns in a different order by dragging and dropping them to new positions. When navigating an editable DataWindow, users can use the popup menu to quickly edit the contents. These features are always available and require no extra coding to be enabled. The following steps show you how to access them:

Step 1 – Select a column header in the Web DataWindow and the selected location is highlighted, as shown in Figure 8-22.

Step 2 – Drag and drop it to the desired location, and the location to be dropped is highlighted as well.

Figure 8-22: Select a column header
Step 3 – Right click on an item in an editable DataWindow and a menu pops up, as shown in Figure 8-23.

The user can select the following menu items from the popup menu to perform quick editing: Undo, Cut, Copy, Paste, Delete, Select All.

Figure 8-23: Popup menu for quick edition
9 Using Information Manager

9.1 Overview

Information Manager provides easy access to all the available reports or log files generated during the application migration process, including debug, development, and deployment. The reports are categorized as follows: Analysis and Performance. The log files are categorized as follows: Deploy Log and Analysis Log.

Table 9-1 provides a brief description of all the reports generated during the Web migration process and the report types they are associated with.

Table 9-1: Report Description

<table>
<thead>
<tr>
<th>Report Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis</td>
<td><strong>Analysis Report</strong> is generated during a feature analysis or application deployment, and records the unsupported features in the application. After you click this button, the UFA Report Window will be displayed. For detailed instructions, refer to Section 5.3: Working with UFA Report.</td>
</tr>
<tr>
<td>Performance</td>
<td><strong>Heavy Window Report</strong> is generated during a feature analysis or application deployment, and records the complex windows in the application. For detailed instructions, refer to Section 5.4: Working with Heavy Window Report.</td>
</tr>
</tbody>
</table>

Table 9-2 provides a brief description of all the log files generated during the Web migration process, and the log types they are associated with.

Table 9-2: Log Type Description

<table>
<thead>
<tr>
<th>Log Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deploy Log</td>
<td><strong>Deploy Log</strong> is generated during application deployment, and records the deployment process status, such as errors and warnings. All log files are named according to the following format: Log Type (i.e. “DeployWizardLog”, or “FeatureAnalysisLog”) + time when the log file is generated (yyyy/mm/dd/hh/mm/ss) + “xml”. For example, DeployWizardLog_20050220101918.xml.</td>
</tr>
<tr>
<td>Analysis Log</td>
<td><strong>Analysis Log</strong> is generated during the feature analysis process, and records the analyzing status.</td>
</tr>
</tbody>
</table>

9.2 Viewing reports and logs

Step 1 – Click the *Information* button ( Image) on the Appeon Developer toolbar and the Information Manager window is displayed, as shown in Figure 9-1.
Figure 9-1: Information Manager

Step 2 – Select the application in the top dropdown listbox.

Step 3 – Select a report type or log type in the left box and the corresponding report button or log files are displayed in the right box.

If there is more than one log file generated, the log files are listed in order according to the date and time of generation; the latest log will be listed at the top.

You can open one log at a time, or open several logs at one time: select a continuous range of log files by holding down Shift and click log files, or select a disconnected range of log files by holding down Ctrl and clicking log files. Then click View to open the selected log files.

The View and Delete buttons will be disabled if no log files were generated.

Note: For Deploy Log, you can click View All to view all information such as errors, warnings, and process description, or click View Warning to view warnings only, or click View Errors to view errors only.

Step 4 – Click the report button or log file to view the report or log.

You can continue with other operations while the Information Manager window opens. For example, you can perform feature analysis, deploy the application, or run the Web application. If any new reports or logs are generated when the Information Manager window is still open, the new reports or logs are not reflected in the Information Manager immediately. You must refresh Information Manager in either of the following ways:

- Close the open Information Manager window and open it again, or
• Select a different application and then select the original application again in the open Information Manager window.
10 Packaging and Installing Web Applications

10.1 Overview
The Appeon Web Application Package Wizard helps PowerBuilder developers to generate portable installation “packages” for their Appeon deployed Web applications. With the installation package, the user can directly install the Web application to servers without installing Appeon Developer and using it to do Web deployments; this will save time. The PowerBuilder developers do not need to provide the source PowerBuilder application to the user; this protects the author’s intellectual property rights.

DO NOT try to copy the generated Web files from one Appeon Server or Web server to another Appeon Server or Web server simply by copying and pasting the files. This copy and paste method cannot guarantee that the Web application will work. If you need to distribute an Appeon Web application to multiple Appeon Servers and Web servers located in different network areas, instead of installing Appeon Developer in each area and performing Web deployments, Appeon recommends that you:

1. Deploy the target PowerBuilder application using the Appeon Deployment Wizard (See Chapter 6: Deploying PowerBuilder Applications) and
2. Pack all the application files using the Appeon Web Application Package Wizard (See Section 10.2: Packaging a Web application), then
3. Copy the installation package to a machine running Windows and run the installation package to install the application to any number of target servers located in the same network area. (See Section 10.3: Installing and uninstalling a Web application). Note that the generated installation packages run on the Windows platform only.

You can customize the installation/uninstallation wizard provided by Appeon to suit your needs. (See the Web_App_Install_Help.chm help file under the help folder of the generated package.)

10.2 Packaging a Web application

10.2.1 What can be packaged?
The package wizard allows you to pack the following files and settings:

- **Mandatory** Web application files
  Include all Web files, DataWindow syntax, INI files, image files, DLL/OCX files etc.

- **Mandatory** application profile, Appeon Server profile, and Web server profile settings
  Include the profile name, Web URL, PowerBuilder version, file encryption, performance settings, server type, Appeon Server version, Appeon Server and Web server connection settings, AEM, connection caches etc. All these settings will be saved to the configuration files in the INI folder of the generated package. Of these configuration files, you can modify the config.xml file using the config.exe tool. For detailed information, refer to the Modifying the deploy-config file section.

- Optional EAServer components
When the Appeon Server is of type EAServer, you can package the local or remote EAServer components used by the application. Note that EAServer components work with EAServer application server only.

10.2.2 Important requirements

Before using the Appeon Web Application Package Wizard, verify that

1. The target PowerBuilder application has been deployed as required:
   - You must have performed a **full deployment** on the target PowerBuilder application on the **same local Developer machine** where you will use the package wizard to pack this application later, this will generate a complete set of files on the local Developer machine where the package wizard will be able to obtain and pack files.

2. The deployed application can run correctly.

10.2.3 Packaging instructions

To package a Web application and the related settings, take the following steps:

Step 1 – Start the servers, as the package wizard will pack settings of Appeon Server, AEM and NVOs.

Step 2 – Click the **Package** button on the Appeon Developer toolbar.

The Appeon Web Application Package Wizard is displayed, as shown in Figure 10-1. Click **Next** to proceed.

Figure 10-1: Welcome page
Step 3 – Select the profile of the application that you want to package (Figure 10-2).
Select the profile that is identified with “Yes” in the Deployed column and click Next.

**Figure 10-2: Select an application**

![AppIcon:Select an application](image)

Table 10-1 gives a brief introduction of the columns:

**Table 10-1: Web Application Package Wizard**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected</td>
<td>Identical to the default application profile selected in the Appeon Developer Configuration window.</td>
</tr>
<tr>
<td>Profile Name</td>
<td>All the application profiles configured in the Appeon Developer Configuration window are listed in the order that the application profiles are created.</td>
</tr>
<tr>
<td>Mode</td>
<td>Identical to the current mode the application profile is set to in the Appeon Developer Configuration window. Note that as the package wizard indicates, packaged applications should be available in <strong>Release</strong> mode. This mode prevents most forms of reverse engineering, and like application packaging, it protects the author’s intellectual property.</td>
</tr>
<tr>
<td>Deployed</td>
<td>Indicates whether the application has been deployed using the local Appeon Developer. Only applications that have been deployed using the local Appeon Developer can be packaged.</td>
</tr>
</tbody>
</table>

Step 4 – Select the Appeon Server whose settings will be packaged and click **Next**.
The Appeon Server settings, such as the server type and the Appeon Server version, will be saved to the config.xml file in the INI folder of the generated package. If necessary, modify this file using the config.exe tool after the packaging is complete (refer to Modifying the deploy-config file).

All the configured Appeon Servers selected into the default deployment profile in the Appeon Developer Configuration window are listed, as shown in Figure 10-3.

The selected Appeon Server determines that its AEM settings will be packaged and that the packaged application can only be installed to Appeon Servers of the same type and version.

**Figure 10-3: Select an Appeon Server**

Step 5 – If the selected Appeon Server is of type EAServer, select whether to package EAServer components, as shown in Figure 10-4.
If EAServer components are used in the application, select the Package EAServer Components option and then create connection profiles for EAServers which host the components. Follow instructions below to create EAServer profiles:

- Click Add to create an EAServer profile.
- In the Edit EAServer Profile dialog box (Figure 10-5), input the following connection parameters: profile name, host name or IP address, port number, user name and password.
- Click Test to ensure that the connection is successful.
- Click OK to save the new EAServer profile.
Step 6 – If the *Package EAServer Components* option is selected in the previous step, select which components to package and click *Next*, as shown in Figure 10-6.

The selected NVOs will be packaged as *ComponentName*.jar and saved under the /NVO/server profile name/package name folder.

Figure 10-6: Select NVOs
Step 7 – Select whether AEM settings and connection caches for the current application will be packaged and click Next, as shown in Figure 10-7.

The AEM settings will be saved in the AEM.xml file and connection caches will be saved in the config.xml file.

Figure 10-7: Select AEM settings and connection caches

Step 8 – Specify the destination folder for the installation package and click Next, as shown in Figure 10-8.
Figure 10-8: Specify destination location for the package

The packaging tool will create a folder named "Application Profile Name_install" to store the packaged application.

Where do you want to place the folder?

Application Profile Name: appeon_code_examples

Specify the directory where the install folder will reside:

C:\Program Files\Appeon\WebAppPackages\  
Browse...

Step 9 – Wait while the wizard is packing files and settings, as shown in Figure 10-9.
Figure 10-9: Web application packaging status window

Step 10 – Once the operation is complete, click Finish to exit the package wizard (Figure 10-10).

To view the log information generated during the packaging process, select the View Package Log box and then click Finish. The log file will be displayed.
After the application is successfully packaged, all the necessary application files will be packaged and generated in the specified destination folder. You can install the application to any Appeon Servers and Web Servers.

10.2.4 Modifying the deploy-config file

The deploy-config file (config.xml) is created during the packaging process and includes the default settings of the application: application profile, Appeon Server profile, Web server profile, EAServer profile, EAServer components, and connection caches. You can modify the config.xml file using the config.exe tool to have identical settings of the target environment, so that the user can run the installation very conveniently without needing to specify them.

When you run the config.exe file in the generated package, an XML File Configuration dialog is displayed and loads the Web URL setting, Appeon Server settings, and Web server settings from the config.xml file. You can modify these settings on the corresponding tab pages and then click OK to save them to the config.xml file. The instructions to modify these settings are the same as to modify in the Appeon Developer Configuration window (see Using Configure tool for detailed instructions).
Figure 10-11: Configuring the XML file

10.3 Installing and uninstalling a Web application

10.3.1 Points to note before installation

Be aware of the following points before installing an Appeon application:

- The Setup program of the Web application runs on the Windows platform only. To install applications to servers running on Unix\Linux, you must run the Setup program on the Windows platform and then install the application to remote servers running on Unix\Linux.
- The target Web servers must have the Appeon Server Web Component correctly installed.
- The packaged EAserver components, if any, can only be installed to EAserver application server.

10.3.2 Installing an application

You can copy the generated package to any computer running Windows and install the application to any number of Appeon Servers and Web servers. Installing an application using the generated package is the same as deploying an application using the Appeon Deployment Wizard.
Step 1 – Start the target Appeon Server(s) and Web server(s) where you want to install the application.

Verify that the target Web servers already have Appeon Server Web Component installed.

Step 2 – Run the Setup.exe file in the generated package.

The Appeon Web Application Deployment wizard is displayed, as shown in Figure 10-12. Click Next to proceed.

Figure 10-12: Welcome page

Step 3 – Select the deploy-config file (config.xml) to ease your configuration of the installation wizard. The installation wizard will use the settings in the config.xml file.

If you do not want to use the deploy-config file, simply skip this option and click Next to configure the settings step by step.
Step 4 – Input the Application Profile Name and the Web URL for the application and click Next.

The Application Profile Name is used as the application name to identify an application. If you want to install multiple instances of the same application on the same server, you can run the setup package repeatedly and specify different Application Profile Name here. For example, input `appeon_code_examples_test` as the Application Profile Name in the first installation, and input `appeon_code_examples_production` as the Application Profile Name in the second installation. The Application Profile Name will be used as the application name to distinguish the multiple application instances on the same server, so they can be run independently from each other.
Figure 10-14: Specify Application Profile Name and Web URL

Please specify the profile name and the Web URL for the application.

Please specify the Web URL for the application.

Application Profile Name: appeon_code_examples

Web URL: http://web.server:port/ appeon_code_examples

Step 5 – Configure and select profiles for the Appeon Server(s) and the Web server(s) where the application will be installed.

The application can only be installed to Appeon Servers of the same type and version as the selected Appeon Server during packaging.
For detailed instructions on how to create the Appeon Server profile and Web server profile, refer to Section 4.3.3: *Managing server profiles*.

Step 6 – Select whether to deploy EAServer components.

After NVO components are deployed, you must generate stub/skeleton in EAServer. Detailed instructions are provided in the *Appeon Migration Guide*. 
If EAServer components are used in the application, select the **EAServer Components** option and then create connection profiles for EAServers where you want to deploy the components. Follow instructions below to create EAServer profiles:

- Click **Add** to open the Edit EAServer Profile dialog box (Figure 10-30).
- Input the connection parameters: profile name, host name or IP address, port number, user name and password.
- Click **Test** to verify the connection and then click **OK** to finish.

**Step 7** – If the **EAServer Components** option is selected in the previous step, select the EAServer profile from the list and the components to deploy. Click **Next** to proceed.
Step 8 – Select whether to install AEM settings and connection caches for the current application and click *Next*.

**Figure 10-19: Deploy AEM settings**
Step 9 – Specify the destination folder for the undeployment tool and the log file and click Next, as shown in Figure 10-20.

**Figure 10-20: Specify location for the Appeon Undeployment Tool**

Step 10 – Wait while the wizard is installing files and settings, as shown in Figure 10-21.
Step 11 – Once the operation is complete, click *Finish* to exit the installation wizard. To run the Web application immediately, select the *Run Web Application Now* option and click *Finish*. Or you can run the application later from the Windows Start | Programs | Appeon Web Application | *ApplicationName*.

To view the log information generated during the installation process, select the *View Log* box and then click *Finish*. The log file will be displayed.
10.3.3 Uninstalling an application

Step 1 – Start the target Appeon Server(s) and/or Web server(s) where you intend to uninstall the application.

Step 2 – Select Windows Start | Programs | Appeon Web Application | Undeploy ApplicationName.

The Appeon Web Application Undeployment wizard is displayed, as shown in Figure 10-23. Click Next to proceed.
Step 3 – Select an existing deploy-config file to automatically use the configuration previously used for the deployment wizard.

Figure 10-24: Select the configuration file

- The wizard automatically loads the configuration file for the current package and uses the settings in this file as the default settings for the subsequent steps. You can modify the settings in this file or select another configuration file.
- Note: You can clear the configuration file path in the Destination File box and click Next to configure the settings manually.

<table>
<thead>
<tr>
<th>Destination File</th>
<th>Browse...</th>
</tr>
</thead>
<tbody>
<tr>
<td>\Appeon\WebApp\Packaging\appname_code-examples_install\xml\config.xml</td>
<td></td>
</tr>
</tbody>
</table>
Step 4 – Specify the Web URL.

**Figure 10-25: Specify Web URL**

```
Specify Undeployment Settings

Please input the undeployment settings.

Please specify the Web URL for the application to undeploy.

Application Profile Name: appeon_code_examples

Web URL

http://web server:port/appeon_code_examples
```

Step 5 – Select the Appeon Server(s) and Web server(s) where application will be undeployed.

To create the Appeon Server profile and Web server profile, refer to Section 4.3.3: *Managing server profiles*. 
Figure 10-26: Specify server settings

Step 6 – Select whether to undeploy EAServer components.

Figure 10-27: Undeploy EAServer components
Select the *EAServer Components* option and then create connection profiles for EAServers where you want to undeploy the components. Follow instructions below to create EAServer profiles:

- Click *Add* to open the Edit EAServer Profile dialog box (Figure 10-28).
- Input the connection parameters: profile name, host name or IP address, port number, user name and password.
- Click *Test* to verify the connection and then click *OK* to finish.

**Figure 10-28: Edit EAServer profile**

![Edit EAServer profile dialog box](image)

---

Step 7 – If the *EAServer Components* option is selected in the previous step, select the EAServer profile from the list and the components to undeploy. Click *Next* to proceed.

**Figure 10-29: Select EAServer components**

![Select EAServer components](image)
Step 8 – Wait while the wizard is removing files and settings.

Step 9 – Once the operation is complete, click *Finish* to exit the undeployment wizard.

To view the log information generated during the uninstall process, select the *View Log* box and then click *Finish*. The log file will be automatically opened.

**Figure 10-30: Undeployment complete**
11 Undeploying Web Applications

11.1 Overview
The Application Undeployment Wizard removes a deployed application from the associated Appeon Server(s) and Web Server(s), including all DataWindows from the Appeon Server(s), all Web files from the Web Server(s), and all Transaction Object mappings from AEM. This is the only way that an Appeon deployed application should be removed. Any other method may not fully remove the application, and may cause errors.

11.2 Undeploying instructions
Step 1 – Verify that the Appeon Server and the Web Server(s) hosting the Web application to be undeployed are running before you proceed with undeployment.

Step 2 – Click the Undeploy button (×) in the Appeon Developer toolbar. The Application Undeployment Wizard is displayed, as shown in Figure 11-1.

Figure 11-1: Application Undeployment Wizard

Step 3 – Click Next to continue. The Specify Undeployment Mode window is displayed, as shown in Figure 11-2, prompting you to select an undeployment mode.
Undeploying Web Applications

Figure 11-2: Specify Undeployment Mode window

Table 11-1 shows when and how to use each mode.

Table 11-1: Undeployment modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>When To Use It</th>
<th>How To Use It</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment Profile mode</td>
<td>If you clearly know which application you want to delete from which server.</td>
<td>Select an application to be undeployed and its associated deployment profile.</td>
</tr>
<tr>
<td>Appeon Server mode</td>
<td>When you are uncertain of the name of the application you want to undeploy, or an application profile does not exist for the application you want to undeploy, but you clearly know the Appeon Server hosting the Web application intended for undeployment.</td>
<td>Specify an Appeon Server by selecting an Appeon Server profile. The selected Appeon Server will refer to all the Web applications that are deployed. Then you can choose one Web application for undeployment.</td>
</tr>
</tbody>
</table>

11.2.1 Undeploying with the Deployment Profile mode

Step 1 – Select the Deployment Profile radio button in the Specify Undeployment Mode window as shown in Figure 11-2, and click Next to continue.

Step 2 – Select a Web application from the Application list box, as shown in Figure 11-3.

The Application list box lists all the application profile names. Be sure to choose one that has been deployed and is intended for undeployment. The application and the deployment profile used in the last deployment will be selected by default.
Step 3 – Select the deployment profile that is used to deploy the Web application from the Deployment Profile list box, as shown in Figure 11-3. Click Next to continue.

Figure 11-3: Deployment Profile Mode window

![Deployment Profile Mode window](image)

Step 4 – Select the Appeon Server(s) or Web Server(s) that are intended for undeployment, as shown in Figure 11-4. The Appeon Server(s) and Web Server(s) defined in the deployment profile selected in the previous step are listed. Click Next to continue.

Figure 11-4: Confirm Appeon Servers and Web Servers

![Confirm Appeon Servers and Web Servers window](image)
Step 5 – Click Undeploy to confirm the undeployment settings and start the undeployment process, as shown in Figure 11-5.

Figure 11-5: Confirm undeployment settings

The undeployment process begins, as shown in Figure 11-6.

Figure 11-6: Undeployment in Progress

Step 6 – Click Finish to close the dialog box, as shown in Figure 11-6.
11.2.2 Undeploying with the Appeon Server mode

Step 1 – Select the **Appeon Server** radio button in the Specify Undeployment Mode window as shown in Figure 11-7, and click **Next** to continue.

**Figure 11-7: Specify Undeployment Mode window**

Step 2 – Select the Appeon Server that contains the Web application to be undeployed from the Appeon Server list box, as shown in Figure 11-8.

The Appeon Server list box lists all the Appeon Server profile names. Make sure the selected Appeon Server is running.

Step 3 – Select the application to be undeployed from the Application list box and click **Next**, as shown in Figure 11-8.

The Application list box lists all the Web applications deployed to the selected Appeon Server.
Step 4 – Select the Appeon Server(s) or Web Server(s) that are intended for undeployment, as shown in Figure 11-9. The Appeon Server(s) and Web Server(s) defined in the default deployment profile are listed. Click Next to continue.

Figure 11-9: Ready to Undeploy Application window

Step 5 – Click Undeploy to confirm the undeployment settings and start the undeployment process, as shown in Figure 11-10.
The undeployment process begins, as shown in Figure 11-11.

Step 6 – Click Finish to close the dialog box, as shown in Figure 11-11.
12 Developing with Code Insight

12.1 Overview

Code Insight is a tool designed to help you write Appeon-supported PowerScript code more quickly by providing a lookup and paste service inside the PowerBuilder Script view. You can use it to develop PowerBuilder applications that are free of Appeon unsupported features and tailored to Web migration.

The usage of Code Insight is similar to that of PowerBuilder AutoScript. Table 12-1 describes the differences between the PowerBuilder AutoScript and Appeon Code Insight.

Table 12-1: Comparison between AutoScript and Code Insight

<table>
<thead>
<tr>
<th>Effective for</th>
<th>PowerBuilder AutoScript</th>
<th>Appeon Code Insight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functions, events, variables, properties, and templates for PowerBuilder DO, FOR, IF, and CHOOSE statements.</td>
<td>Items referred in dot notations</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>What pops up</th>
<th>A list of properties, variables, methods, or statements.</th>
<th>1) A list of properties, variables, or methods. 2) The Appeon unsupported properties, variables or methods that are marked with a red icon left to them.</th>
</tr>
</thead>
</table>

| Ways to use it | Two ways to use AutoScript:  
- Turn automatic popup on to pop up a list automatically when you pause while typing, or  
- Invoke AutoScript only once when you need it by selecting the menu items. | Only one way to use Code Insight:  
Step 1: Activate Code Insight;  
Step 2: Configure the default PBT file;  
Step 3: Enable Code Insight.  
Refer to the following sections for detailed instructions. |
|---------------|----------------------------------------------------------|-------------------------------------------------------------------|

<table>
<thead>
<tr>
<th>Priority</th>
<th>When Code Insight is activated, AutoScript will be automatically turned off.</th>
</tr>
</thead>
</table>

12.2 Activating Code Insight

Click the Code Insight button ( ) in the Appeon Developer toolbar to launch Code Insight.

At the click of the Code Insight button, an icon will appear in the status area of the task bar. Code Insight has five status icons:

Table 12-2: Status of Code Insight

<table>
<thead>
<tr>
<th>Icon</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🚫</td>
<td>Disabled</td>
<td>Code Insight is disabled.</td>
</tr>
<tr>
<td>🔄</td>
<td>Loading</td>
<td>Code Insight is analyzing the default PBT and generating the supported and unsupported feature list based on the analysis result.</td>
</tr>
<tr>
<td>⚪️</td>
<td>Off</td>
<td>Code Insight is enabled but ineffective.</td>
</tr>
</tbody>
</table>
Developing with Code Insight

### 12.2.1 Configuring Code Insight

Before you can enable Code Insight for a PowerBuilder application, you need to configure the PBT file of the desired application as the default PBT file. You can also specify the Appeon feature sets and the feature types for which feature list will be displayed.

Step 1 – Click on the icon of Code Insight, and the status-area-icon menu pops up with three items, Enable (or Disable), Config, and Quit, as shown in Figure 12-1.

![Figure 12-1: The status-area-icon menu](image)

Step 2 – Click **Config** to open the Code Insight Configuration window, as shown in Figure 12-2.

![Figure 12-2: Code Insight Configuration window](image)

The Code Insight Configuration window contains two tab pages where you can manage the PBT files and specify what to display in the list.
### Table 12-3: Code Insight Configuration window

<table>
<thead>
<tr>
<th>Select this tab</th>
<th>To do this</th>
<th>Detailed Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBT File tab page</td>
<td>Add the PBT files.</td>
<td>1) Click <em>Add</em>.  2) Search and select the PBT file in the common dialog box.  3) Click <em>Open</em> in the common dialog box.  4) Click <em>OK</em>.</td>
</tr>
</tbody>
</table>
|                 | Delete the PBT files. | 1) Select the PBT file you want to delete.  2) Click *Delete* to delete the selected PBT file.  3) Click *OK*.  

**NOTE:** The default PBT file cannot be deleted.

| Set the default PBT file. | 1) Select the check box in front of a PBT file.  2) Click *OK*.  

**NOTE:** 1) only one PBT file can be set to the default PBT.  2) Code Insight can only be effective to the default PBT.

| System tab page | Specify what to display in the list. | 1) In the *Auto List* group box, opt to automatically display functions, events, properties, controls and/or variables in the list.  
|                 | 2) In the *Feature Type* group box, opt to display only the unsupported features and/or the supported features in the list.  3) Click *OK*. |

### 12.2.2 Enabling Code Insight

After you complete the configuration, you can enable Code Insight and use it to develop the PowerBuilder application.

Click on the icon of Code Insight and select *Enable* from the status-area-icon menu, as shown in Figure 12-3.

**Figure 12-3: Enable Code Insight**

The Code Insight icon appears in one of the following statuses:

<table>
<thead>
<tr>
<th>Status</th>
<th>Detailed Steps</th>
</tr>
</thead>
</table>
| ![Status Icon] | Code Insight remains *disabled*.  
This happens when the PBT File list in the Code Insight Configuration window is empty.  Make sure you have added the PBT file to the list. |
Code Insight is **enabled but ineffective** in the current application. To make it effective, make sure you have

1) Configured the PBT file of the current application as the default PBT file in the Code Insight Configuration window.

2) Opened an object of the current application in a painter (not in the Source editor) by double-clicking it in the Workspace or selecting Edit from the popup menu.

After taking the above steps, Code Insight becomes effective (on).

Code Insight is **enabled and effective**.

You can develop an application with the assistance of Code Insight.

### 12.3 Coding with Code Insight

When Code Insight is enabled and effective (on) for an application, you can start editing the objects of the application using Code Insight. For example, take the following steps to add a new control and use Code Insight to edit it:

**Step 1** – Add a button to the current object in the painter.

**Step 2** – Save the object by clicking the Save button on the PowerBar.

**Note:** Saving the object enables Code Insight to generate the supported and unsupported feature list for the new content. Code Insight cannot detect the new content unless the object is saved.

**Step 3** – Double click the button to open the Script view and pause for a few seconds after typing an identifier followed by a dot (for example, pause after “This.”).

If there is more than one property, variable, method, or statement that could be inserted, the Code Insight feature list pops up for you to select a supported property, variable or method, as shown in Figure 12-4.
Figure 12-4: Develop an application with Code Insight

The unsupported properties, variables or methods in the popup list will have a red icon in front, as shown in Figure 12-5. Avoid using the unsupported features.

Figure 12-5: Appeon unsupported features displayed in popup window

Step 4 – Select a supported property, variable or method from the popup list.
You can use Code Insight to help develop an application free of unsupported features and suitable for migration.
13 Launching Appeon Enterprise Manager

Appeon Enterprise Manager (AEM) is a Web-based application that manages the converted Web applications and Appeon Server over the Internet, an intranet or an extranet. It includes a comprehensive set of easy-to-use tools for system configuration, system maintenance, performance optimization, and application security.

AEM is included when Appeon Server is installed.

AEM is accessible from any Client’s Web browser or it is possible to access AEM through the AEM button that resides on the Appeon Developer toolbar. Clicking the AEM button will automatically open a Web browser window and load the AEM entry page.

13.1 Requirements

Before launching AEM, verify that:

1. The Appeon Server hosting AEM has been started.
2. You have correctly specified the host name and port of the Appeon Server in the Appeon Server Profile Configuration page.

13.2 Launching Appeon Enterprise Manager

Step 1 – Click the AEM button (’ex) on the Appeon Developer toolbar.

If more than one Appeon Server profile is configured, the “Select an AEM URL to run” page is displayed, as shown in Figure 13-1. All Appeon Server profiles and corresponding AEM URLs are listed. The Appeon Server configured as the default server is highlighted and selected.

If only one Appeon Server profile is configured, the entry page of AEM is displayed, as shown in Figure 13-2.

Figure 13-1: Select an AEM URL to run

![Select an AEM URL to run](image-url)
Step 2 – Select an AEM URL and click *Run*, as shown in Figure 13-1. The entry page of AEM is loaded in an Internet Explorer page, as shown in Figure 13-2.

**Figure 13-2: Appeon Enterprise Manager entry page**

![AEM Logon - Windows Internet Explorer](image)

The AEM URL is automatically entered into the Internet Explorer address bar when you click the *Run* button. You can configure AEM’s URL and connection method (https or http) in the Appeon Server Profile Configuration page.

The default username is “admin” and the default password is “admin”. The username and password settings for AEM can be modified in Security settings in AEM. Refer to the *Appeon Server Configuration Guide* for more information.
14 Appeon for PowerBuilder Help

Appeon provides a comprehensive HTML help system that contains:

- Introduction to Appeon
- What's New in Appeon, the HTML version of the *New Features Guide*.
- Appeon Installation Guide
- Appeon Migration Guide
- Appeon Features Help, the HTML version of the *Supported Features Guide*
- Working with Appeon Developer, the HTML version of the *Appeon Developer User Guide*
- Appeon Server Configuration Guide
- Web Server Configuration Guide
- Appeon Troubleshooting Guide & Error Messages
- Appeon Workarounds Guide & Workarounds PBL Reference
- Appeon Performance Tuning Guide

Appeon for PowerBuilder Help is in compiled HTML format, so that you can search by index or by typing keywords, navigate back and forth, or print a specified page.

To access Appeon HTML Help:

Step 1 – Click the *Help button* ( 🌐 ) in the Appeon Developer toolbar.

Step 2 – Click the corresponding help book to access the detailed help information, as shown in Figure 14-1.
Figure 14-1: Appeon for PowerBuilder Help
15 Extended Toolkit

15.1 Overview

Appeon provides extended tools to facilitate the Web migration process. These tools include:

- DLL/OCX Files Package Tool: It is designed as a solution for packaging and installing large-sized DLL/OCX files.
- Appeon Silent Installer (EAServer Windows only): Installs EAServer, Appeon Server, Appeon Server EBF (Emergency Bug Fix), and Appeon Web applications in the silent mode.
- Appeon Developer Toolbar Register Tool: Enables you to register (or unregister) one Appeon Developer version to more than one PowerBuilder version or different Appeon Developer versions to different PowerBuilder versions.

15.2 DLL/OCX Files Package tool

15.2.1 Overview

DLL and OCX files are frequently used in PowerBuilder client/server application. Appeon offers you two ways to distribute DLL/OCX files.

Table 15-1: Two ways to distribute DLL/OCX files

<table>
<thead>
<tr>
<th>When to use…</th>
<th>Which tool to use…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deploy DLL/OCX files to the server and then download them to the client at runtime.</td>
<td>In the Additional Files tab of the Application Profiles Configuration window. Refer to the Deploy external non-System DLL/OCX files section.</td>
</tr>
<tr>
<td>Package DLL/OCX files and install them to the client before the application is run for the first time.</td>
<td>Use the DLL/OCX file packaging wizard. Refer to the following sections for detailed information.</td>
</tr>
</tbody>
</table>

15.2.2 Packaging DLL/OCX files

Step 1 – Run the EonDLLOCX.exe file under the \Toolkit\DLL&OCX Package folder in the Appeon Developer installation directory.

The DLL/OCX File Packaging Wizard starts, as shown in Figure 15-1. Click Next to proceed.
Figure 15-1: DLL/OCX File Packaging Wizard

Step 2 – Select an application profile and click Next (Figure 15-2).

Only application profiles that are already configured with DLL/OCX files will be listed. For configuring DLL/OCX files for an application profile, refer to the Deploy external non-System DLL/OCX files section. When Next is clicked, the pre-configured DLL/OCX files will be automatically added for you.

Or, skip the selection and click Next to manually add the DLL/OCX files.
Figure 15-2: Select an application

Step 3 – Click *Add* to add the DLL/OCX files, select the *Register* box to automatically register the file when installed, and click *Next* to proceed (Figure 15-3).

Figure 15-3: Add DLL/OCX files

Step 4 – Specify the location where the package will be created (Figure 15-4). A folder named Package will be created under this location.
Step 5 – Review the settings and click **Next** to generate the package (Figure 15-5).

**Figure 15-5: Review settings**

Step 6 – Wait until the file packaging process is complete.

Step 7 – When packaging is complete, click **Finish** to close the wizard (Figure 15-6).
To view the log information generated during the process, select the View Log box and then click Finish. The log file will be automatically opened. You can also manually open the DLLOCX.log file in the generated package.

Figure 15-6: Packaging complete

15.2.3 Installing and uninstalling DLL/OCX files

You will use the same setup program (Setup.exe file) in the generated package for installing and uninstalling DLL/OCX files:

- If the DLL/OCX files are not installed on the client, when you run the setup program, you will be guided to install the files. Refer to 15.2.3.a: Installing DLL/OCX files for detailed instructions.

- If the DLL/OCX files are already installed on the client, when you run the setup program, you will be guided to reinstall or uninstall the files. If you select to reinstall the files, you will run through the same process as a new installation, as described in Section 15.2.3.a: Installing DLL/OCX files. If you select to uninstall the files, refer to Section 15.2.3.b: Uninstalling DLL/OCX files for detailed instruction.

15.2.3.a Installing DLL/OCX files

Step 1 – On a client machine, run the Setup.exe file in the generated package.

The DLL/OCX File Installation Wizard starts, as shown in Figure 15-7. Click Next to proceed.
Figure 15-7: DLL/OCX File Installation Wizard

Step 2 – Specify the location where the DLL/OCX files will be installed. Click Next to start the installation process (Figure 15-8).

The default installation location is %SystemRoot%\system32\AppeonPlugin (%SystemRoot% indicates the folder where Windows is installed, typically C:\Windows); if you specify a custom location in AEM (see Appeon Server Configuration Guide for details), then specify the same location here.
Step 3 – Wait until the installation is complete.

During the installation process, the DLL/OCX files will first be copied to the specified location and then registered if they have been selected for auto-registration when packaged.

Step 4 – When the installation is complete, select the **Let wizard restart the system now** option and click **Finish** to close the installation wizard and restart your machine (Figure 15-9).

To view the log information generated during the process, select the **View Log** box and click **Finish**. The log file will be automatically opened. You can also manually open the DLLOCX.log file in the generated package.
15.2.3.b Uninstalling DLL/OCX files

Step 1 – On a client machine, run the Setup.exe file in the generated package. The DLL/OCX File Installation Wizard starts, as shown in Figure 15-10. Click Next to proceed.

Figure 15-10: DLL/OCX File Installation Wizard

This wizard will help you setup DLL/OCX files for distribution.

Click Next to proceed.
Step 2 – Select **Uninstall** and click **Next** to proceed.

If you select to reinstall the files, you will run through the same process as a new installation, as described in Section 15.2.3.a: *Installing DLL/OCX files.*

**Figure 15-11: Reinstall or uninstall**

![Maintenance Mode](image)

Step 3 – Click **Yes** to confirm the uninstall (Figure 15-12).

**Figure 15-12: Uninstall confirmation**

![Question](image)

Step 4 – Wait until the uninstall is complete.

Step 5 – When uninstall is complete, click **OK** to exit (Figure 15-13).
15.3 Appeon Silent Installer (Windows EAServer only)

15.3.1 Overview

Note: this tool does not work in JBoss, WebLogic and WebSphere versions.

To help ISVs create an easy-to-use Appeon installation package for the end users, Appeon specially provides a silent installer which can be customized to install EAServer, Appeon Server, Appeon Server EBF (Emergency Bug Fix), and Appeon Web applications in the silent mode.

This guide provides some useful instructions for ISVs on:

- Creating folder structure of Silent Installer
- Copying installation program of EAServer, Appeon Server, Appeon Server EBF or Web applications
- Customizing configuration file
- Burning installation CDs

Additional instructions are also provided for end users on:

- Running Appeon Silent Installer

15.3.2 Creating folder structure of Silent Installer

The installation program files of EAServer, Appeon Server, Appeon Server EBF and Web application must be placed in the required folder structure, so that Appeon Silent Installer can run them. For your convenience, Appeon provides a compressed folder structure which can be decompressed under any directory. To decompress and create the folder structure and the required files, take the following steps:

Step 1 – Run SilentInstall.exe in the “\Toolkit\Silent Install” under the Appeon Developer installation directory, for example, C:\Program Files\Appeon\Developer6.5\Toolkit\Silent Install.

Step 2 – Select where to create the folder structure. Click OK.
After the creation, a folder named “Setup” is created with the following sub-folders and files:

The following is a brief introduction to each folder in the Appeon Silent Installer:

- **Setup**: The root directory of the Silent Installer. It contains the “Appeon” folder, the Setup.exe, Uninstall.exe, SilentInstall.txt, and msvcp60.dll files.

- **Appeon**: It contains the “AS”, “AS EBF”, “EAServer” and “WebApp” directory.
  - **AS**: It stores the installation program of Appeon Server.
  - **AS EBF**: It stores the installation program of Appeon Server EBF.
  - **EAServer**: It stores the installation program of EAServer.
  - **WebApp**: It stores the Web application files packaged by the Appeon application packaging tool.
• Setup.exe: The setup program for launching the silent installation.
• SilentInstall.txt: The configuration file which can be modified to customize the installation.
• Uninstall.exe: The uninstall program for silently removing what has been installed with Appeon Silent Installer.
• msvcp60.dll: The DLL file for Appeon Silent Installer.

15.3.3 Copying installation program of EAServer, Appeon Server, Appeon Server EBF or Web applications

After the required folders and files are generated, you can start copying the installation program files of Appeon Server, EAServer and Web applications to the corresponding folder.

To copy the EAServer installation program to Appeon Silent Installer:
Copy all files and folders from the directory where setup.exe file resides to the “EAServer” folder of Appeon Silent Installer.

To copy the Appeon Server installation program to Appeon Silent Installer:
Copy all files from the “ASP” folder on the Appeon installation CD to the “AS” folder of Appeon Silent Installer.

To copy the Appeon Server EBF installation program to Appeon Silent Installer:
Copy all files from the "asebf" folder on the Appeon installation CD to the "AS EBF" folder of Appeon Silent Installer.

To copy Web application packages to Appeon Silent Installer:
Step 1 – Verify that the Web application packages have already been generated using the Appeon application packaging tool. If no package is generated, follow steps in Packaging a Web application.
Step 2 – Copy the generated packages to the “WebApp” folder of Appeon Silent Installer.

15.3.4 Customizing configuration file

With Appeon Silent Installer, you can customize the installation program using the configuration file (SilentInstall.txt). The following is a detailed description of each parameter in the SilentInstall.txt file. Each parameter contains no default value.

15.3.4.a Installation types

Specify whether to install EAServer, Appeon Server and Web applications.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAServer</td>
<td>Specifies whether to install EAServer. If you do not want to install EAServer using this setup program, make sure that a supported EAServer was installed already.</td>
</tr>
<tr>
<td>Accepted Values</td>
<td>Yes – Install EAServer. No – Not to install EAServer, instead use an existing EAServer.</td>
</tr>
<tr>
<td>Default Value</td>
<td>None</td>
</tr>
<tr>
<td>Example</td>
<td>EAServer =&quot;yes&quot;</td>
</tr>
</tbody>
</table>
### Parameter: Appeon Server

**Description**: Specifies whether to install Appeon Server.

**Accepted Values**: Yes, No

**Default Value**: None

**Example**: Appeon Server="yes"

### Parameter: Appeon Server EBF

**Description**: Specifies whether to apply Appeon Server EBF.

**Accepted Values**: Yes, No

**Default Value**: None

**Example**: Appeon Server EBF="yes"

### Parameter: WebApplication

**Description**: Specifies whether to install Web applications.

**Accepted Values**: Yes, No

**Default Value**: None

**Example**: WebApplication="yes"

### 15.3.4.b EAServer and ASA path

Specify the path of EAServer and ASA. ASA is required because Appeon Server has an ASA database for storing the sensitive system data.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Accepted Values</th>
<th>Default Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAServerPath</td>
<td>If the EAServer parameter is set to “No”, this specifies the path of an existing EAServer. The specified path will be validated. If the EAServer parameter is set to “Yes”, this indicates where EAServer will be installed.</td>
<td>A valid path</td>
<td>None</td>
<td>EAServerPath=&quot;C:\Program Files\Sybase\EAServer&quot;</td>
</tr>
<tr>
<td>ASAPath</td>
<td>Specifies the ASA installation path. The specified path will be validated.</td>
<td>A valid path</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>
**15.3.4.c EAServer Host Name, Port, Username and Password**

Specify the EAServer host name, port number, user name and password.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Accepted Values</th>
<th>Default Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAServerName</td>
<td>Specifies the IP address of EAServer.</td>
<td>A valid IP address</td>
<td>None</td>
<td>EAServerName=&quot;localhost&quot;</td>
</tr>
<tr>
<td>EAServerPort</td>
<td>Specifies the port number of EAServer.</td>
<td>A valid port</td>
<td>None</td>
<td>EAServerPort=&quot;9000&quot;</td>
</tr>
<tr>
<td>EAServerUserName</td>
<td>Specifies the user name to connect to EAServer.</td>
<td>A valid user name on EAServer.</td>
<td>None</td>
<td>EAServerUserName=&quot;jagadmin&quot;</td>
</tr>
<tr>
<td>EAServerPassWord</td>
<td>Specifies the password to connect to EAServer.</td>
<td>A valid password for the current user name.</td>
<td>None</td>
<td>EAServerPassWord=&quot;&quot;</td>
</tr>
</tbody>
</table>

**15.3.4.d Appeon Server Port, Username and Password**

Specify the Appeon Server port number, user name and password.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Accepted Values</th>
<th>Default Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppeonServerPort</td>
<td>Specifies the HTTP port number of Appeon Server.</td>
<td>A valid port</td>
<td>If both EAServer and AppeonServer parameters are set to “Yes”, port “9988” is</td>
<td></td>
</tr>
</tbody>
</table>
the only available port that can be used.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Accepted Values</th>
<th>Default Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppeonServerPort</td>
<td>Specifies the user name for deploying an application to Appeon Server. This deployment user name can be created using the Security tool in AEM.</td>
<td>A valid deployment user name on Appeon Server.</td>
<td>None</td>
<td>AppeonServerPort=&quot;9988&quot;</td>
</tr>
<tr>
<td>AppeonServerUserName</td>
<td>Specifies the password for deploying an application to Appeon Server.</td>
<td>A valid deployment password for the current user name.</td>
<td>None</td>
<td>AppeonServerPassWord=&quot;&quot;</td>
</tr>
</tbody>
</table>

### 15.3.4.e Web Server path

Specify the Web root of the Web server.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Accepted Values</th>
<th>Default Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServerRootPath</td>
<td>Specifies the document root of the Web server. If the WebApplication parameter is set to “No”, this parameter will not take effect. If the WebApplication parameter is set to “Yes”, the specified document root will be validated.</td>
<td>A valid path</td>
<td>None</td>
<td>WebServerRootPath=&quot;C:\Program Files\Sybase\EAServer\html&quot;</td>
</tr>
</tbody>
</table>

### 15.3.4.f Setup Log path

Specify the log file path.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Accepted Values</th>
<th>Default Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogPath</td>
<td>Specifies the path for the setup log file (APBLog.txt).</td>
<td>A valid path</td>
<td>None</td>
<td>LogPath=&quot;c:\log&quot;</td>
</tr>
</tbody>
</table>
15.3.4.g Web applications

Specify the number of Web applications, application name, connection cache, transaction object and database type.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Accepted Values</th>
<th>Default Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebApplications</td>
<td>Specifies the number of Web applications that you want to install.</td>
<td>Any number greater than zero</td>
<td>None</td>
<td>WebApplications=&quot;2&quot;</td>
</tr>
</tbody>
</table>

If more than one application will be installed, create the following parameters through incrementing by “1”, for example, “WebApplication2”, “EAServerNVO2”, “ConnectCache2” and “AEM2” for the second application, “WebApplication3”, “EAServerNVO3”, “ConnectCache3” and “AEM3” for the third application etc.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Accepted Values</th>
<th>Default Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebApplication1</td>
<td>Specifies the actual name of the first Web application.</td>
<td>An alphanumeric string</td>
<td>None</td>
<td>Examples for the first application: WebApplication1 = &quot;appeon_sales_application_demo&quot; Examples for the second application: WebApplication2 = &quot;appeon_code_examples&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Accepted Values</th>
<th>Default Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAServerNVO1</td>
<td>Specifies whether to install NVO components.</td>
<td>Yes, No</td>
<td>None</td>
<td>EAServerNVO1=&quot;Yes&quot;</td>
</tr>
</tbody>
</table>

To make connection caches work on the target Appeon Server, be sure to install the required database drivers to the target Appeon Server. For more information on database drivers, refer to the Database Connection Setup section in the Appeon Server Configuration Guide.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Accepted Values</th>
<th>Default Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectCache1</td>
<td>Specifies whether to install connection caches.</td>
<td>0 – Not to install connection caches. 1 – Install and overwrite existing connection caches. 2 – Install but not to overwrite existing connection caches.</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>
Example | ConnectCache1="1"
---|---

<table>
<thead>
<tr>
<th>Parameter</th>
<th>AEM1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Specifies whether to install AEM settings.</td>
</tr>
<tr>
<td>Accepted Values</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>No</td>
</tr>
<tr>
<td>Default Value</td>
<td>None</td>
</tr>
<tr>
<td>Example</td>
<td>AEM1=&quot;Yes&quot;</td>
</tr>
</tbody>
</table>

### 15.3.4.h Web application uninstall program path

Specify the directory for the Web application uninstall program. A shortcut for the uninstall program will be created in Windows Start menu.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>UninstallWebAppPath</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Specifies where the uninstall program for Web applications will reside.</td>
</tr>
<tr>
<td>Accepted Values</td>
<td>A valid path.</td>
</tr>
<tr>
<td>Default Value</td>
<td>None</td>
</tr>
<tr>
<td>Example</td>
<td>UninstallWebAppPath=&quot;C:\uninstallwebapp&quot;</td>
</tr>
</tbody>
</table>

### 15.3.4.i Resource language

Specify the language of the messages displayed during the installation or uninstallation process.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Specifies the language of UI messages.</td>
</tr>
<tr>
<td>Accepted Values</td>
<td>EN -- English</td>
</tr>
<tr>
<td></td>
<td>JP -- Japanese</td>
</tr>
<tr>
<td>Default Value</td>
<td>EN</td>
</tr>
<tr>
<td>Example</td>
<td>Language=&quot;EN&quot;</td>
</tr>
</tbody>
</table>

### 15.3.4.j SilentInstall.txt example

The following example shows a sample SilentInstall.txt file.

```
# Configure installation type: Yes or No
EAserver ="Yes"
Appeon Server="Yes"
Appeon Server EBF="Yes"
WebApplication="Yes"
```
# Enter the path of the EAServer installed or you want to install.
EAServerPath="C:\Program Files\Sybase\EAServer"

# Enter the ASA path
ASAPath="C:\Program Files\Sybase\Shared\ASA802"

# Enter the Port, Username, and Password of EAServer
EAServerName="localhost"
EAServerPort="9000"
EAServerUserName="jagadmin"
EAServerPassWord=""

# Enter the Port, Username, and Password of Appeon Server
AppeonServerPort="9988"
AppeonServerUserName=""
AppeonServerPassWord=""

# Enter the Web root of Web server
WebServerRootPath="C:\Program Files\Sybase\EAServer\html"

# Enter the path of Setup Log: APBLog.txt
LogPath="c:\log"

# Enter the Web application number
WebApplications="1"

# Web Application Configuration
# ConnectCache:
# 0: not install
# 1: install, overwrite if exist
# 2: install, not overwirte if exist
# EAServerNVO: Yes or No
# AEM: Yes or No
15.3.5 Burning installation CDs

After configuration is completed, you can burn installation CDs with the customized silent installer (the entire “setup” folder) and distribute CDs to your end users.

The Appeon Silent Installer can be provided in one, two or three CDs:

- In one CD: Installing the Appeon Server (“AS” folder), Appeon Server EBF (“AS EBF” folder), EAServer (“EAServer” folder) and the Web application (“WebApp” folder).
- In two CDs: One for installing the Appeon Server (“AS” folder), Appeon Server EBF (“AS EBF” folder), and EAServer (“EAServer” folder), and the other for installing the Web application (“WebApp” folder).
- In three CDs: One for installing the Appeon Server (“AS” folder) and Appeon Server EBF (“AS EBF” folder), one for installing EAServer (“EAServer” folder), and one for installing the Web application (“WebApp” folder).

Each CD must contain the following four files right under the Setup folder (that is, place the following files in the same directory level of the “AS”, “EAServer” and “WebApp” folder):

- Setup.exe
- SilentInstall.txt
- Uninstall.exe
- msvcp60.dll

15.3.6 Running Appeon Silent Installer (for End Users)

This section provides some additional instructions for end users on using the Appeon Silent Installer to

- Install EAServer, Appeon Server, Appeon Server EBF and/or Web applications
- Uninstall EAServer, Appeon Server, Appeon Server EBF and/or Web applications
15.3.6.a Installing EAServer, Appeon Server, Appeon Server EBF and/or Web applications

Before running the Appeon Silent Installer, be aware that

1) The Setup.exe program can only be run on the Windows platform.

2) The Appeon Server and the Web server must reside on the same machine.

Start the silent installation by running the Setup.exe file on the installation CD.

By default, the configuration file in the same location of the Setup.exe file will be used during the installation process. To use a configuration file residing in a different location, run the Setup.exe file from the command line and append the configuration file location, for example, `Setup.exe d:/setuptmp/SilentInstall.txt`.

15.3.6.b Uninstalling EAServer, Appeon Server, Appeon Server EBF and/or Web applications

To silently uninstall EAServer, Appeon Server, Appeon Server EBF, and/or Web applications:

Run Uninstall.exe on the installation CD. The uninstall program will automatically load the SilentInstall.txt file on the same CD and remove what has been installed with Appeon Silent Installer.

To uninstall Web applications from Windows Start menu:

Select Windows Start | Programs | Appeon Web Application | Undeploy Appeon Web Application. Or run the uninstall program under the location specified by the “UninstallWebAppPath” parameter.

You will go through the same process as those in the Uninstalling an application section in the “Packaging and Installing Web Applications” chapter.

15.3.6.c FAQs

1. What happens if EAServer is started during the process of silent installation?

   Appeon Silent Installer will shut down the EAServer and continue the installation.

2. What happens if EAServer or Appeon Server was already installed during the process of silent installation?

   Appeon Silent Installer will overwrite the existing EAServer or Appeon Server.

3. What happens if the names of NVO packages for a Web application are duplicated?

   Appeon Silent Installer will still deploy the first NVO package, but will not deploy the second NVO package. This activity will be recorded in the log file.

15.4 Appeon Developer Toolbar Register tool

Appeon Developer can run with PowerBuilder 8, 9, 10, 10.5, 11, 11.5, and 12 (Classic) at the same time and multiple Appeon Developer versions can run on the same machine. The Appeon Developer Toolbar register tool enables you to register (or unregister) one Appeon Developer version to more than one PowerBuilder version or different Appeon Developer versions to different PowerBuilder versions, so that you can easily and flexibly select which Appeon Developer version and PowerBuilder version to use.

Following are steps to use the Appeon Developer Toolbar register tool:
Step 1 – Select Start | Programs | Appeon 6.5 for PowerBuilder | Appeon Developer | Toolbar Register Tool.

**Figure 15-16: Registry tool**

The PowerBuilder versions existing on this computer will be listed.

Step 2 – Select the Appeon Developer version to be registered in the PowerBuilder version.

If you want to unregister an Appeon Developer version from a PowerBuilder version, clear the selection in the Appeon Developer Version column.

**Figure 15-17: Register toolbar to PowerBuilder**

Step 3 – Click OK to register (or unregister) the selected Appeon Developer version with the PowerBuilder version.
16 Technical Support


When logging a case, be prepared to provide the following files to Sybase Technical Support in order to aid in debugging:

Table 16-1: Provide the following files for technical support

<table>
<thead>
<tr>
<th>File Type</th>
<th>File Name</th>
</tr>
</thead>
</table>
| Appeon Server configuration file | For EAServer\WebLogic\WebSphere\JEUS\JBoss:  
  All files in the %appeonserver%repository\<instancename>\config folder.  
For .NET IIS:  
All files in the %appeonserver%\AEM\config folder.  
%appeonserver% indicates the Appeon Server installation path. |
| Appeon Server log file           | For EAServer\WebLogic\WebSphere\JEUS\JBoss:  
  All files in the %appeonserver%repository\<instancename>\log folder.  
For .NET IIS:  
All files in the %appeonserver%\AEM\Log folder. |
| Appeon Server Repository DB      | For EAServer:  
%appeonserver%\db\AppeonServer.db  
For .NET IIS:  
All files in the %appeonserver%\AEM\db folder.  
For WebLogic\WebSphere\JEUS\JBoss:  
%appeonserver%\db\appeondb.script |
| Appeon license file              | For EAServer\WebLogic\WebSphere\JEUS\JBoss:  
%appeonserver%\license.appeon  
For .NET IIS:  
%appeonserver%\AEM\bin\license.appeon |
| Web Server configuration file    | httpd.conf                                                               |
| Web Page file                    | Web Server path\application name\window name.html                     
Web Server path\application name\window name.js  
Web Server path\application name\all DataWindows on window.xml  
Note: If the window named includes inheritance, then also supply all ancestors.html and ancestors.js files. |
| PowerBuilder exported source code file | PowerBuilder application path\exported window name.srw  
PowerBuilder application path\exported DataWindow names.srd  
Note: Exported windows should include exported ancestor windows and exported ancestor user objects. |
Index

A
About This Book, 1
Accessing the UFA tool, 62
Accessing Web applications, 108
adding application profile, 25
adding deployment profile, 54
Adding variables or expressions to Watch view, 96
Additional enhanced features, 117
advanced Internet options settings, proxy server, 103
advanced Internet options settings, running Web application, 102, See also temporary Internet files settings, running Web application
AEM, launching, 159
AEM, launching requirements, 159
Analyzing an application, 61
analyzing unsupported features, 64
Analyzing Unsupported Features, 60
analyzing Web migration process, 6
Appeon Developer configuration
application profiles configuration selecting debug mode, 24
Appeon Developer toolbar, 4
Appeon Developer Toolbar Register Tool, 182
Appeon enhanced Web features. See enhanced Web features
Appeon Enterprise Manager, launching. See AEM, launching
Appeon for PowerBuilder Help, 161
Appeon installation, 7
Appeon Server profile settings, 46
Appeon Server profiles configuration
Appeon Server Settings
Password, 48
Server Port, 47, 48
Appeon Silent Installer
Appeon Server Port, Username and Password, 176
Burning installation CDs, 181
copy installation program, 174
create folder structure, 172
Customizing configuration file, 174
EAServer and ASA Path, 175
EAServer Host Name, Port, Username and Password, 176
Installing Appeon Server, EAServer and/or Web applications, 182
Installation types, 174
Running Appeon Silent Installer (for End Users), 181
Setup log path, 177
specify Web application parameters, 178
Web Application Uninstall Program Path, 179
Web server path, 177
Appeon Silent Installer, 172
application profile adding, 25
for deployment, specifying, 24
removing, 25
application profile properties Deploy .NET/COM components, 31
application profile properties Additional Files, 30
basic settings, 26
Database profiles settings, 28
DLL/OCX, 18
How INI files are supported in Appeon, 30
INI file, 30
Misc settings application language, 20
command line arguments, 19
log-writing, 21
application profile properties Images, 31
application profile properties DLL/OCX, 31
application profile properties How the deployed DLL/OCX files work on Web, 32
application profile properties Misc settings, 33
application profile properties Misc settings command line arguments, 33
application profile properties
Misc settings
  application language, 34
application profile properties
  Misc settings
    performance settings, 35
application profile properties
  Misc settings
    Parsing options for AX only, 36
application profile properties
  Misc settings
    log-writing, 37
application profile properties
  Web Service Profiles, 39
application profile properties
  Runtime Settings, 40
Application Profiles tab page, 24

B
before analyzing unsupported features, 61
browser considerations, 107
build, PowerBuilder feature
  incremental, 61
  regenerate, 61
Burning installation CDs, 181

C
Code Insight, 154
  activate Code Insight, 154
  Code Insight configuration, 155
  coding with Code Insight, 157
  configure Code Insight, 155
  develop applications with Code Insight, 157
  enable Code Insight, 156
  start Code Insight, 154
Code lines that can be set as breakpoints, 93
configuring Appeon Developer, 9
configuring application profiles, 23
Configuring basic settings, 10
configuring connection cache profiles, 55
configuring database, 7
configuring database type, 28
configuring DB type, 28
configuring deployment profiles, 53
Configuring deployment settings, 12
Creating folder structure of Silent Installer, 172
Customizing configuration file, 174

D
database configuration, 7
DataWindow menu feature, HTML
DataWindows
  filter, 114
  sort, 112
DataWindow printin, 115
DataWindow SaveAs, 116
debug mode
  Debug – JS Code, 24
  Debug – PS/JS, 24
  Encrypted, 25
debug mode, selecting. See Web file debug mode
Debugging
  fix the code or stop the debug procedure, 98
Debugging Web applications, 90
  examine at a breakpoint, 95
  introduction, 90
  run the application in debug mode, 94
  scripts, 90
  Setting breakpoints, 93
  starting Appeon Debugger, 91
Declaring transaction object(s), 14
deploying PowerBuilder applications, 78
Deploying the PowerBuilder application, 83
deploying Web migration process, 6
Deployment duration for full deployments, 79
Deployment duration for incremental deployments, 79
deployment mode
  automatically set deployment mode, 83
  deployment process
    selecting deployment profile, 81
deployment process
  build PB application, 80
  prepare PB application, 80
  specify deployment settings, 80
deployment process, 80
deployment process
  unattended deployment option, 81
deployment process
  select deployment mode, 82
deployment profile
  adding, 54
Deployment profile settings, 54
deployment profiles, 54
deployment profiles configuration
Profile Name, 55
Deployment Profiles tab page, 54
deployment time, 79
deployment, performance considerations, 78
Developer toolbar, 4
Developing with Code Insight, 154
Disabling anti pop-up software, 107
DLL/OCX Files Package Tool, 163

E
enable Appeon DataWindow menu, 110
Enabling Code Insight, 156
enhanced Web features, 110
enable Appeon DataWindow menu, 110
Web DataWindow menu, 111
Evaluating an expression, 97
examine at a breakpoint
add variables or expressions to Watch view, 96
change the value of a variable, 97
change the value of an expression, 97
evaluate an expression, 97
examine context in Call Stack view, 97
step through the application, 98
variable and expression, 95
Examining an application at a breakpoint, 95
Examining context in Call Stack view, 97
Extended Toolkit, 163

F
FAQs for installing/uninstalling Appeon Server, EAServer and/or Web applications, 182
File Transfer Type
FTP File Transfer, 49
Local File Transfer, 49
filter UFA report items
Filtering by categories, 71
Filtering by objects, 71
Filtering by priorities, 72
filter UFA report items, 70
Fixing the code/stopping the debug procedure, 98
FTP File Transfer, 49
FTP File Transfer Settings
FTP Username, 50
FTP server, permission settings, 51

H
heaviness of a window, measure, 76
Heavy Window report
reading Heavy Window report, 76
help, 161
how to use this book, 1
how to view reports/logs, 119
HTML DataWindow menu feature
filter, 114
sort, 112

I
if you need help, 3
information manager, usage, 119
Installing a Web application
instructions, 132
Installing a Web application, 132
Installing and uninstalling a Web application, 132
Installing and Uninstalling DLL/OCX files, 167
installing Appeon, 7
Installing DLL/OCX files, 167
Internet Explorer settings, 100
Introduction, 4
Introduction to the debugging procedure, 90

L
language settings requirements, 106
launching AEM, 159
launching AEM, requirements, 159
Launching Appeon Enterprise Manager, 159
Launching Web applications from Run button, 108
Local File Transfer, 49

M
Making heavy windows lighter, 77
Managing connection cache profiles, 55
Managing database type profiles, 42
Managing deployment profiles, 53
Managing server profiles
Server Profiles tab page, 45
Managing server profiles, 45
Methods for setting breakpoints, 93
Modifying the deploy-config file, 131
modifying unsupported features, 66
modifying Web migration process, 6
Index

O
Open/Save UFA report, 69

P
packaging a Web application
instructions, 123
packaging a Web application, 122
what can be packaged?, 122
Packaging and Installing Web
Applications, 122
Packaging DLL/OCX files, 163
performance considerations, deployment, 78
permission settings, FTP server, 51
Points to note before installation, 132
PowerBuilder applications, deploying, 78
Preparing the PowerBuilder application, 80
priority of unsupported features, 73

R
Reading Heavy Window Report, 76
related documents, 2
removing application profile, 25
Resource language, 179
Run button, 108
Running Appeon Silent Installer, 181
Running the application in debug mode, 94
running web application
Internet Explorer settings, 100
running Web application
Windows account with administrator
rights, 99
running Web applications, 99
requirements, 99
running Web applications, 108
Running Web applications
Selecting a run mode, 109

S
Searching for UFA Report items, 70
Security settings, ActiveX control
download, 104
UFA report, 70
Selecting a run mode, 109
Selecting DB Type(s), 13
selecting deployment profile, 81
Selecting image files, 17
Selecting INI files, 17
Selecting PBL file(s), 11
Selecting report view mode, 70
Selecting the deployment mode, 82
selecting Web file debug mode, 24
server profiles
Appeon Server profiles, 46
default server profiles, 46edit/add/remove server profiles, 46
Web Server profiles, 48
Set the database types for the application, 28
Setting breakpoints, 93
Setting breakpoints
code lines, 93
Setting breakpoints
methods, 93
Setup Log path, 177
SilentInstall.txt example, 179
Sort and filter, 112
Specify the transaction object and the
connection cache, 29
specifying application profile for
deployment, 24
Specifying report display level, 73
Specifying the deployment settings, 80
Speed of Web Deployment Process, 78
SSL Web server, 53
Start Appeon Server, shortcut, 7
starting Appeon Debugger
views, 92
Stepping through the application, 98
Summary, 22

tasks required before unsupported feature
analysis, 61
Technical support, 184
temporary Internet files settings, run Web
application. See also advanced Internet
options settings, run Web application
temporary Internet files settings, run Web
application, 100

U
UFA issue, 75
UFA report
customize general settings, 74
UFA Report Window manipulating the
UFA Report, 69
Undeploying Web Applications, 147
Undetected unsupported features, 75
Uninstall a Web application
   instructions, 147
Uninstall with Appeon Server mode, 151
Uninstall with Deployment Profile mode, 148
uninstalling a Web application, 141
Uninstalling Appeon Server, EAServer
   and/or Web applications, 182
Uninstalling DLL/OCX files, 170
Uninstalling Web application with
   Appearance Server, 148
Uninstalling DLL/OCX files, 170
Uninstalling EAServer and/or Web applications, 182
Uninstalling DLL/OCX files, 170
Uninstalling DLL/OCX files, 170
Uninstalling Appeon Server, EAServer
   and/or Web applications, 182
Uninstalling DLL/OCX files, 170
Uninstall a Web application
   instructions, 147
Uninstall with Appeon Server mode, 151
Uninstall with Deployment Profile mode, 148
uninstalling a Web application, 141
Uninstalling Appeon Server, EAServer
   and/or Web applications, 182
Uninstalling DLL/OCX files, 170
Uninstalling Web application with
   Appearance Server, 148
Uninstalling DLL/OCX files, 170
Uninstalling EAServer and/or Web applications, 182
Uninstalling DLL/OCX files, 170
Uninstalling DLL/OCX files, 170
Uninstalling EAServer and/or Web applications, 182
Uninstalling DLL/OCX files, 170
Uninstalling EAServer and/or Web applications, 182
Web application, running, 108
Web applications, 178
Web applications, running, 99
Web applications, URLs, 108
Web DataWindow menu feature
   DataWindow printing, 115
   DataWindow SaveAs, 116
Enhanced Menus, 117
Find, 111
sort and filter, 112
Web Features. See enhanced Web features
Web file debug mode
   Debug – JS Code, 24
   Encrypted, 25
Web file debug mode
   Debug – PS/JS, 24
Web migration process
   analyzing, 6
   deploying, 6
   modifying, 6
Web Server path, 177
Web Server profile settings, 48
Web Server profiles configuration
   FTP File Transfer Settings
   FTP Username, 50
Working with Heavy Window Report, 76
Working with UFA Report, 66