Appeon Developer User Guide

Appeon® 3.1 for PowerBuilder®

FOR WINDOWS

DOCUMENT ID: DC20238-01-0310-01

LAST REVISED: September 13, 2005

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

Contents

1 About This Book

1.1 Audience

This book is written for PowerBuilder developers using Appeon Developer.

1.2 How to use this book

There are fifteen 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: Analyzing Unsupported Features

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: Launching Appeon Enterprise Manager

Instructions for launching Appeon Enterprise Manager from Appeon Developer.

Chapter 8: Generating and Using Runtime Reports

Instructions for configuring and using the Web Debug Report and Web Performance Runtime Tracing Report to track runtime information and performance with deployed Web applications.

Chapter 9: Running Web Applications

Instructions for running Web applications after deployment.

Chapter 10: Using Information Manager

Instructions for viewing reports and log files generated in the Appeon migration process.

Chapter 11: Packaging and Installing Web Applications

Instructions for using Web application packaging tool to generate installation packages.

Chapter 12: Undeploying Web Applications

Instructions on how to undeploy Appeon Web applications.

Chapter 13: Developing with Code Insight

Instructions on how to use the Appeon Code Insight feature.

Chapter 14: Appeon for PowerBuilder Help

Instructions for launching the Appeon HTML help file.

Chapter 15: 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, and the Appeon ACF Demo, 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 3.1.

Working with Appeon Developer Toolbar is an HTML version of the *Appeon Developer User Guide*.

- *Appeon Enterprise Manager User Guide*:

Introduces the Appeon Enterprise Manager, a Web application that maintains Appeon Web applications and Appeon Server over the Internet, an intranet, or an extranet.

- *Appeon Supported Features Guide for Appeon Xcelerator* (or *Appeon Features Help for Appeon Xcelerator*):

Provides a detailed list of what PowerBuilder features are supported and can be converted to the Web with Appeon 3.1, using the Appeon Xcelerator deployment option, and what features are unsupported.

Appeon Features Help for Appeon Xcelerator is an HTML version of the *Appeon Supported Features Guide for Appeon Xcelerator Deployment*.

- *Appeon Supported Features Guide for Pure-JavaScript* (or *Appeon Features Help for Pure-JavaScript*):

Provides a detailed list of what PowerBuilder features are supported and can be converted to the Web with Appeon 3.1, using the Pure-JavaScript deployment option, and what features are unsupported.

Appeon Features Help for Pure-JavaScript is an HTML version of the *Appeon Supported Features Guide for Pure-JavaScript Deployment*.

- *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 includes various topics related to steps in the procedure.

- *Appeon Migration Tutorial*:

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 3.1 for PowerBuilder.

- *Using the PowerBuilder Foundation Class Library with Appeon* (or *Appeon-compliant Framework Reference*):

Provides a detailed list of what PowerBuilder PFC features are supported and can be converted to the Web with Appeon, and what features are not supported.

Appeon-compliant Framework Reference is an HTML version of the *Using the PowerBuilder Foundation Class Library with Appeon*.

- *What's New in Appeon*:

Introduces new features and changes in Appeon 3.1 for PowerBuilder.

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.

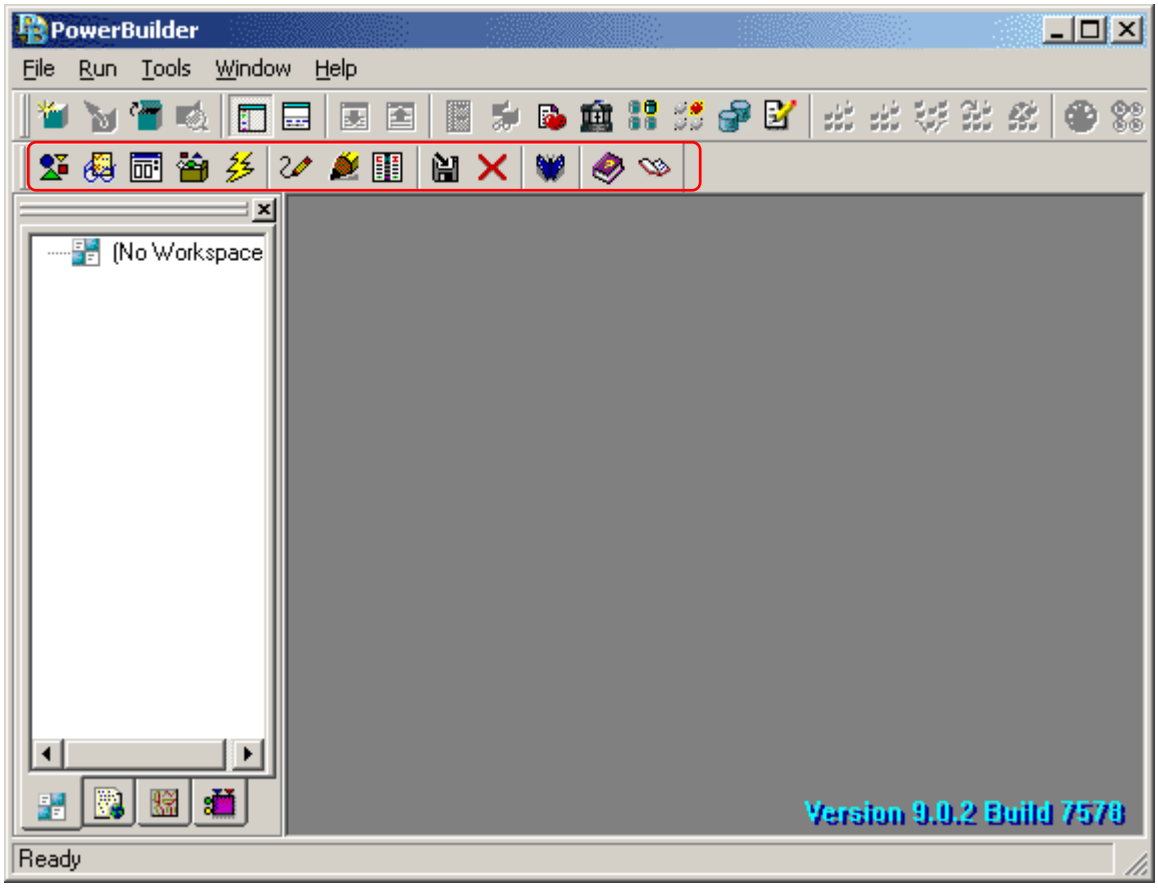
2 Introduction

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

Button	Name	Function
	Configure	Sets configurations of Appeon Developer for Web conversion.
	Analyze	Analyzes application source code for unsupported features before deployment. This analysis can cover the whole application, at object level or within an inheritance hierarchy.
	Code Insight	Develops PowerBuilder applications that are free of Appeon unsupported features.
	Deploy	Starts the complete process of deploying a PowerBuilder application to the Web.
	Run	Runs the deployed Web application in Internet Explorer.
	Trace Configure	Configures how execution performance is recorded for fine-tuning Appeon Web applications.
	Debug Configure	Configures how execution information is recorded for debugging Appeon Web applications.
	Information	Manages logs and reports.
	Package	Packages application files for installation.
	Undeploy	Undeploys Web applications from Web Server(s) and Appeon Server(s).
	AEM	Launches Appeon Enterprise Manager (AEM).
	Help	Searches, browses, prints, copies and pastes supported PowerBuilder features and example code.
	About	Displays the Appeon version information.

Apart from Code Insight, Debug Configure, Trace Configure, and Information, only one Appeon Developer toolbar function can be launched at any given time. This prevents one 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 usually 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 *Appeon Migration Guide*). It is recommended that the new PowerBuilder application conforms to Appeon's coding styles, as laid out in *Appeon Features Help for Appeon Xcelerator* or *for Pure-JavaScript*, then you will expend less effort in the first two tasks: Analyze and Modify.

- Task 1 – Analyze

  You need to 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 for Appeon Xcelerator* or *for Pure-JavaScript* 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 for Appeon Xcelerator* or *for Pure-JavaScript* 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.

  Then, automatically 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, please 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 Sybase EAServer. Once Appeon Server has been installed to EAServer, the machine hosting EAServer is then referred to as Appeon Server.

If using a non-EAServer Web Server (e.g. Apache), 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 Migration 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 Migration Guide*, or establish the mapping statically in AEM following the instructions in the *Appeon Enterprise Manager User Guide* after the application deployment.

3.3 Starting Appeon Server from a Windows shortcut

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 EAServer as the Web server, you need only start EAServer/Appeon Server. Otherwise, start both the Web server and Appeon Server. Refer to the Web server user documents for how to start the Web server, and refer to the following table for how to start the Appeon Server:

Running EAServer as a service causes two problems:

1. Appeon Web Deployment may fail during the last step.
2. Appeon PDF printing will not work.

It is recommended to always start EAServer from the Windows shortcut.

In Windows: Select Start | Programs | Appeon 3.1 for PowerBuilder | Jaguar Server.

If EAServer is running as a service, please stop the service first and manually start EAServer.

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 deployment option, 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.

4.2 Application profiles configuration

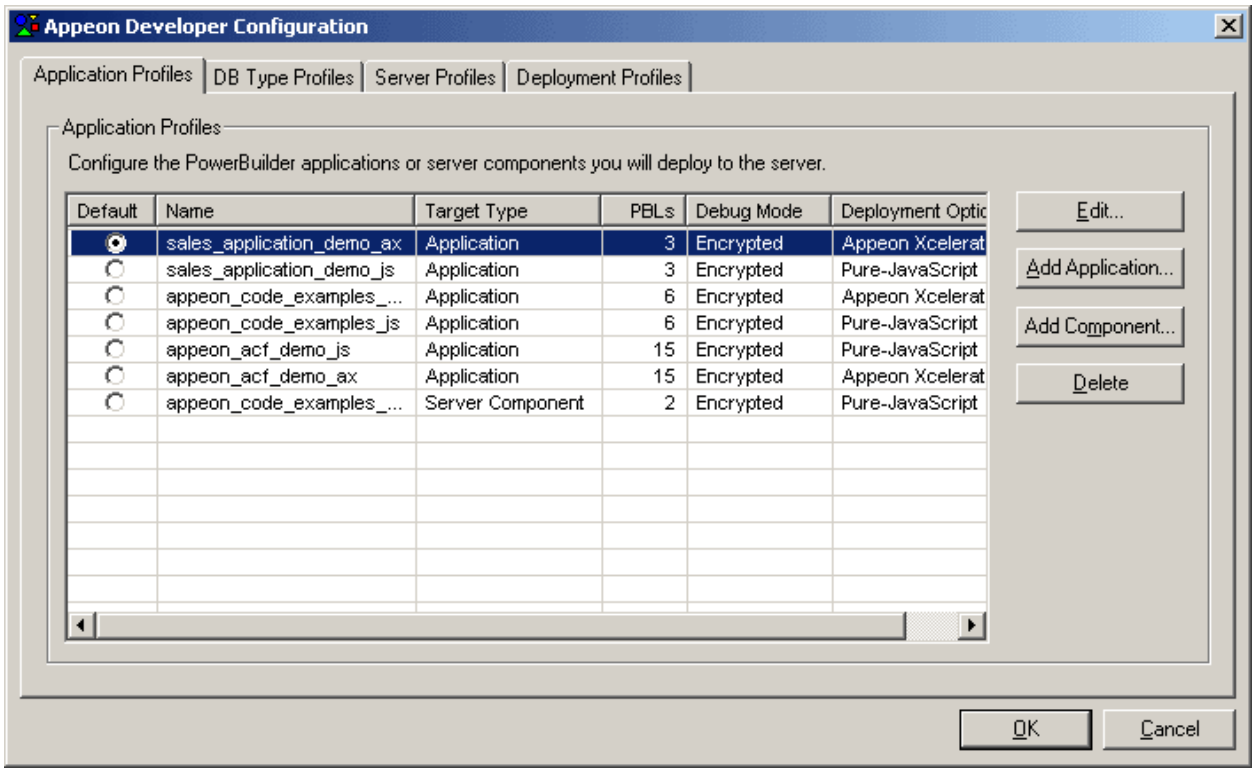
Application profiles provide important information about an application to be used in the PowerBuilder-to-Web conversion. You must configure an application profile for each application.

Each application profile has a unique application name. Before you create an application profile, verify that the PBLs for that application do not contain an application object with the same name as the application object already linked to an existing application profile.

4.2.1 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 4-1.

Figure 4-1: Appeon Developer Configuration Window



4.2.1.a Adding application profile or adding component profile

There are two types of profiles: 1) the application profile is the same as the profile created in previous versions, 2) the component profile is a new feature. Create an application profile and/or a component profile according to the following different situations:

If you want to configure…	And if you want to use the following deployment option…	You should…
Normal PowerScript applications. For example, non-distributed applications, or distributed applications **without** distributed DataWindows.	Appeon Xcelerator or Pure-JavaScript For more about Appeon Xcelerator and Pure-JavaScript deployment options, refer to the "*Choose a deployment option: Pure-JavaScript or Appeon Xcelerator*" section.	Click *Add Application* to create an application profile. Creating an application profile is necessary for any type of application and deployment option.
Distributed applications **with** distributed DataWindows.	Pure-JavaScript	1) Click *Add Application* to create an application profile for the client application in your distributed application. 2) Click *Add Component* to create a component profile for the server application in your distributed application.
	Appeon Xcelerator	Click *Add Application* to create an application profile. Note: You do not need to create the component profile.

The configuration of the two types of profiles is identical except that the configuration of the component profile is much simpler; you need only configure the *Basic Settings* and *DB Type* tabs.

4.2.1.b 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, runtime reports configuration and generation, application packaging, and application undeployment.

4.2.1.c Selecting a debug mode

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

Table 4-1: Debug mode

Debug Mode	Description	Use In The…
Debug PS/JS	Generates unencrypted JavaScript files for debug use. The PowerBuilder source code is provided as comments in the JavaScript files for easy reference.	Web application debugging and tuning stage.
Debug JS	Generates unencrypted JavaScript files for debug use. No PowerBuilder source code is provided in the JavaScript files.	Web application debugging and tuning stage.
Encrypted	Generates encrypted JavaScript files.	Production stage.

4.2.1.d Deleting an application profile

Click the *Delete* button to delete a selected application profile. Note that 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 will have the same name as the application profile. If the application is deployed using the Appeon Xcelerator option, the folder is created in the \AXProject folder, otherwise, it is located in the \JSProject folder. These folders are located under the root directory where Appeon Developer is installed (e.g. C:\Program Files\Appeon\Developer\). If the application is deployed with both options, you should delete the folders in both the \AXProject and \JSProject folders.

4.2.2 Configuring application profiles

The *Application Profiles Configuration* window provides four tab pages for you to specify the required application information to be used in the PowerBuilder-to-Web conversion:

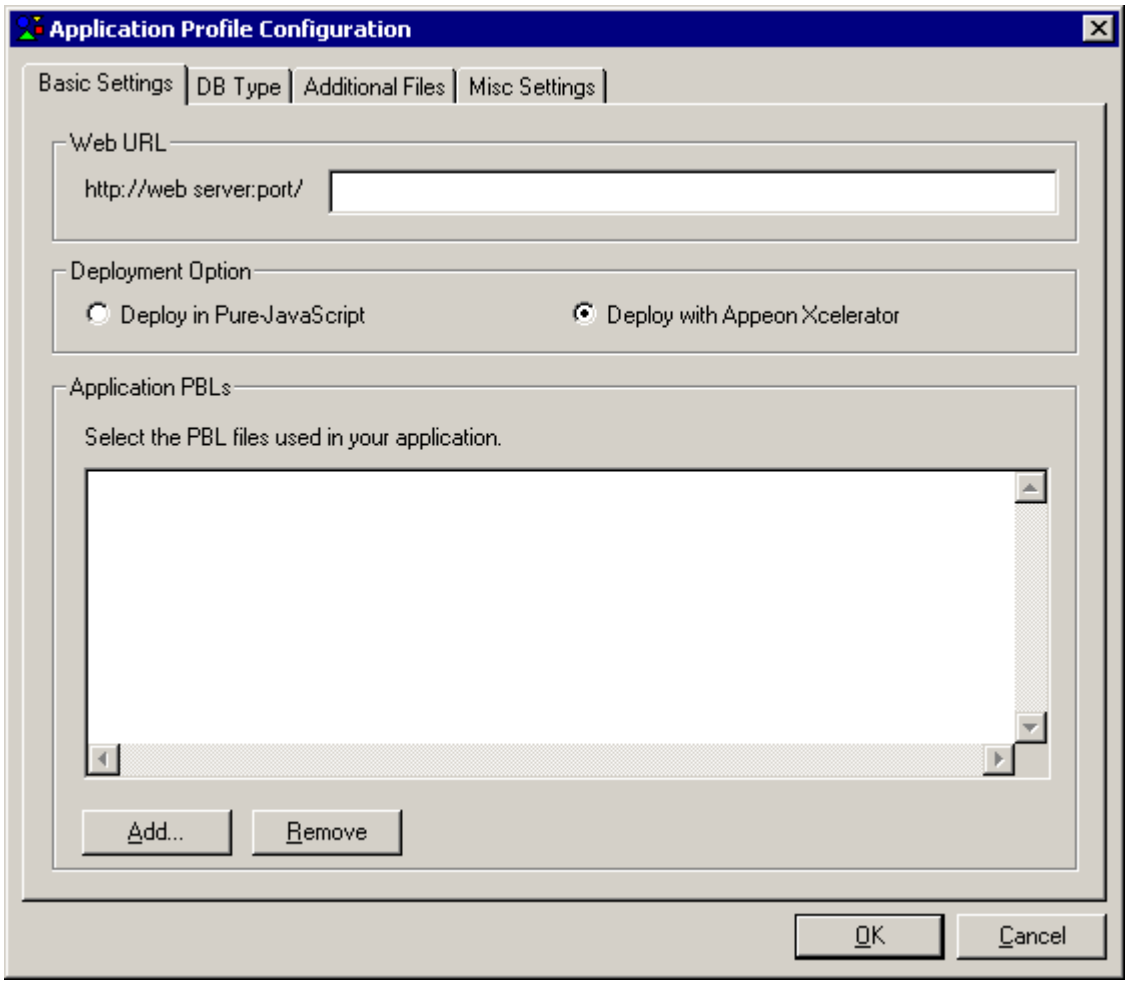
Table 4-2: Application Profiles settings

On this tab	You can …	To make changes effective, you need to …
Basic Settings tab	Specify the Web deployment path, deployment option, and source code location of an application. Refer to Section 4.2.2.a for detailed instructions.	Perform a full or incremental deployment on the application.
Database Type tab	Specify the types of databases that the application uses. Refer to Section 4.2.2.b for detailed instructions.	Perform a full or incremental deployment on the application.
Additional Files tab	Specify INI files, image files and DLL/OCX files that the application uses. Refer to Section 4.2.2.c for detailed instructions.	Perform a full or incremental deployment on the application.
Misc Settings tab	Specify command line arguments, application language, runtime performance, and log-writing mode. Refer to Section 4.2.2.d for detailed instructions.	Perform a full deployment on the application.

4.2.2.a Configuring Basic Settings tab

After you click *Add Application* or *Add Component* on the *Application Profiles* tab, the *Application Profile Configuration* window is displayed with the *Basic Settings* tab selected by default, as shown in Figure 4-2.

Figure 4-2: Basic Settings



Specify the Web deployment path

Specify the Web deployment path with a combination of letters, underscores ("_"), and numbers in the *Web URL* group box.

The specified Web deployment path is used as:

- a part of the Web URL to identify the Web application; and
- 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.

A Web deployment path 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 deployment path in the last deployment, and performs the runtime report generation, application packaging, and application undeployment based on the Web files generated in the corresponding Web folder.

Different from the Web files, the DataWindow syntax, profile, and registry information are stored in the *appeondb* database on Appeon Server, and the DataWindow syntax stored on Appeon Server will always be one version that is deployed to Appeon Server in the last deployment for the application profile.

Choose a deployment option: Pure-JavaScript or Appeon Xcelerator

Appeon provides two options for deploying PowerBuilder applications: Pure-JavaScript deployment and Appeon Xcelerator deployment.

Deployment option	Advantages	When to use it
Pure-JavaScript	Supports many PowerBuilder features and 85% of standard PFC features.	1) Develops new Web applications, or 2) Converts small and medium-sized PowerBuilder applications.
Appeon Xcelerator	1) Supports more Client/Server features that generally cannot be supported with JavaScript. 2) Boosts the runtime performance of Appeon Web applications to levels approaching PowerBuilder Client/Server. 3) Requires a minimal amount of work for the conversion process.	Converts large and complex Client/Server-centric applications.

Note: If you intend to deploy an application using both options to the same server, you should define the application with a different application object and then specify different Web deployment paths in the Web URL group box for the application.

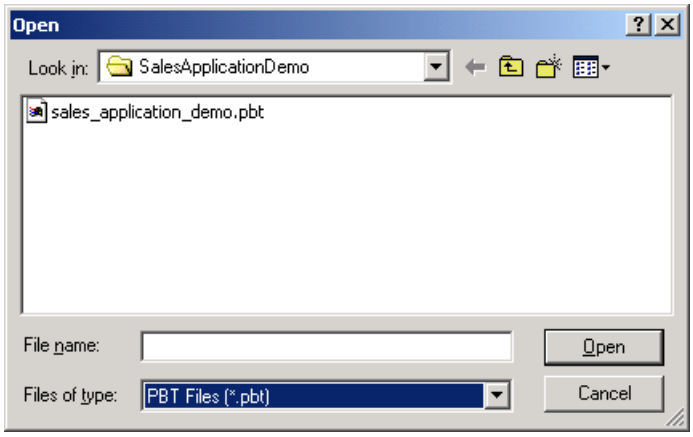
Add the application PBL files

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

Method 1 – 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 or of mistakenly adding a PBL twice.

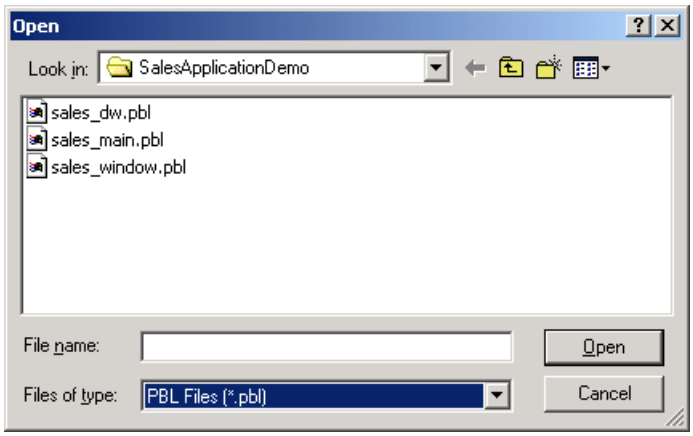
Figure 4-3: Open



Method 2 – 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.

Figure 4-4: Open



Method 3 – Copy and paste (Ctrl+C and Ctrl+V) the path list of PBLs into the Application PBLs field.

Important note:

When application PBLs are added into a profile, Appeon Developer does not verify the completeness of the PBLs list. Make sure that the PBLs being added are a complete list of all the PBLs that make up the PowerBuilder application. Stick to the following rules:

- There must be only one Application object in total for all the PBLs being added.
- Do NOT add unnecessary PBLs that do not belong to the application.
- Do NOT exclude any PBLs that belong to the application.
- Do NOT add the same PBLs repeatedly to the list.

If these precautions are not adhered to, the PBLs list may be corrupted and unpredictable errors may occur during Web deployment in the future.

4.2.2.b Configuring DB Type tab

The DB Type settings are required if the application configured connects to one or more databases. In the *DB Type* tab, you need to specify the types of databases that the application uses, but it is not necessary to specify the actual databases that the application uses.

What the DB 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 application connects to. The DB 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 4-5.

Figure 4-5: Database types

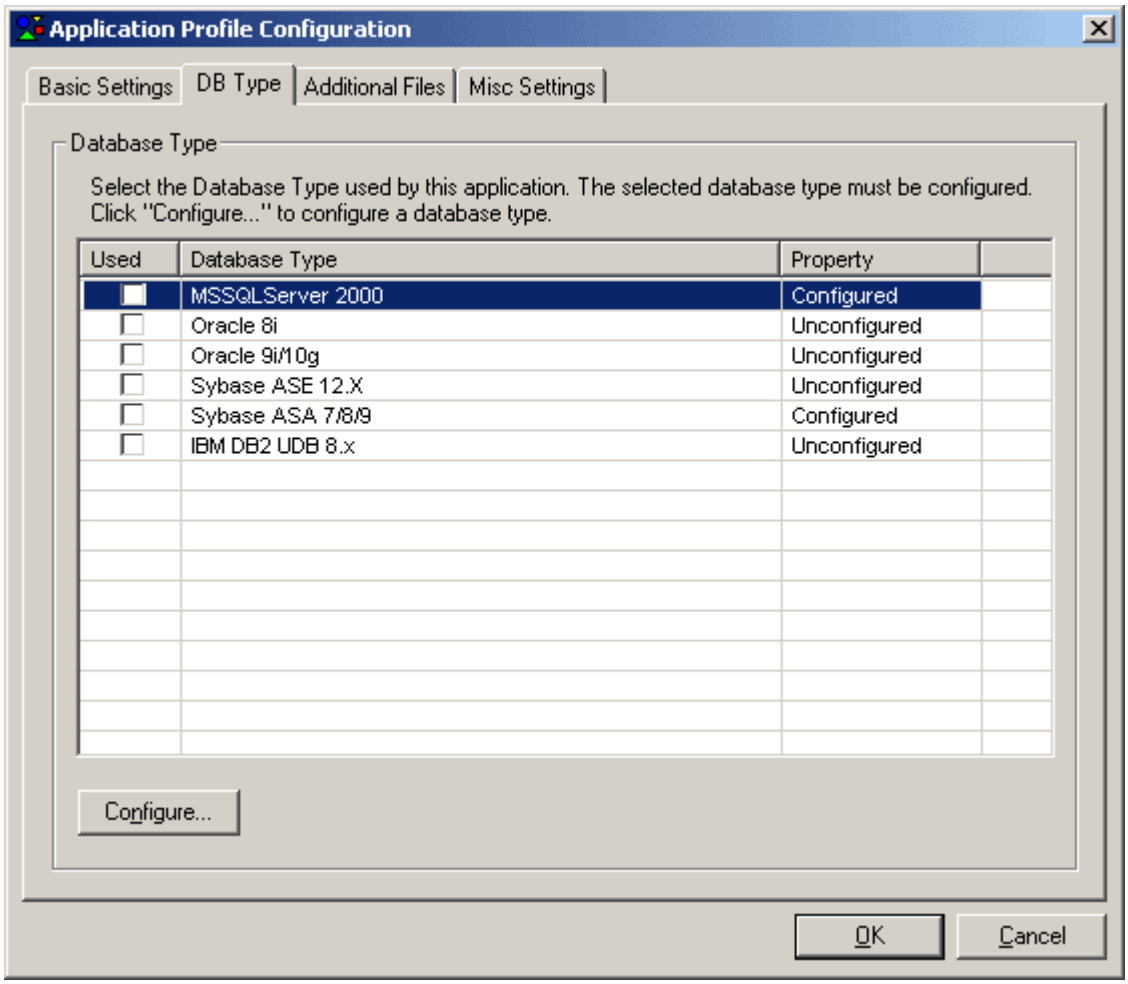


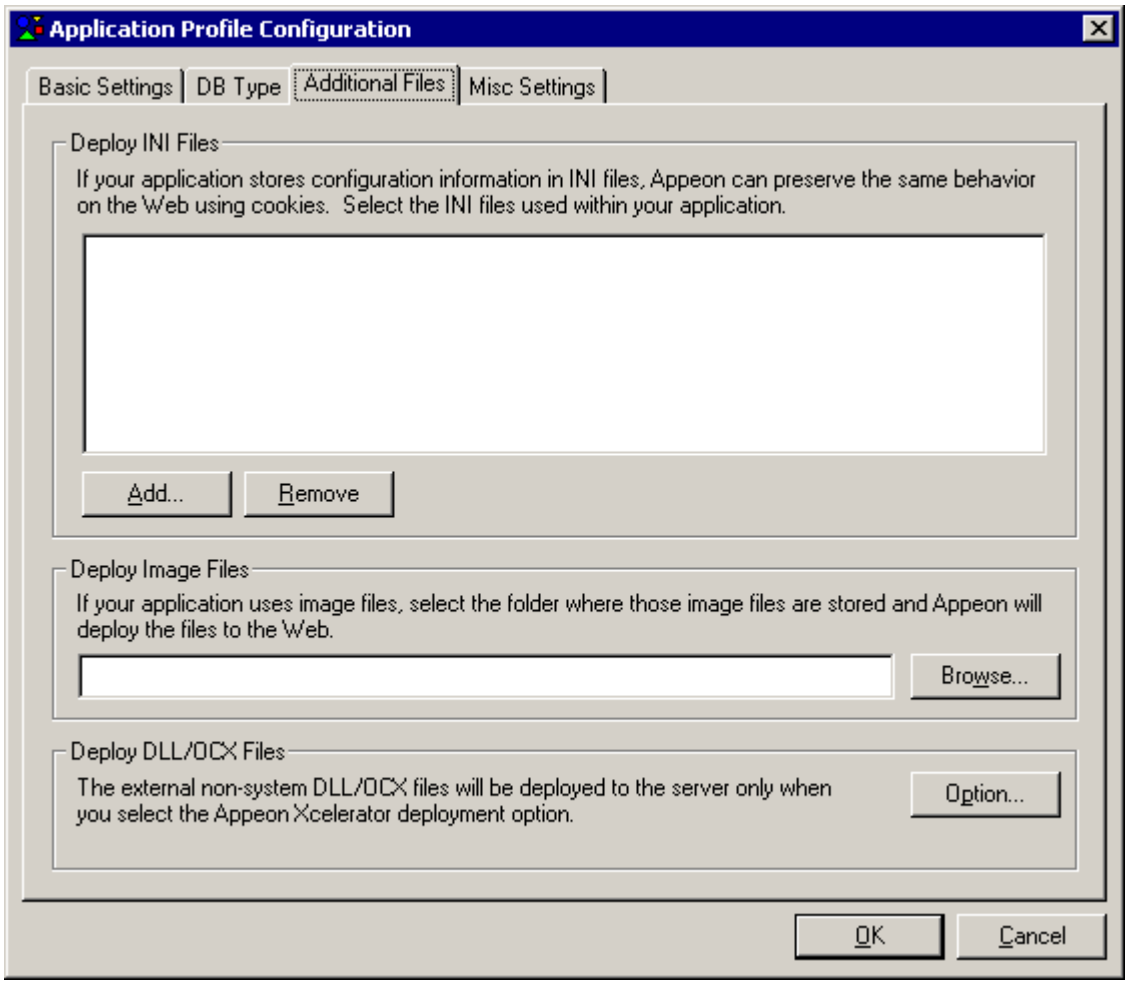
Table 4-3: Database type tab page

Column	Description
Used	You can click the checkboxes to select the database type(s) used by the application. More than one database type can be selected, but only "configured" database types are selectable.
Database Type	This column lists all the database types that are supported by Appeon.
Property	This column is identical to the Property column in the DB Type Profiles tab. It indicates whether a profile has been configured for the database type. If the Property column of the desired database type is "Unconfigured", 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 Property column in both this tab and the *DB Type Profiles* tab changes from "Unconfigured" to "Configured".

4.2.2.c Configuring Additional Files tab

"Additional files" refer to the files that are outside the application source code and are necessary for running the PowerBuilder application, including INI files, image files, and DLL/OCX files. If any such additional files are used in your application, specify them in the *Additional Files* tab.

Figure 4-6: 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 Enterprise Manager User Guide*.

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 must specify them here, so that they can be deployed to the Web Server, and then automatically downloaded to the Client when the Web application is run.

Click *Options* in the *Deploy DLL/OCX Files* group box to open the Deploy DLL/OCX Files dialog box (as shown in Figure 4-7) and select the required DLL/OCX files to be deployed.

Note that to deploy the required DLL/OCX files to the Web Server, you must select the Appeon Xcelerator deployment in the *Basic Settings* tab; otherwise, the *Option* button will be disabled. In this case, the required DLL/OCX files must be manually copied to the *%SystemRoot%\system32\AppeonPlugin* folder at each Client, instead of being deployed to Web Server first and then automatically downloaded to the Client. %SystemRoot% indicates the folder where Windows is installed (for example, C:\WINNT).

Figure 4-7: DLL/OCX Automatically Downloading

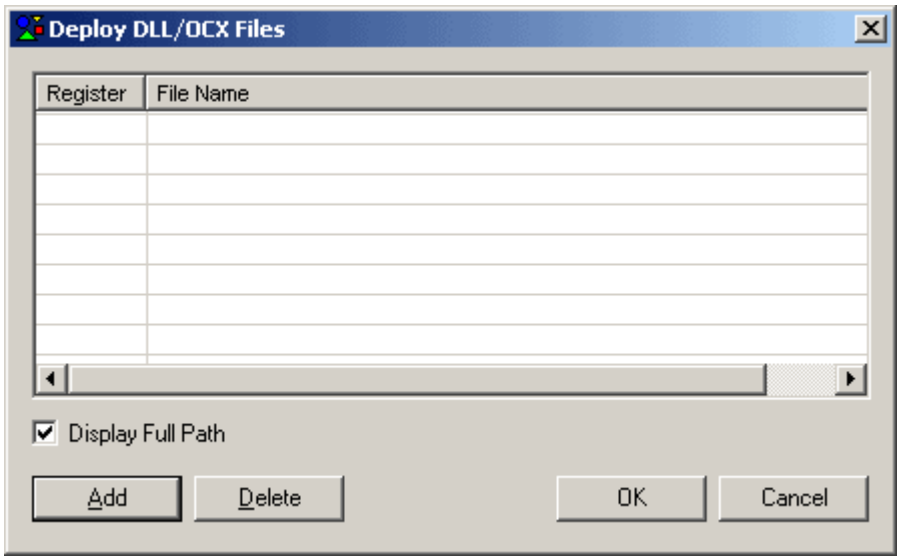


Table 4-4 introduces the elements on the Deploy DLL/OCX Files dialog box.

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

Element	Description
Register column	Allows you to select whether the file will be automatically registered after it is downloaded to the Client at runtime. If you choose not to automatically register a file, you need to register it manually after it is downloaded.
File Name column	Lists the added files. Click the header of this column to display the files in alphabetical order.
Display Full Path option	Enables you to show or hide the file location before the file name.

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

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

To Do This	Do This
To add files	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.
To remove files	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.
To save the settings	Click *OK*. Appeon will copy the files to a temporary folder and then deploy them to the Web Server during deployment.

How the deployed DLL/OCX files work on Web

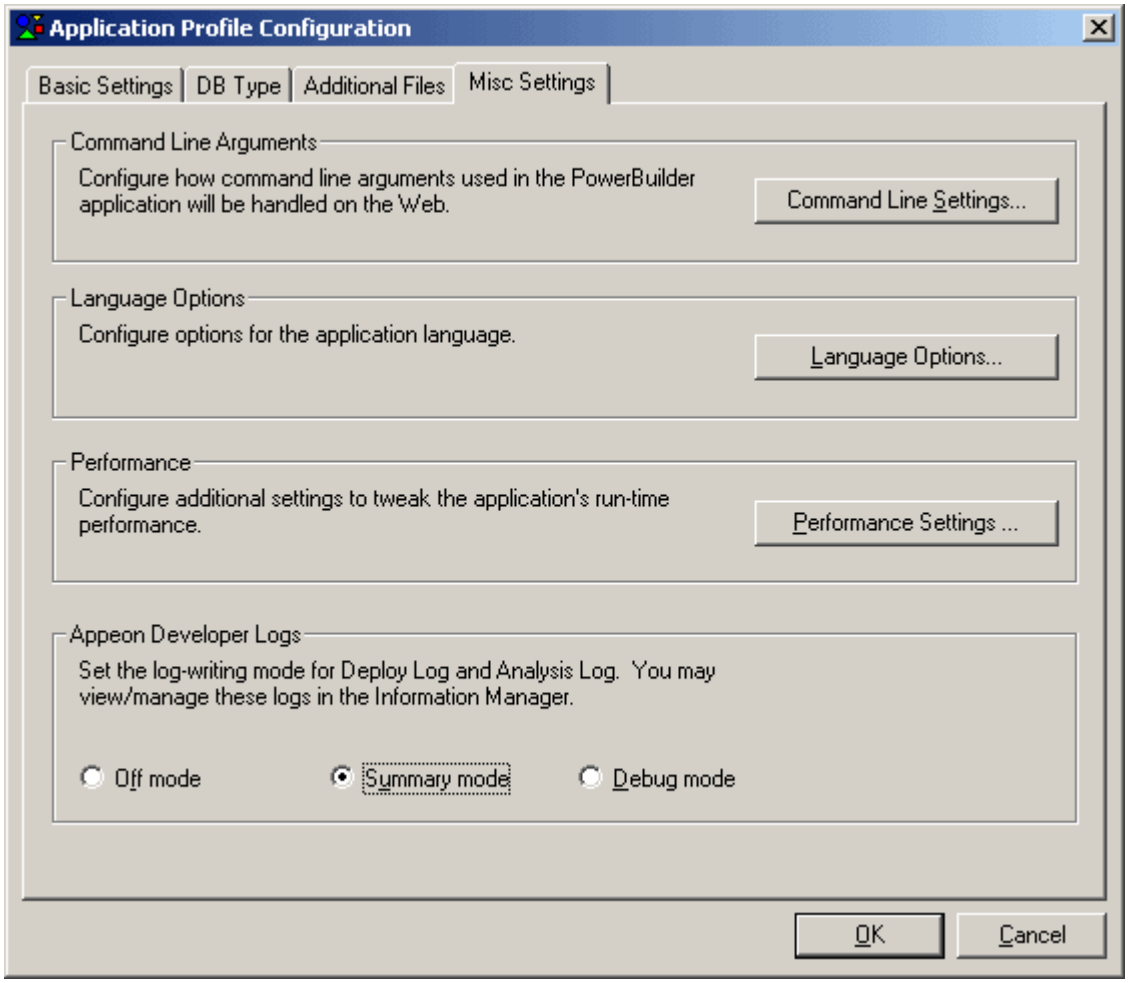
To successfully call the deployed DLL/OCX files on the Web application, you need to make sure that the DLL/OCX files are successfully downloaded and installed to the Client during the initial run of the deployed Web application.

Appeon Enterprise Manager (AEM) provides tools for manually or automatically downloading and installing the files. The files should be saved to the *%SystemRoot%\system32\AppeonPlugin* folder at the Client. *%SystemRoot%* indicates the folder where Windows is installed (for example, C:\WINNT). For detailed instructions, refer to the *Appeon Enterprise Manager User Guide*.

4.2.2.d Configuring Misc Settings tab

The *Misc Settings* tab page offers settings for command line arguments, application language, runtime performance, and log-writing mode.

Figure 4-8: Misc settings



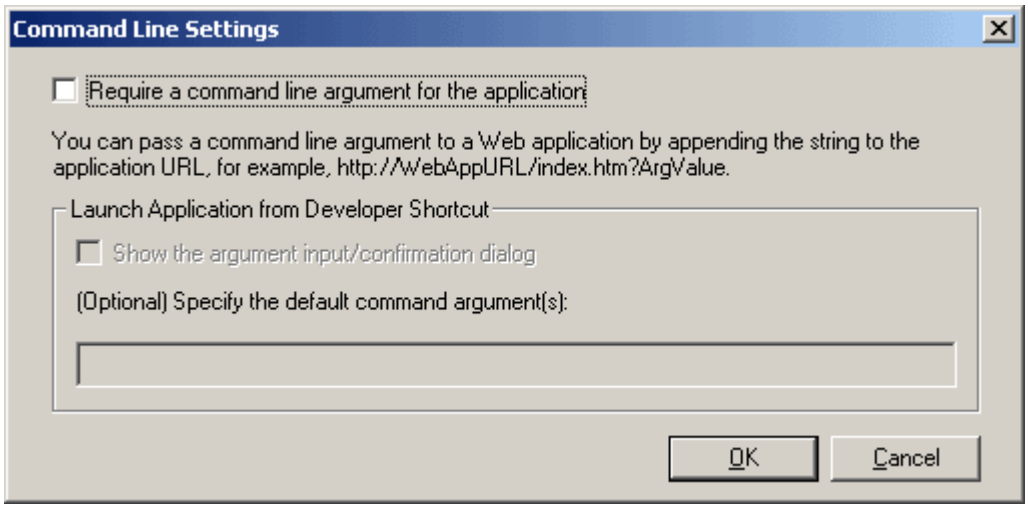
Specify Command Line Arguments

If an application needs to use command line arguments, enable the command line argument option, as shown in Figure 4-9.

To enable the command line argument option, perform the following steps:

Step 1 – Click *Command Line Settings* on the *Misc Settings* tab. The Command Line Settings dialog box is displayed, as shown in Figure 4-9.

Figure 4-9: Command Line Settings dialog box

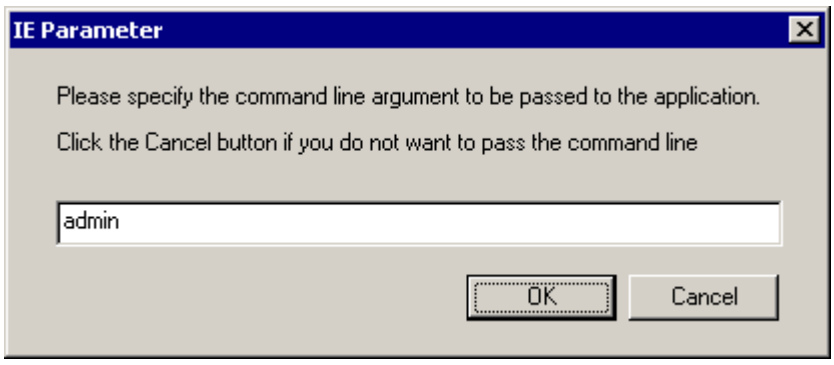


Step 2 – Click the *Require a command line argument for the application* check box.

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 4-10. You can either change the default arguments or leave them alone.

Figure 4-10: Command line argument dialog box



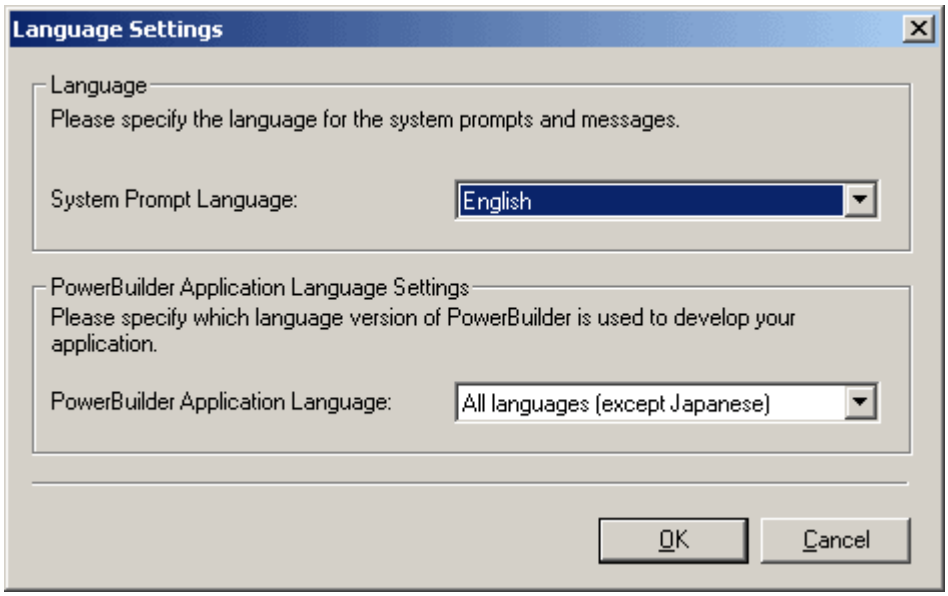
To run a Web application with arguments from Clients, type the application URL with the arguments in the Internet Explorer address bar.

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.

Configure Application Language

Step 1 – Click *Language Options* on the *Misc Settings* tab. The Language Settings dialog box is displayed, as shown in Figure 4-11.

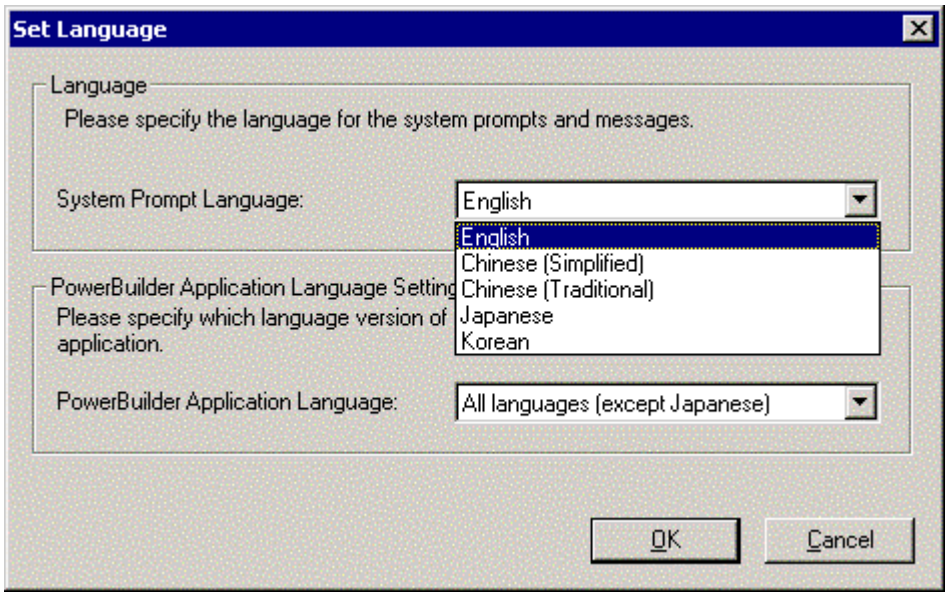
Figure 4-11: Application language settings



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. There are five languages (as shown in Figure 4-12) to choose from for the system prompts and messages. English is selected by default.

Figure 4-12: System Prompt Language

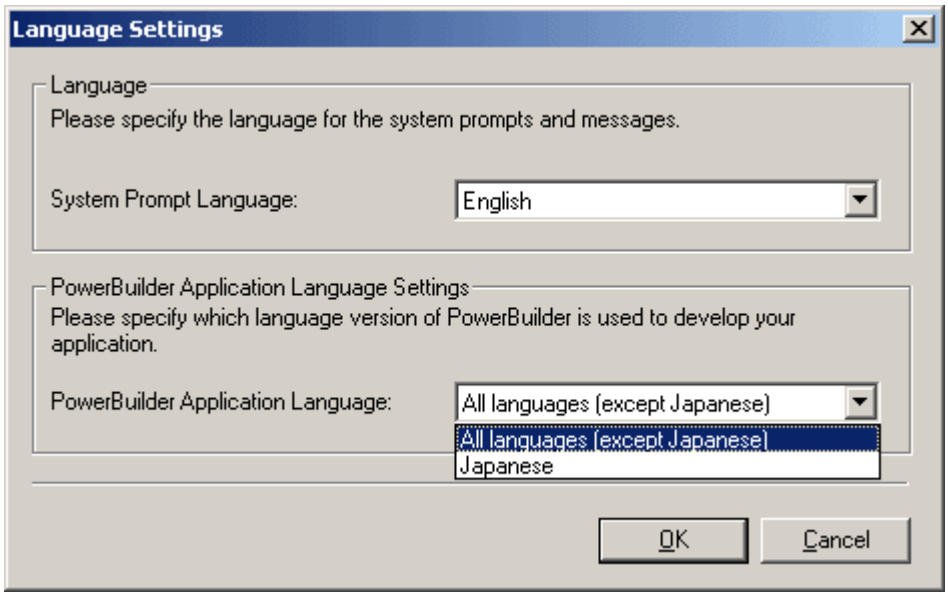


Step 3 – Select the language from the *PowerBuilder Application Language* dropdown listbox that is identical to the language version of PowerBuilder used for developing the application, as shown in Figure 4-13.

The *PowerBuilder Application Language* option determines the correct conversion formula for the PowerBuilder units (PBUs) of the application that will be used in the deployment of the application. If Japanese is selected, the Japanese PBUs conversion formula will be used; otherwise, the English PBUs conversion formula will be used.

If the selected formula is incorrect, the appearance of the windows displayed on Web will be different from that on the PowerBuilder application.

Figure 4-13: PowerBuilder Application Language

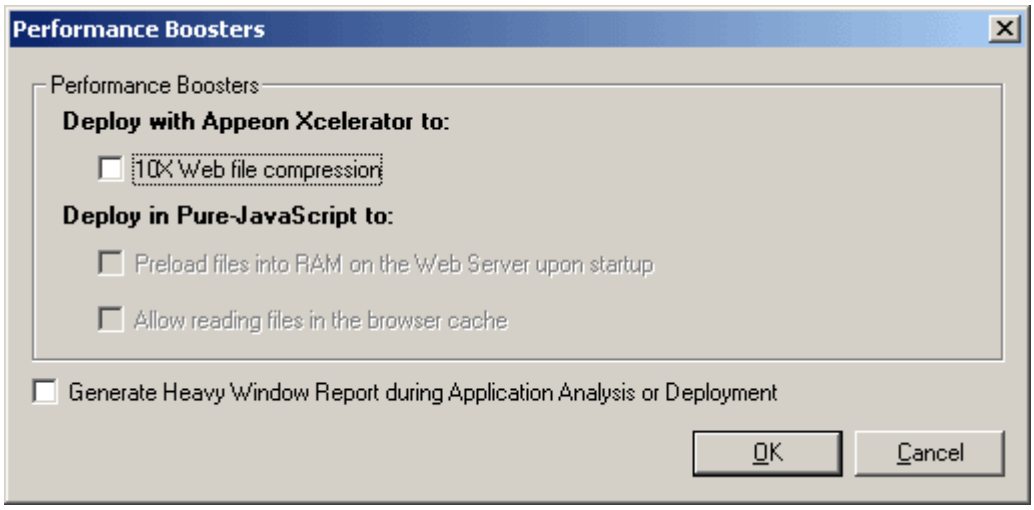


Specify Performance Settings

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* on the *Misc Settings* tab. The Performance Boosters dialog box is displayed, as shown in Figure 4-14.

Figure 4-14: Performance settings



Step 2 – Select the performance features in the Performance Boosters box.

Table 4-6 describes how to use the performance features.

Table 4-6: Performance features

Performance Feature	Description	Apply To…	When To Use…
10X Web File Compression	Compresses files when they are transferred over the network.	Appeon Xcelerator Deployment	Always
Preload files into RAM on the Web Server upon startup	Enables all Web files to be loaded into RAM on the Web Server when the server is started. Each time a Client visits the application, the Web files in the server's RAM will be used, instead of the Web files on the server's hard disk.	Pure-JavaScript Deployment	In the production stage.
Allow reading files in the browser cache	Enables the Client to read Web files directly from the Internet Explorer browser cache. After the file has been downloaded the first time, the file can be cached in the browser. For subsequent requests to load the file, the Client can read it directly from the browser's cache, unless a newer version of the file is detected on the Web Server. This can greatly speed up the loading of files.	Pure-JavaScript Deployment	In debugging and production stages.

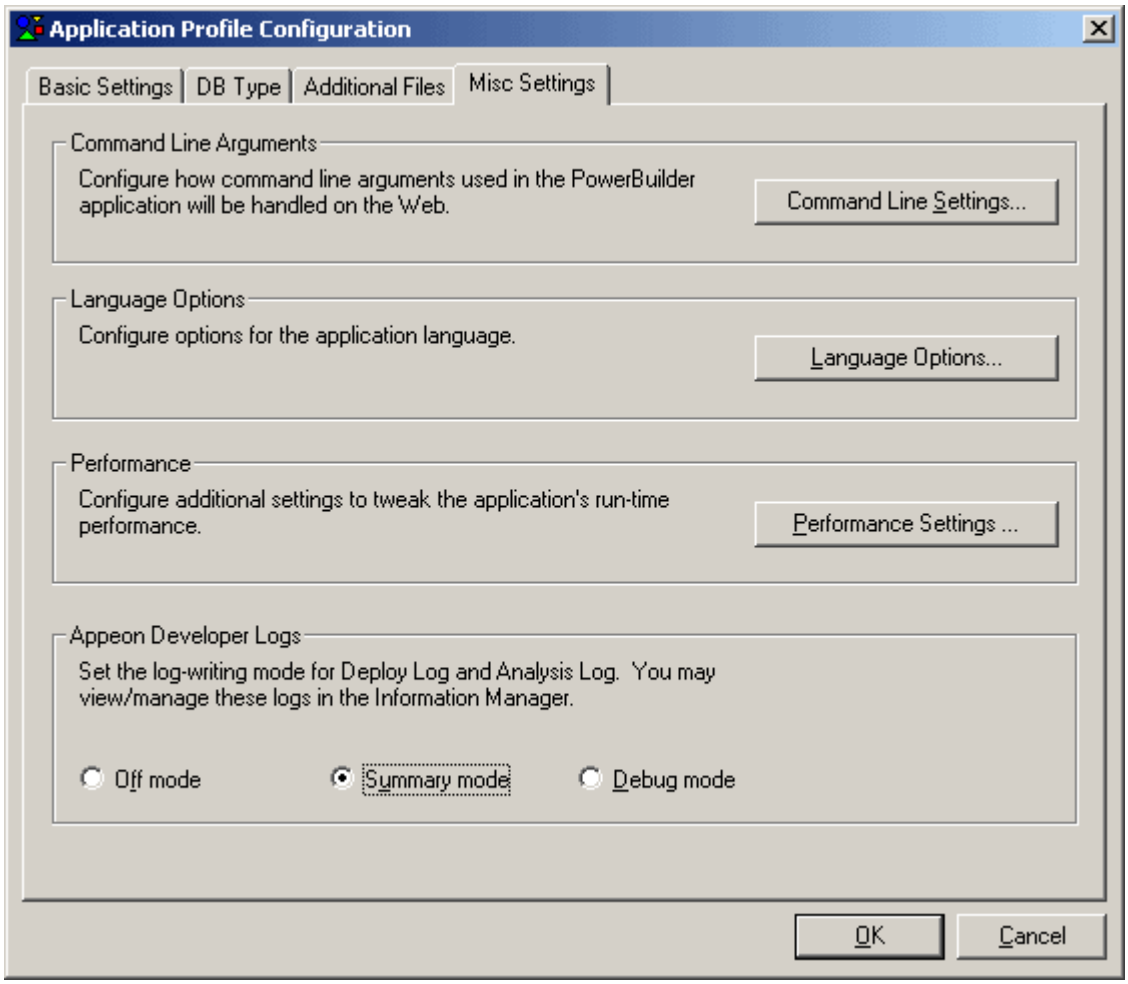
Step 3 – Check or uncheck the *Generate Heavy Window Report during Application Analysis or Deployment* option.

Checking or unchecking this option will also automatically check or uncheck the *Generate Heavy Windows Report* option in the Appeon Deployment Wizard, and vice versa. If either of them are checked for an application, a Heavy Window Report will be generated during a feature analysis or a Web deployment. For more details, please refer to Section 5: *Analyzing Unsupported Features*.

Specify Appeon Developer Log-writing 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 Logs box, as shown in Figure 4-15.

Figure 4-15: Log-writing mode



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

Mode	Description
Off	Generates no log files. This mode offers the fastest performance since nothing is written to a log file.
Summary mode (Default)	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.
Debug mode	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.

You can view or delete these log files. For detailed instructions, refer to Chapter 10: *Using Information Manager*.

4.3 Database type profiles configuration

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 enabling the profile in the DB Type 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" for both applications, although the connection information in the profile "Sybase ASA 7/8/9" 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 configure 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 4-16.

Figure 4-16: DB Type Profiles

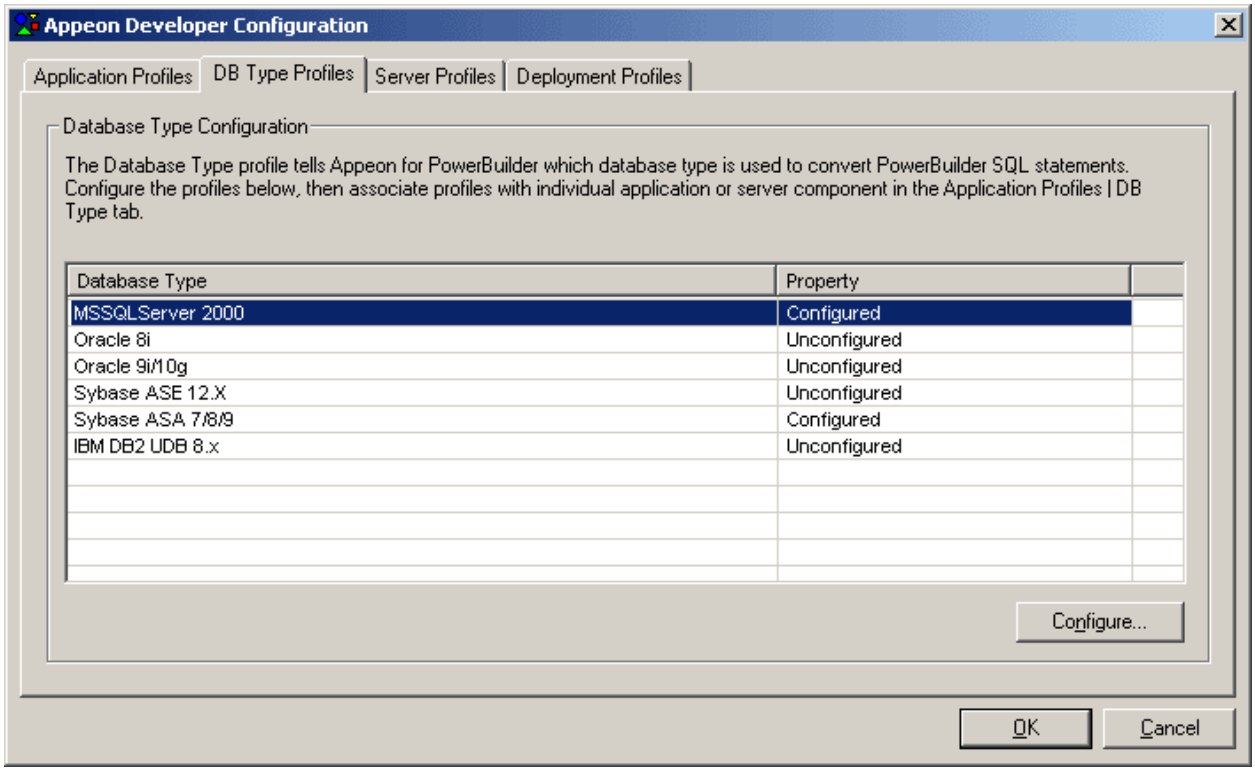
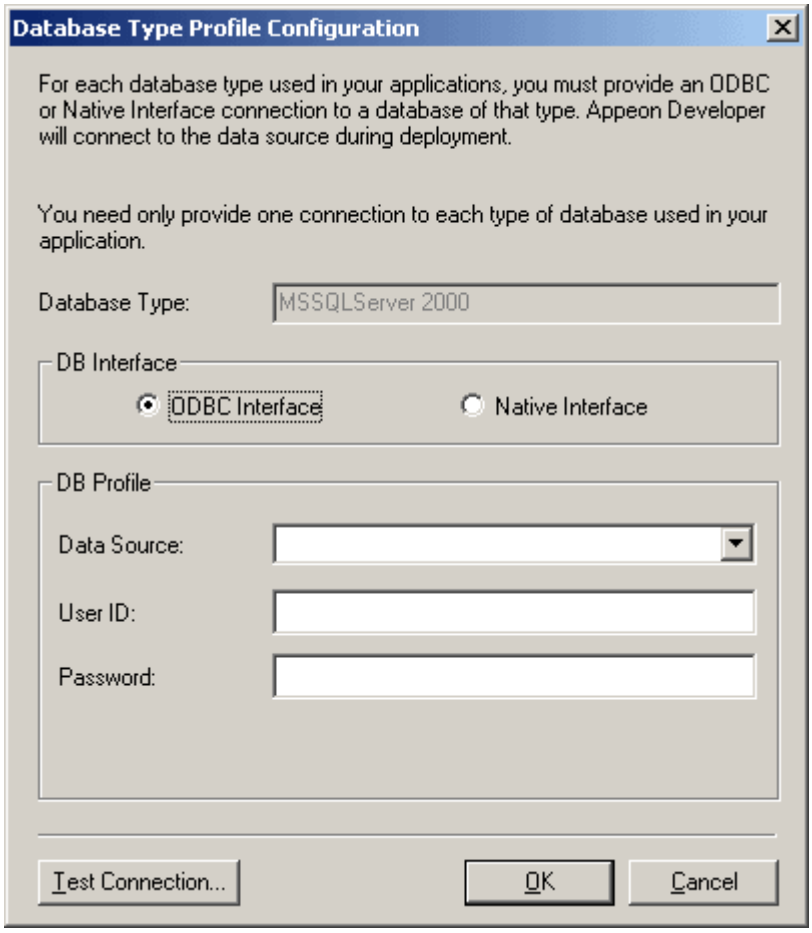


Table 4-8: DB Type Profiles tab

Element	Description
Database Type	Lists all the database types that are supported by Appeon.
Property	Indicates whether a database type profile has been configured or not.
Configure	Enables you to configure the profile for a selected database type.

Step 2 – Select an "Unconfigured" database type profile (for example, Sybase ASA 7/8/9), and click the *Configure* button to configure the database type profile (as shown in Figure 4-17).

Figure 4-17: Database Type configuration



Step 3 – Specify the information needed for setting up the database type profile.

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

Table 4-9: ODBC interface settings

Setting	Description
Data Source	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.
User ID	Type in a recognized (set) user name to login to the database. If no user name has been set, leave this field blank.
Password	Type in the password used to login to the database. If no password has been set, leave this field blank.

Native Interface – Select this option if you use a native database interface. Specify the server name/IP, User ID, Password, and the database name.

Note that not all the listed database types are supported by Appeon for the native database interface. If Sybase ASA 7/8/9 or IBM DB2 is your database type, the *Native Interface* option is disabled.

Table 4-10: Native database interface settings

Setting	Description
Server	Enter the machine name or IP address of the server where the target database resides.
User ID	Type in a recognized (set) user name to login to the database. If no user name has been set, leave this field blank.
Password	Type in the password used to login to the database. If no password has been set, leave this field blank.
Database	This option is only available for the MSSQL Server 2000 database type. Specify the name of the database that will be used by the application.

Step 4 – Click *Test Connection* to test the connection to the database.

4.4 Server profiles configuration

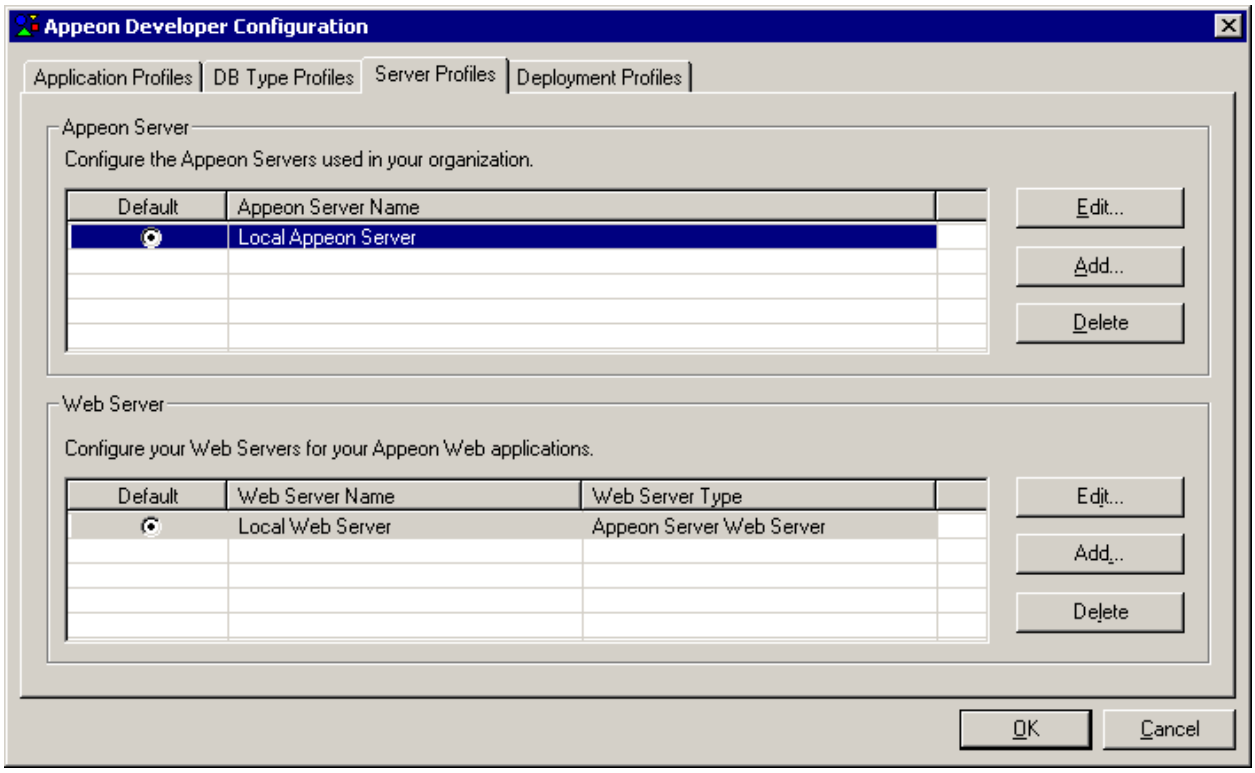
A server profile is a set of parameters for Appeon Developer to connect to, and deploy applications to, a particular Appeon Server or Web Server. Creating and using server profiles is the easiest way to manage your application deployments because you can:

- 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.

4.4.1 Server Profiles tab page

There are two main parts in the *Server Profiles* tab page, as shown in Figure 4-18. The upper part is for management of Appeon Server profiles, while the lower part is for management of Web Server profiles.

Figure 4-18: Server Profiles



4.4.1.a Default 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 quickly access them. Applications that are deployed to non-default server profiles are not available in the shortcut. For details about the shortcut, refer to Section 9.3.2: *Running Web applications*.

4.4.1.b 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 4.4.2 and 4.4.3 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.

4.4.2 Configuring Appeon Server profiles

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 4-19 shows the Appeon Server Profile Configuration window that displays when you click the *Add* button in the *Appeon Server* group box of the *Server Profiles* tab page.

Figure 4-19: Appeon Server Profile

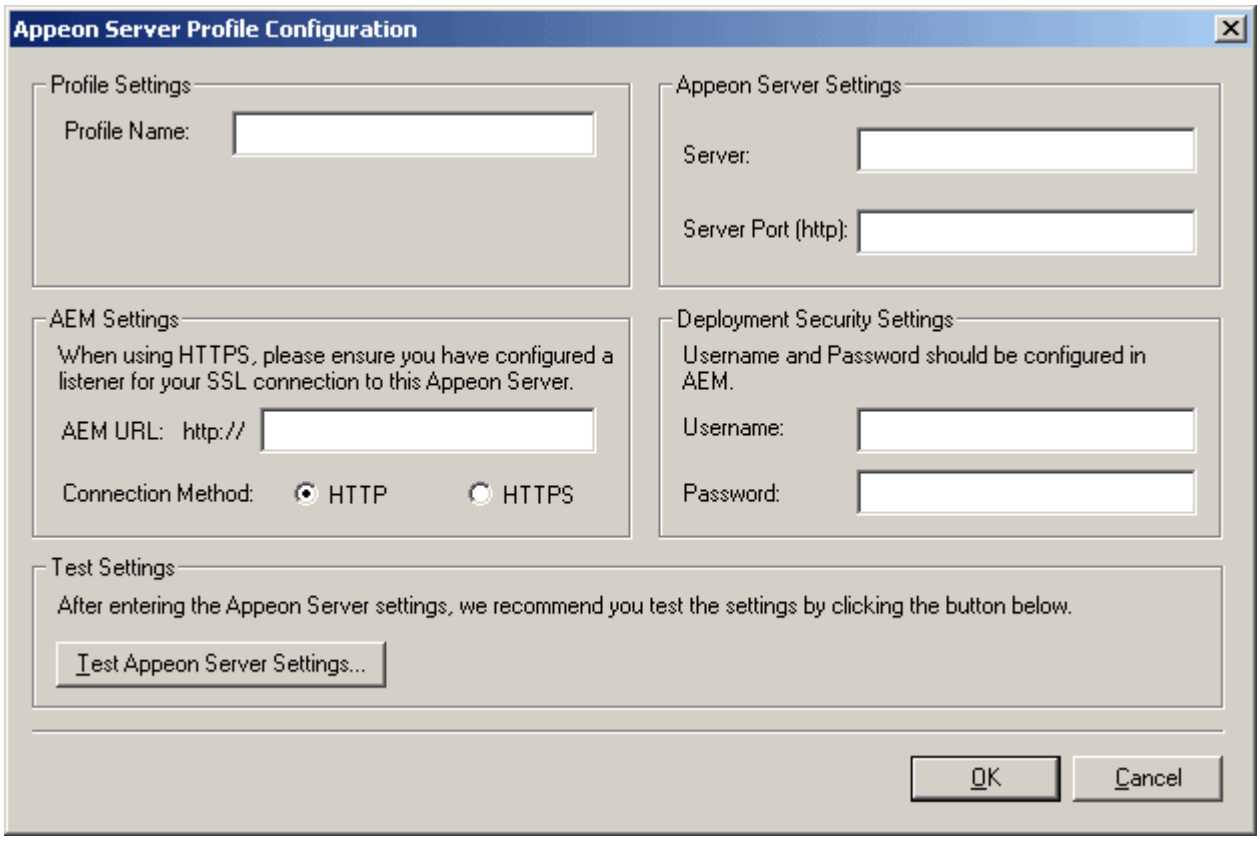


Table 4-11 lists the properties you should specify for an Appeon Server profile.

Table 4-11: Appeon Server profile properties

	Property	Description
Profile Settings	Profile Name	Assigns a name to the Appeon Server profile. You can use names that are easy to remember and identify (for example "Appeon Server for Test" or "Remote Appeon Server").
Appeon Server Settings	Server	Enables you to enter the IP address or the machine name of Appeon Server.
	Server Port (http)	Enables you to enter the HTTP port number used by Appeon Server.
AEM Settings	AEM URL	The URL for Appeon Enterprise Manager (AEM) will be automatically generated after you specify the Server and Server Port. The URL will be in the following format: http://*server:port*/AEM, (for example, http://localhost:9988/AEM). Note: 1) 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.

	Connection method	Connects to AEM using either HTTP or HTTPS. If the Web Server is configured as an SSL Web Server, check the HTTPS (secure) option. Otherwise, check the HTTP (insecure) option.
Deployment Security Settings	Username	Enables you to 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.
	Password	Enables you to 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.

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 get 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.

4.4.3 Configuring Web Server profiles

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

Figure 4-20 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 4-20: Configure the Web Server profile

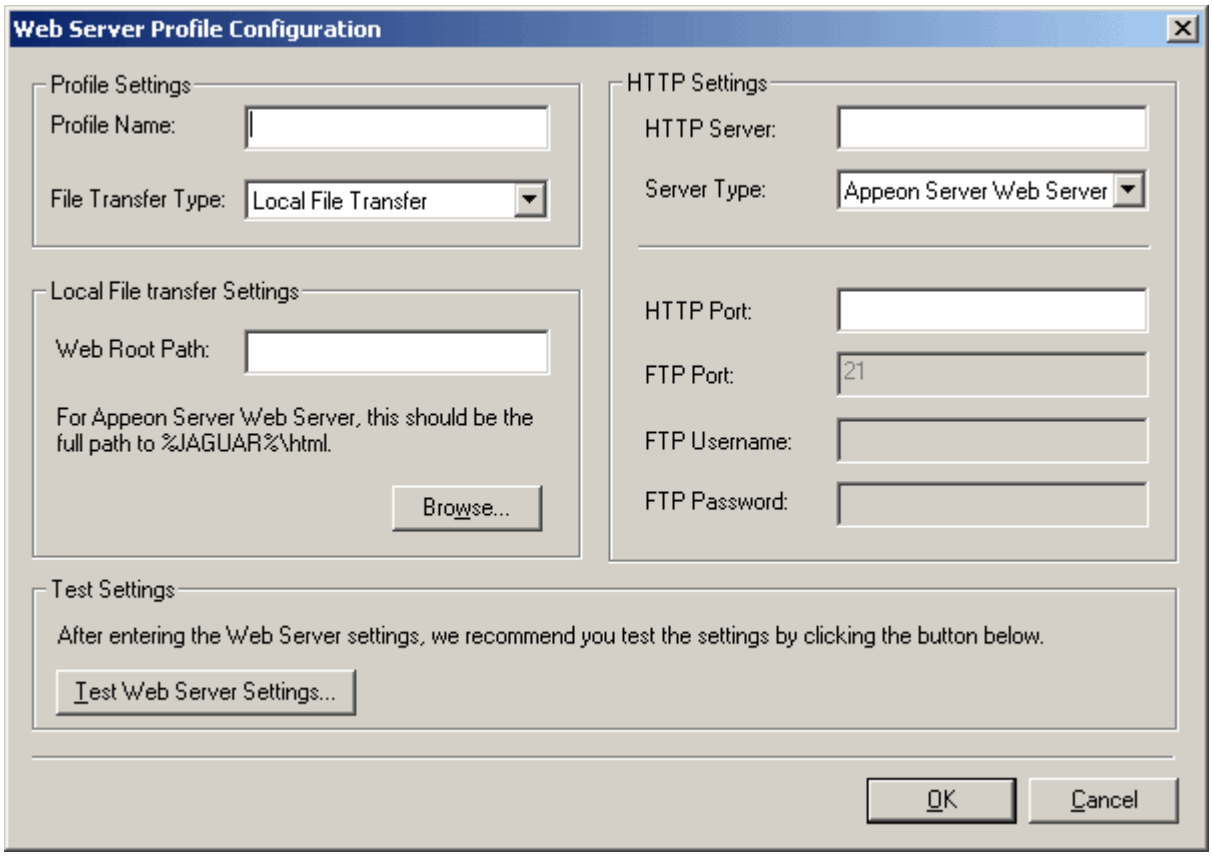


Table 4-12 lists the properties you should specify for a Web Server profile.

Table 4-12: Web Server profile properties

	Property	Description
Profile Settings	Profile Name	Assigns a name to the Web Server profile. You can use names that are easy to remember and identify such as "Web Server for Test" or "Production Web Server".
	File Transfer Type	Uses **Local File Transfer** if the Web Server is a local computer. Uses **FTP File Transfer** if the Web Server locates remotely.
HTTP Settings	HTTP Server	Enables you to enter the IP address or the machine name of Web Server.
	Server Type	Selects the Web Server type. Appeon supports three Web Server types: Apache 1.3/2.0, Appeon Server Web Server, and Microsoft IIS 5.0/6.0.
	HTTP Port	(For local file transfer only) Enables you to enter the Web Server port number. If the Web Server is an SSL Web server, enter an HTTPS port for this property.
	FTP Port	(For FTP file transfer only) Enables you to enter the FTP server port number. The typical FTP port is port 21.

	FTP Username	(For FTP file transfer only) Enables you to enter the username for FTP login. If the FTP server offers "anonymous" access then the username should be "anonymous".
	FTP Password	(For FTP file transfer only) Enables you to 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.
Local File Transfer Settings	Web Root Path	(For local file transfer only) Enables you to enter the home directory of the Web Server (for example: C:\Program Files\Sybase\EAServer\html). Click the *Browse* button to select the home directory of the Web Server.

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 Section 4.4.3.a: *Two requirements for FTP settings* to make sure the configuration for the Web Server profile is successful.

  If the Web Server is an SSL Web server, refer to Section 4.4.3.b: *If the Web Server is an SSL Web Server* to make sure the configuration for the Web Server profile is successful.

4.4.3.a Two requirements for FTP settings

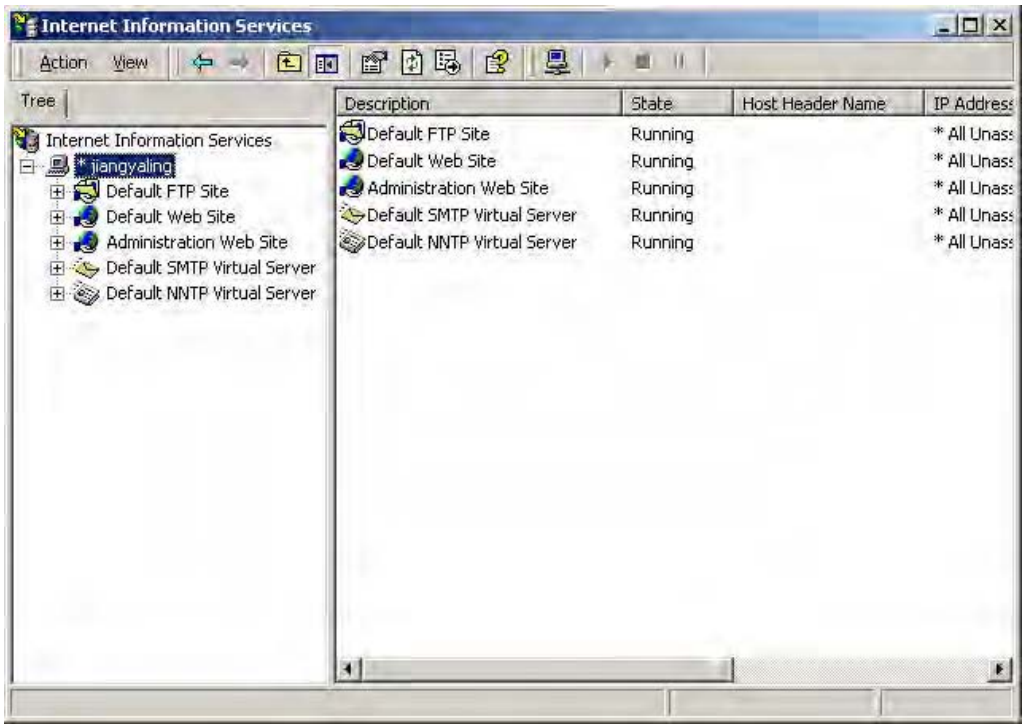
If you configure a Web Server profile for a remote Web Server, make sure the FTP settings in the Web Server profile meet the following two requirements:

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 4-21.

Figure 4-21: 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:

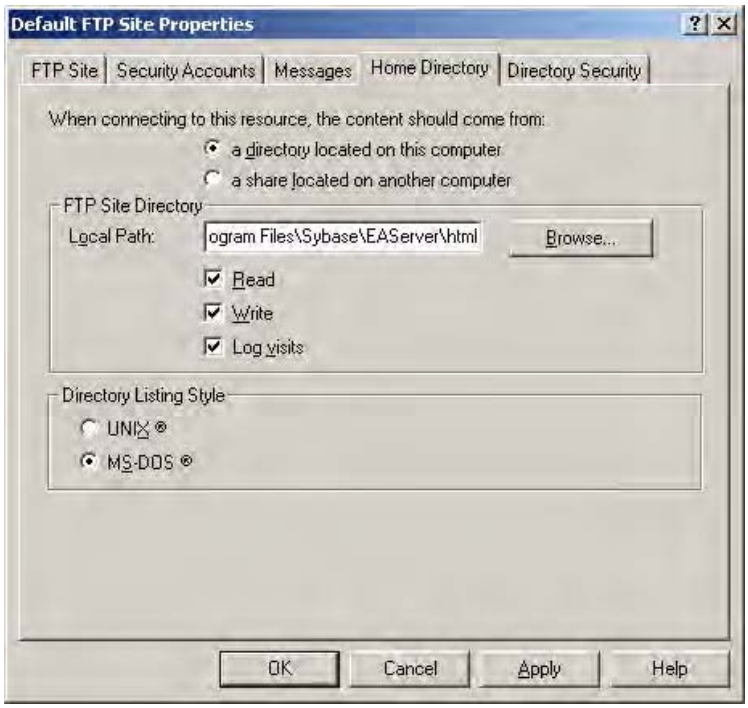
- The Local Path is the full path to the Web Server document root.

  If the Web Server is an Appeon Server Web Server, the local path is the full path to the EAServer document root: \%JAGUAR%\html (As shown in Figure 4-22).

  %JAGUAR% is the EAServer installation directory (for example: C:\Program Files\Sybase\EAServer).

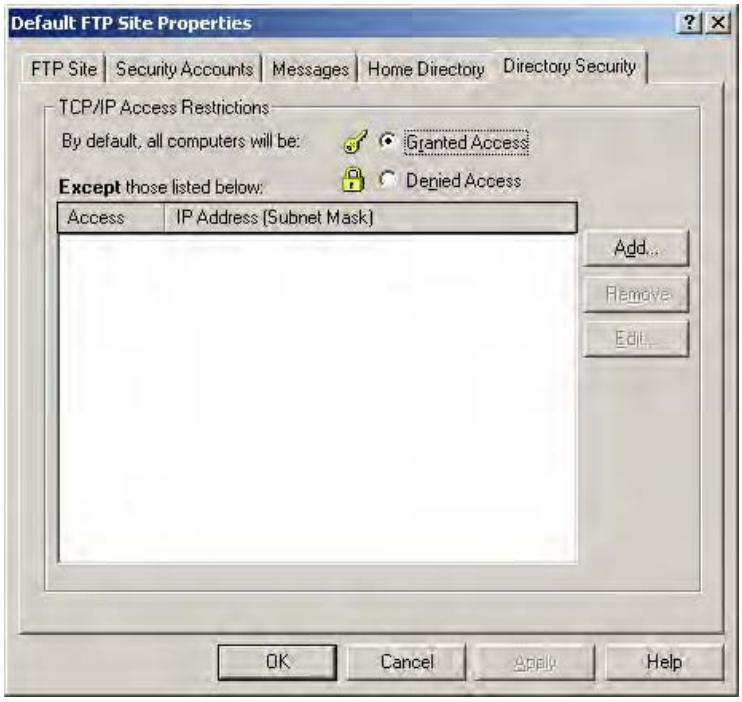
- The Write property of the FTP Site Directory is enabled.

Figure 4-22: FTP site properties



Step 4 – Go to the *Directory Security* tab and verify that the *Granted Access* option is checked, as shown in Figure 4-23.

Figure 4-23: Directory security



4.4.3.b If the Web Server is an SSL Web Server

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 4-12, except for the following two points:

1. You need to configure an HTTPS listener and port number for the Web Server.

2. You need to specify the HTTPS listener and its port number in the HTTP server settings of the Web Server profile.

If a Web application is deployed to an SSL Web Server, you need to input “https” instead of “http” at the beginning of the URL in order to access the application, (for example, https://192.0.0.80:8181/appeondemo/).

4.5 Deployment profiles configuration

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.

4.5.1 Deployment Profiles tab page

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

Figure 4-24: Deployment Profile tab page

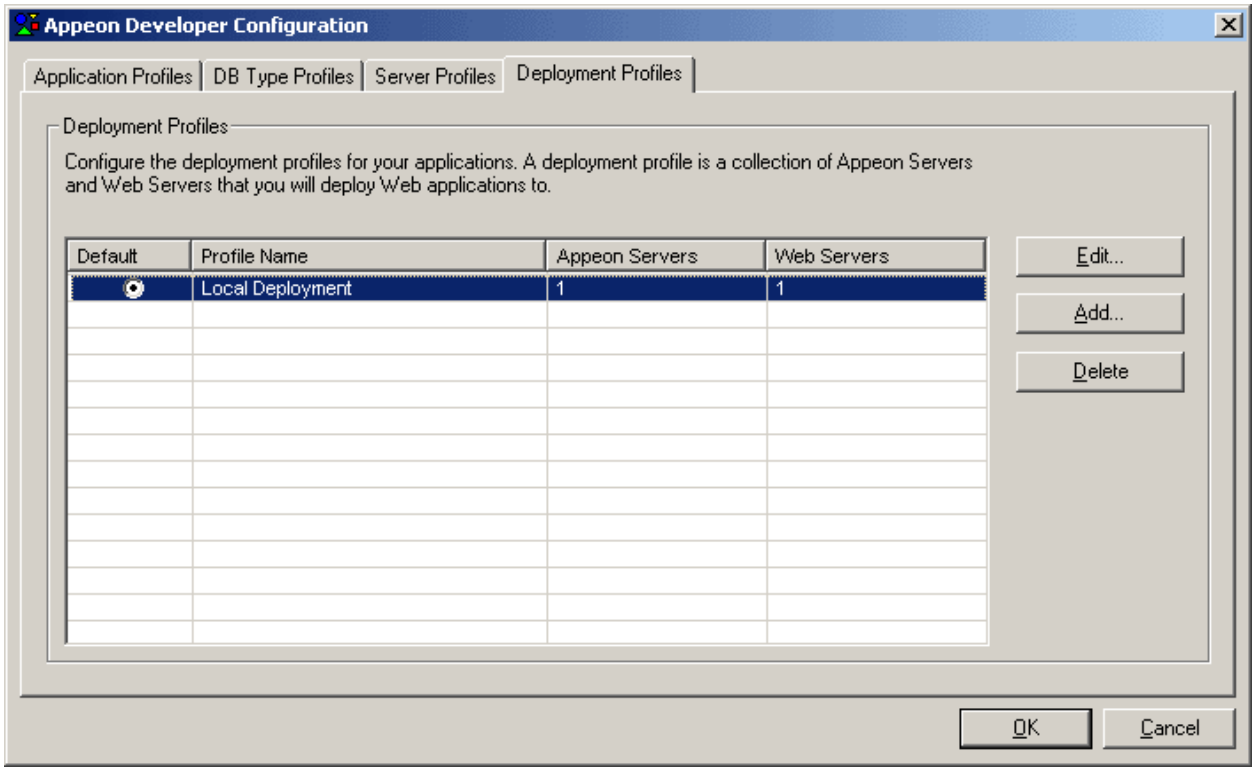


Table 4-13: Deployment Profile tab page

Use this button	To…
Default radio button	Specify the default deployment profile used for Web deployment.
Edit button	Modify an existing deployment profile.

Add button	Create a new deployment profile.
Delete button	Remove a deployment profile.

4.5.2 Configuring deployment profiles

In the Deployment Profile Configuration dialog box, set up the necessary configuration, as shown in Figure 4-25.

Figure 4-25: Deployment Profile

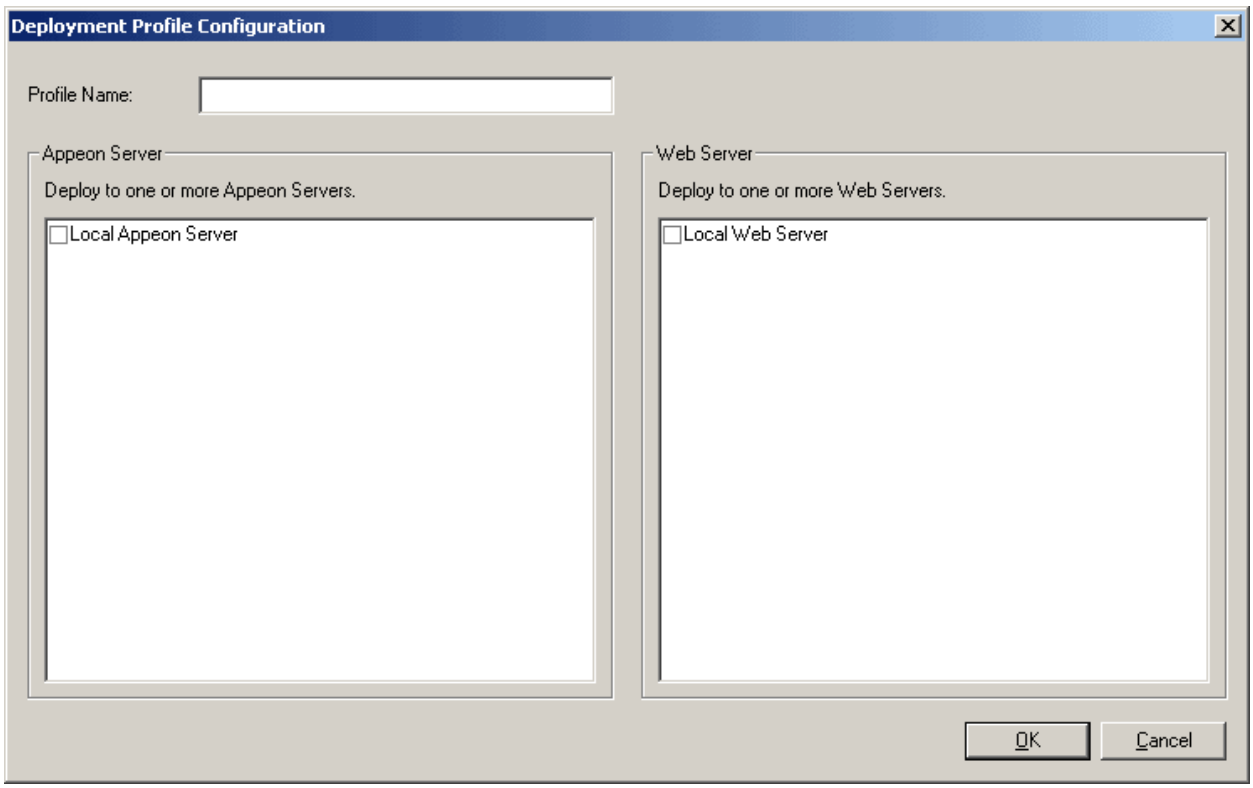


Table 4-14: Deployment Profile settings

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

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

5 Analyzing Unsupported Features

5.1 Overview

Unsupported Features Analysis generates two reports. 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 5-1: UFA Report and Heavy Window report

	UFA Report	Heavy Window Report
Purpose of the report	Enables you to work around or remove unsupported features in order to make the PowerBuilder application suitable for Appeon Web migration.	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.
What's reported	Unsupported PowerBuilder coding features.	Complex windows that most affect the performance of the deployed Web application.
How to generate the report	Use the Unsupported Features Analysis tool, or perform the first task in Appeon Deployment Wizard.	Use the Unsupported Features Analysis tool, or perform the first task in Appeon Deployment Wizard. Requirement: the *Generate Heavy Window Report during Application Analysis or Deployment* option must be enabled in the Application Profile.
Features	These features apply for both reports: 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. 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.	

5.2 Feature analysis on an application or objects

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 Section 3.1: *System Requirements* 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

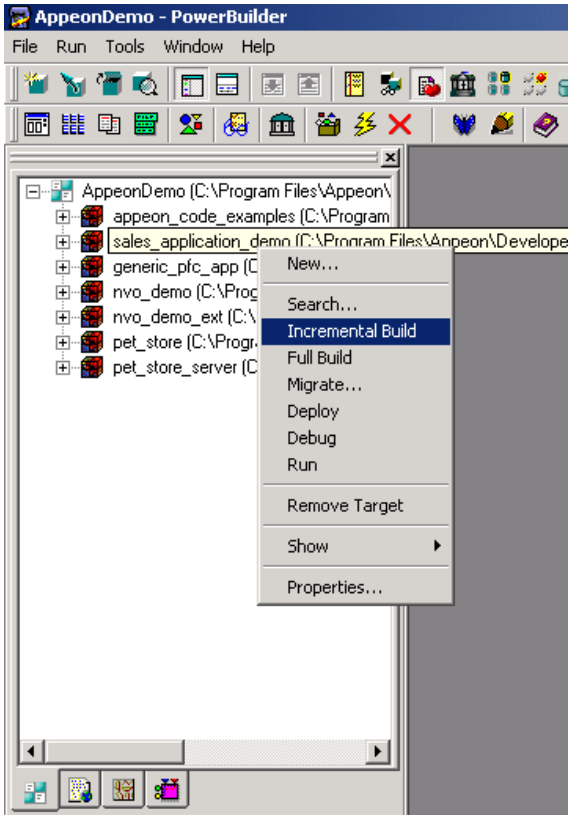


Table 5-2 shows you which type of build options you should select for the application.

Table 5-2: Recommended build options

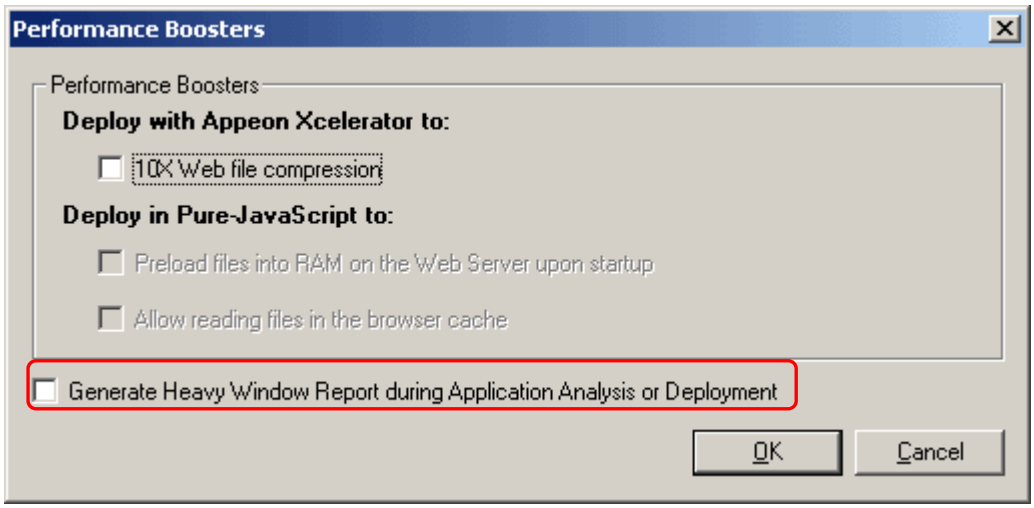
Build Option	What It Does	Recommended For
Regenerate	Refreshes the timestamp for a PowerBuilder object.	Object-level Features Analysis. If the object is regenerated, the object will be analyzed even if no change has made to it when a Features Analysis is run.
Incremental	Refreshes the objects that have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

Step 4 – Perform PBL optimization for the PowerBuilder application.

The purpose of PBL optimization is to remove gaps from libraries and defragment 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 during Application Analysis or Deployment* option in the Application Profile Configuration page, as shown in Figure 5-2.

Figure 5-2: Performance options



5.2.2 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 task 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 – Click the *Analyze* button ( ) in the Appeon Developer toolbar. The Select Objects to Analyze dialog box will appear, as shown in Figure 5-3.

Figure 5-3: Select Objects to Analyze dialog box

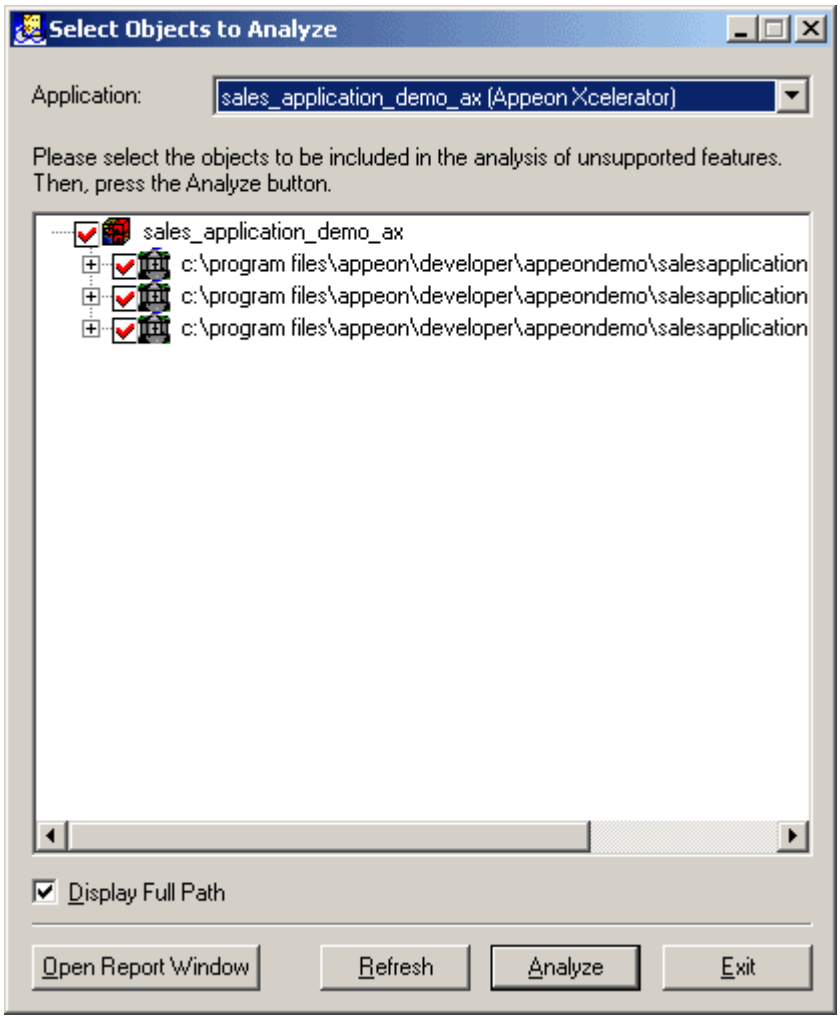


Table 5-3 gives a brief description of the elements in the Select Objects to Analyze window.

Table 5-3: Select Objects to Analyze window

Element	Description
Application dropdown listbox	Provides a list of application profiles for you to select and analyze. The application profiles listed and the deployment option given for each profile are the ones that are currently configured in the Appeon Developer Configuration window.
Treeview of objects	Gives the treeviews of PBLs and objects for the application that is selected in the *Application* dropdown listbox.
Display Full Path option	Gives you options to display or hide the full path of the PBLs and objects in the treeview.
Open Report Window button	Allows you to directly open the UFA Report Window (Figure 5-5) without performing the feature analysis.
Refresh button	Enables you to refresh the PBL and object list in the treeview.
Analyze button	Starts feature analysis.

Step 2 – Select the application that you want to analyze from the *Application* dropdown listbox.

- The default target to analyze is the default application profile in the Appeon Developer Configuration window.
- For one application profile, the UFA Reports generated for Pure-JavaScript deployment and Appeon Xcelerator deployment can be different due to the different supported feature sets of Pure-JavaScript deployment and Appeon Xcelerator deployment.

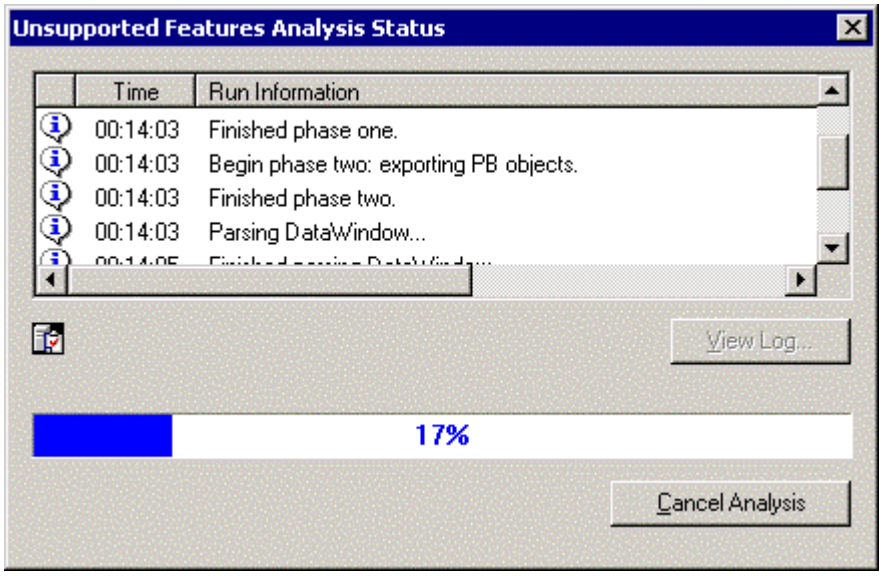
Step 3 – Check the PBLs and/or objects in the treeview for which you want to generate the UFA Report and Heavy Window Report.

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

Step 4 – Click the *Analyze* button.

The feature analysis of the selected objects begins, as shown in Figure 5-4.

Figure 5-4: Unsupported Features Analysis Status



5.2.3 UFA Report Window that displays at the end of feature analysis

When the analysis is completed, the Unsupported Feature Analysis Report window (UFA Report Window) will be displayed. You can view the generated UFA Report and access the Heavy Window Report in the UFA Report Window.

Figure 5-5: UFA Report Window

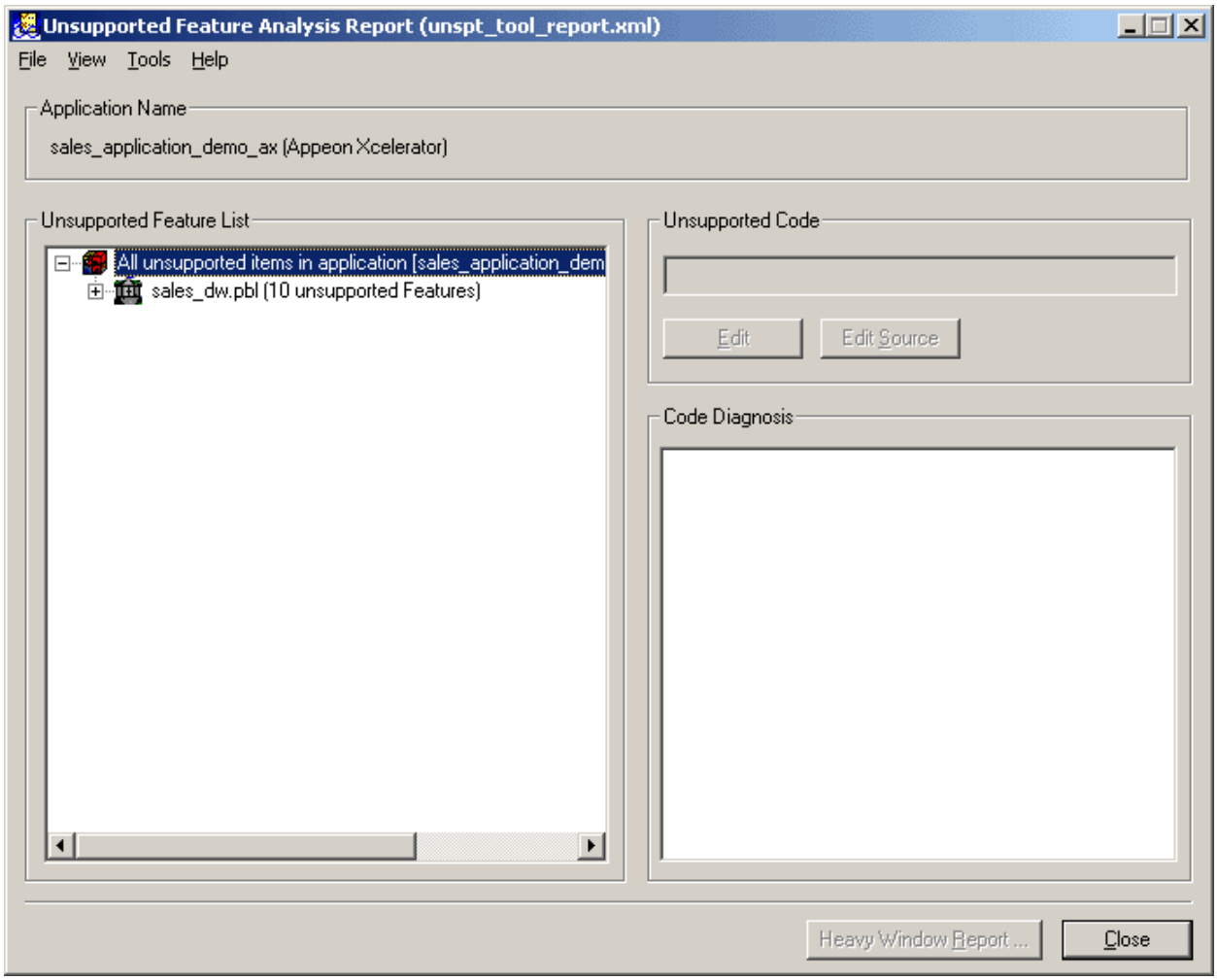


Table 5-4 gives a brief description of the elements in the UFA Report Window. You can find detailed descriptions and instructions in later sections.

Table 5-4: UFA Report Window

Element	Description
File menu	Provides menu items for opening or saving a UFA Report. Refer to Section 5.3.2.a for more information.
View menu	Provides menu items for you to select a display mode of the objects in the report. Refer to Section 5.3.2.b for more information.
Tools menu	Sets the report item filtering according to objects, unsupported feature types or priorities. Refer to Section 5.3.2.c for more information. Allows you to define the priority settings of UFA Report items. Refer to Section 5.3.2.d for more information. Allows you to customize the general settings of the UFA Report. Refer to Section 5.3.2.e for more information.
Help menu	Provides the version number of the UFA tool.

Unsupported Feature List treeview	Displays the unsupported features of selected PBLs and objects in the default view mode.
Unsupported Code textbox	Displays the source code that uses the selected unsupported features.
Edit button	Accesses the PowerBuilder Script view of the corresponding object in the object painter and displays the unsupported code for you to modify. Refer to Section 5.3.1 for more information.
Edit Source button	Accesses the PowerBuilder Script view and displays the unsupported code in the Source editor for you to modify. Refer to Section 5.3.1 for more information.
Code Diagnosis textbox	Diagnoses the unsupported code in five aspects: the Location, Unsupported Item, Type, Reason, and Priority.
Heavy Window Report button	Accesses the Heavy Window Report of the application or selected objects. Refer to Section 5.4 for more information.
Close button	Closes the UFA Report Window and returns to the Select Objects to Analyze window.

5.3 Working with UFA Report

The UFA Report is loaded into the UFA Report Window immediately after the feature analysis is completed. This section describes:

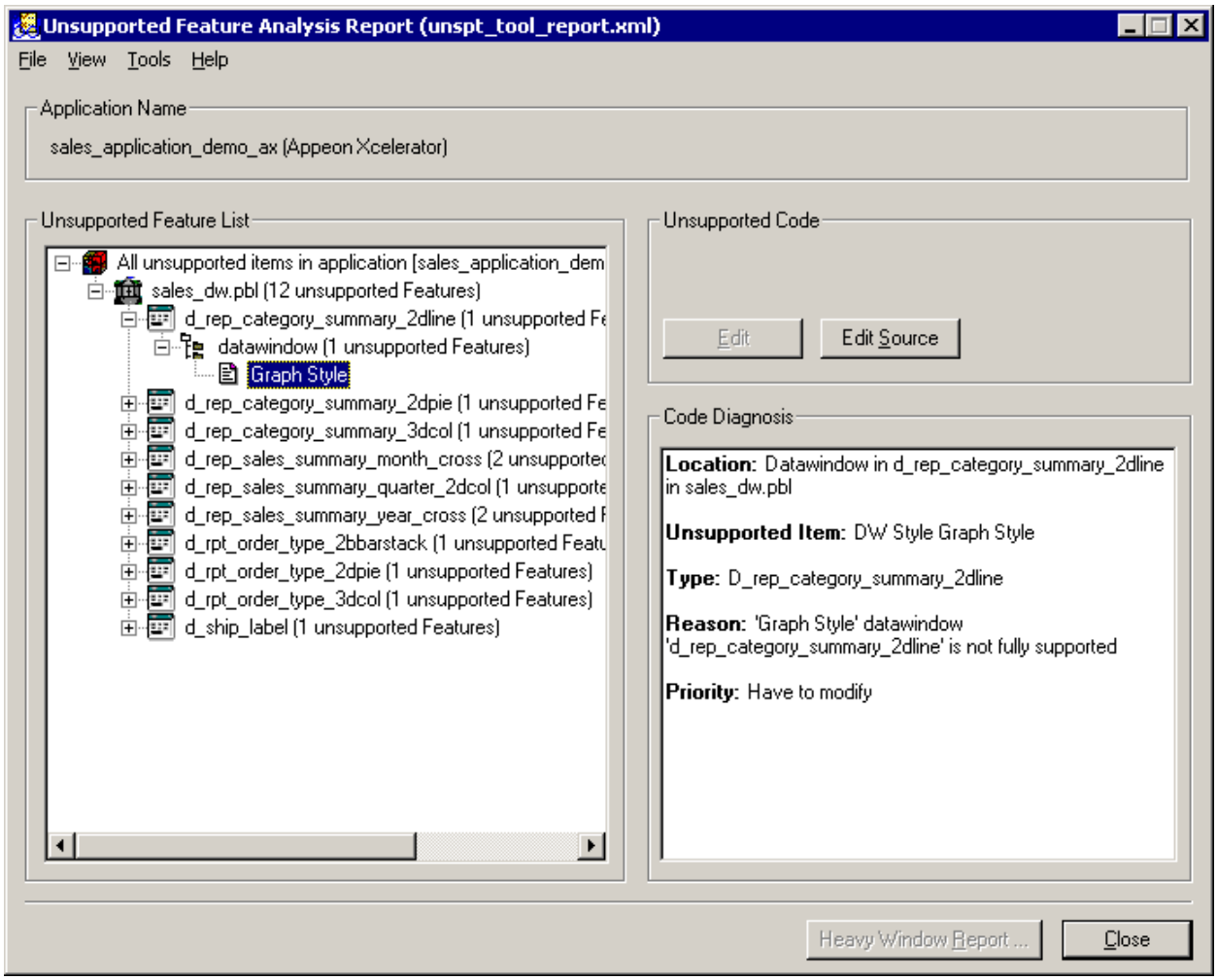
- How to modify unsupported features. You can use the UFA Report to effectively remove unsupported features in the application.
- How to manipulate the UFA Report in the UFA Report Window. You can adjust the view of the UFA Report so it suits your preferences.
- Limitations of UFA Report. There are two major limitations of the UFA Report: (1) There are a small number of unsupported features that the UFA Report cannot detect; (2) For ACF applications (PFC applications that apply Appeon-compliant PFC framework), the UFA Report catches too many unsupported features that should not be caught.

5.3.1 Modifying unsupported features

You can view all the unsupported features in the Unsupported Feature List treeview and modify them one by one.

Step 1 – Expand the Unsupported Feature List treeview and click the method under each node. The detailed analysis of the selected method will be displayed on the right of the window, as shown in Figure 5-6.

Figure 5-6: Unsupported feature details

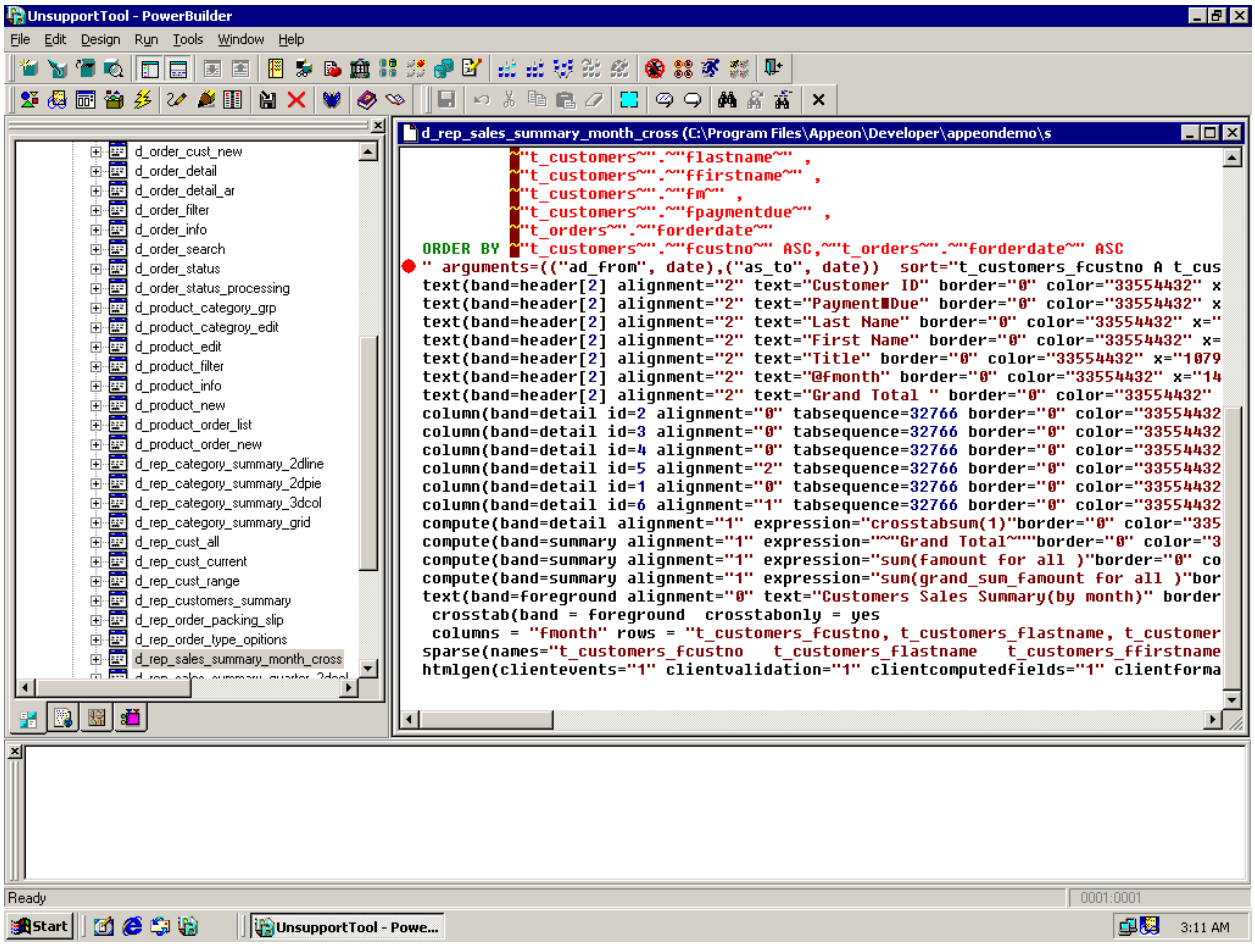


The *Unsupported Code* group box shows the code that uses the selected unsupported feature. The *Code Diagnosis* group box provides a diagnosis on the unsupported code in five aspects: the Location, Unsupported Item, Type, Reason, and Priority.

Step 2 – Click *Edit* or *Edit Source* to modify the source code in the PowerBuilder Script view.

The PowerBuilder Script view automatically opens and scrolls to display the unsupported code. A red bullet is placed at the left side of the unsupported code for easy identification, as shown in Figure 5-7.

Figure 5-7: Unsupported code in the PowerBuilder painter



An icon is displayed on the status bar to help you quickly modify the unsupported features. Click the icon to access the popup menus, as shown in Figure 5-8.

Figure 5-8: Popup menu items

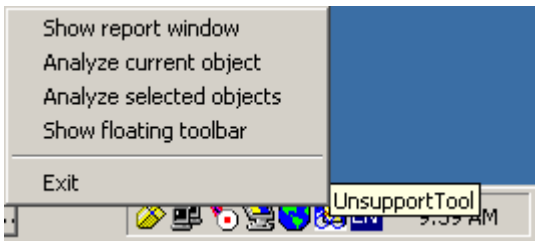


Table 5-5: Icon description

Menu Item	Description
Show/Hide report window	Displays or hides the UFA Report Window.
Analyze current object	Analyzes the current object and generates an UFA Report for the current object.
Analyze selected objects	Analyzes the selected objects again and refreshes the UFA Report for all the selected objects.

Show floating toolbar	Displays a floating toolbar consisting of the shortcut icons of menu items: Show/Hide report window, Analyze current object and Exit.
Exit	Quits the UFA Report Window and returns to the Select Objects to Analyze window.

Step 3 – Remove or work around the unsupported code in the PowerBuilder painter.

You can refer to the *Appeon Workarounds Guide* on how to work around common unsupported features.

Step 4 – Click *Save* on the PowerBuilder IDE to save the modified code.

Step 5 – Click *Analyze* on the floating toolbar at the top of the screen or select *Analyze current object* from the popup menu to analyze the current object and to generate an UFA Report for it.

You can also select *Analyze selected objects* from the popup menu to analyze all the selected objects again and refresh the UFA Report based on the modified source code.

Step 6 – Click *Show* on the floating toolbar at the top of the screen or select *Show report window* from the popup menu to display the UFA Report Window, and to continue to modify the other unsupported code in the Unsupported Feature List treeview.

Repeat Step 1 to Step 6 until you remove all the unsupported features for the application.

5.3.2 Manipulating the UFA Report in the UFA Report Window

5.3.2.a Open/Save a UFA Report

You can open or save a UFA Report using the File menu in the UFA Report Window. The File menu enables you:

Table 5-6: Using File menu

To Do This	Do This
To open an existing UFA Report	Select the File | Open menu.
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.
To export the UFA Report to an HTML file	Select the File | Export As HTML menu.
To exit the UFA Report Window	Select the File | Exit menu.

5.3.2.b Select report view mode

You can view the UFA Report in three 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 | Priority menu to define the priorities of the PowerBuilder features): Have to modify, Suggest to modify, Can be ignored.

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

5.3.2.c Filter UFA Report items

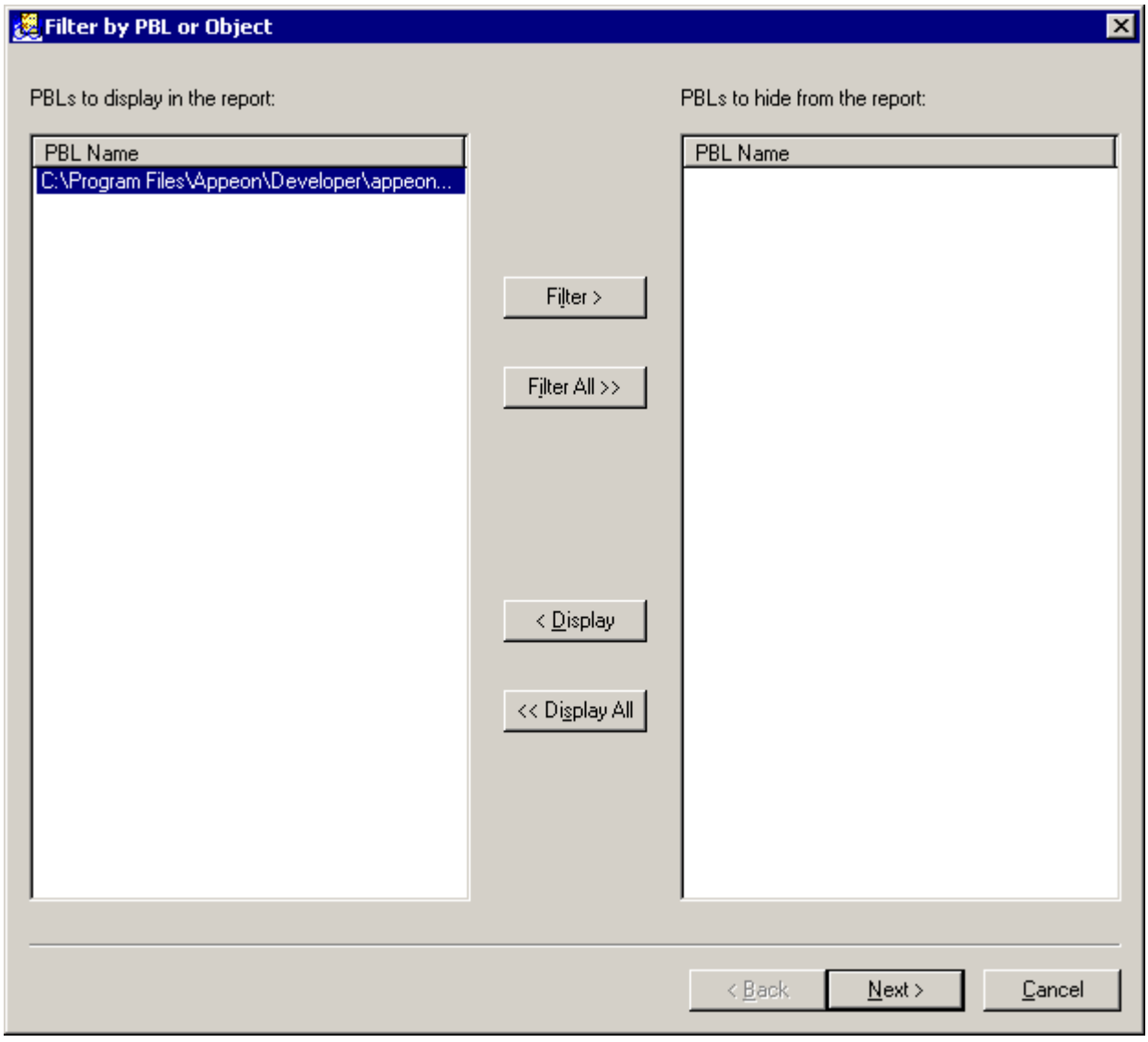
Filtering helps you restrict the number of the unsupported features displayed in the UFA Report and quickly find the target unsupported features, especially when there are a large number of unsupported features in the application.

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.

Filter by objects

Step 1 – Select the Tools | Filter | Object menu in the UFA Report Window. The Filter by PBL or Object dialog box is displayed, as shown in Figure 5-9.

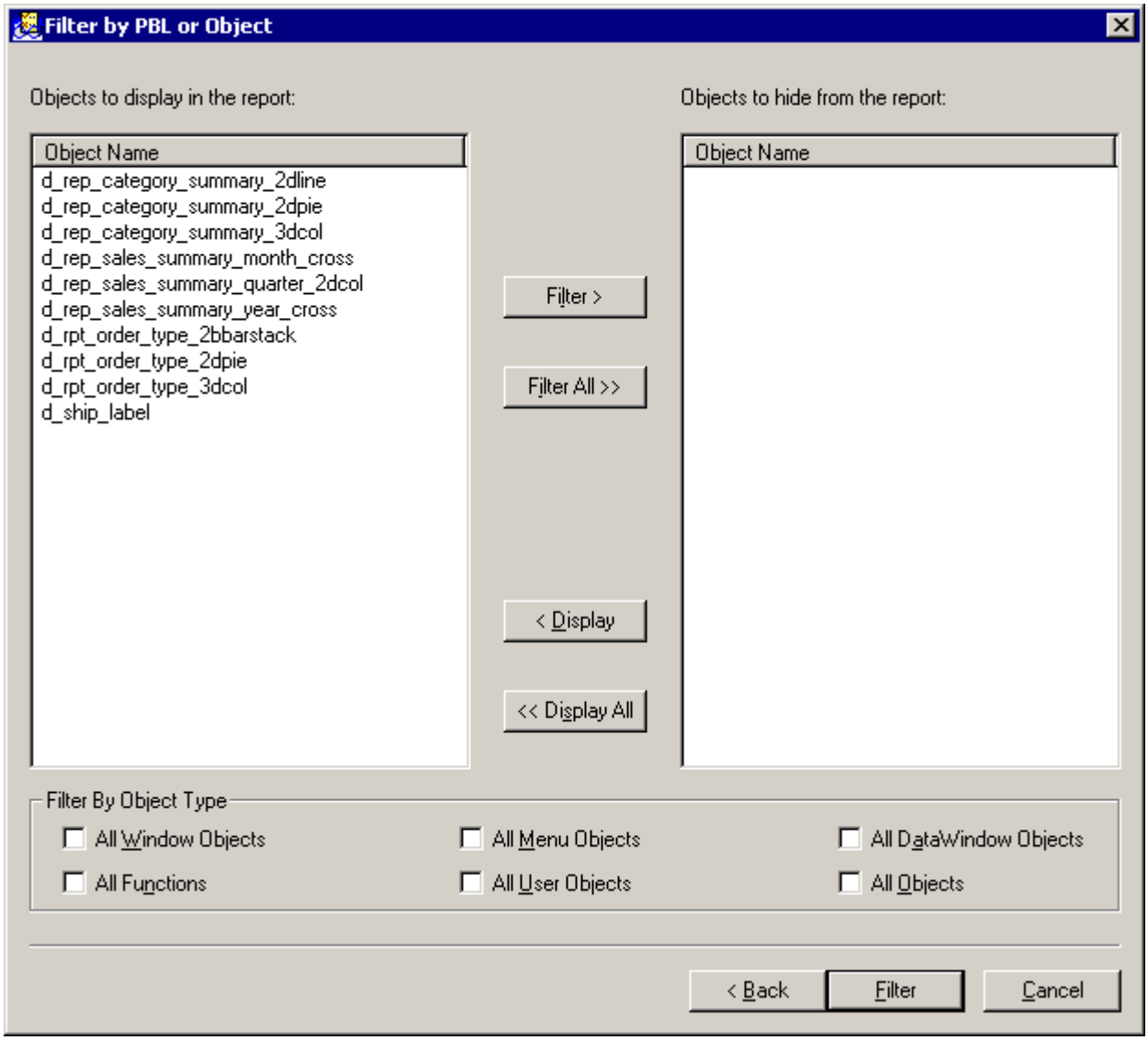
Figure 5-9: Filter PBLs



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

All the unsupported objects contained in the selected PBLs are listed in the left listbox of the window, as shown in Figure 5-10.

Figure 5-10: Filter Objects



Step 3 – Make sure the unsupported objects you want to display in the report are on the left listbox, whereas the unsupported objects you want to hide are on the right listbox.

Step 4 – (Optional) In the *Filter by Object Type* group box, you can select one or more object types to make all objects of the selected types on the left listbox automatically selected, and then click *Filter* to add them to the right listbox.

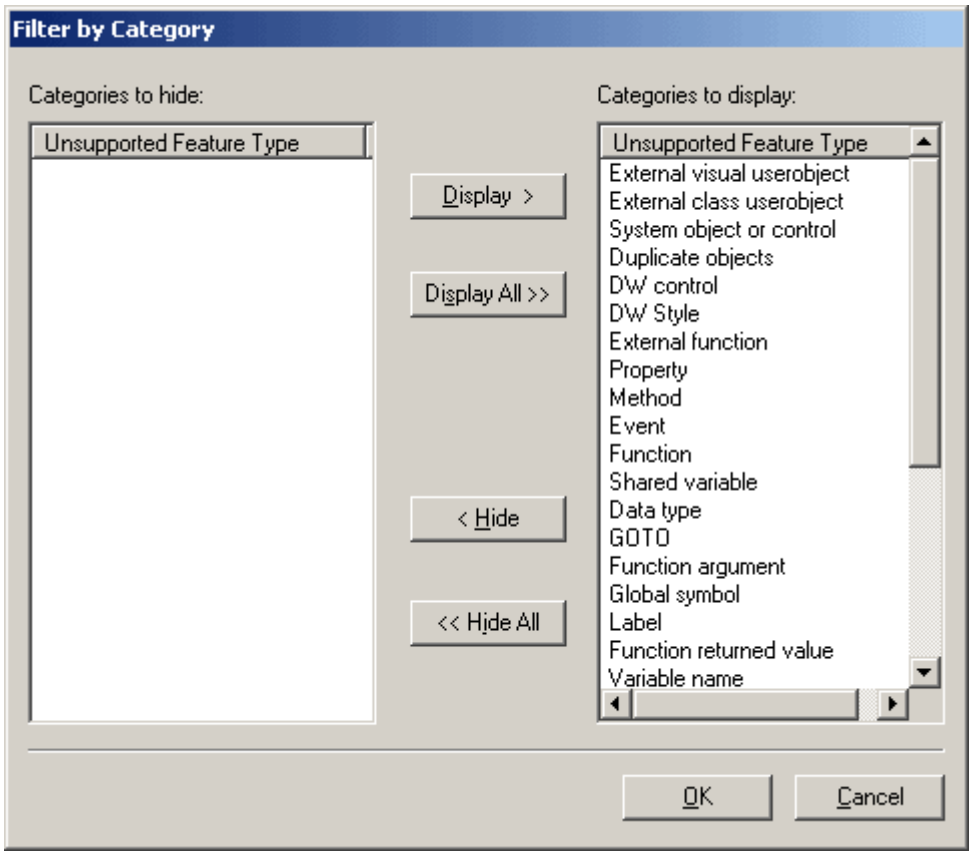
Step 5 – Click *Filter*.

The UFA Report hides the unsupported objects that are on the right listbox in the Filter by PBL or Object window.

Filter by categories

Step 1 – Select the Tools | Filter | Category menu in the UFA Report Window. The Filter by Category dialog box is displayed, as shown in Figure 5-11.

Figure 5-11: Filter by Category



Step 2 – Move the categories of the unsupported features that you want to hide from the right box to the left box.

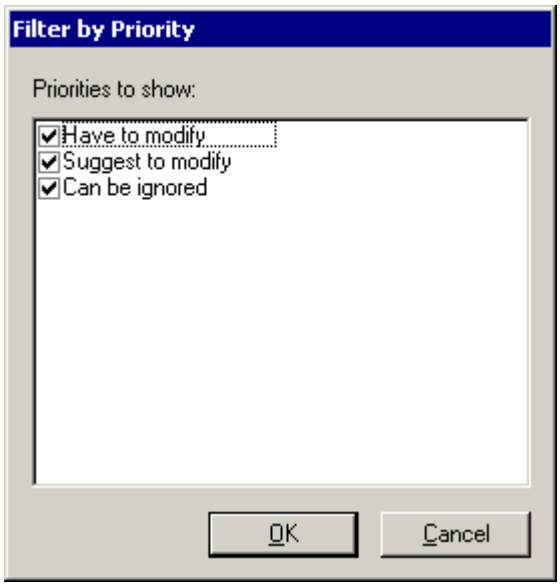
Step 3 – Click *OK*.

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

Filter by priorities

Step 1 – Select the Tools | Filter | Priority menu in the UFA Report Window. The Filter by Priorities dialog box is displayed, as shown in Figure 5-12.

Figure 5-12: Filter by Priorities



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.d Configure the priority settings of unsupported features

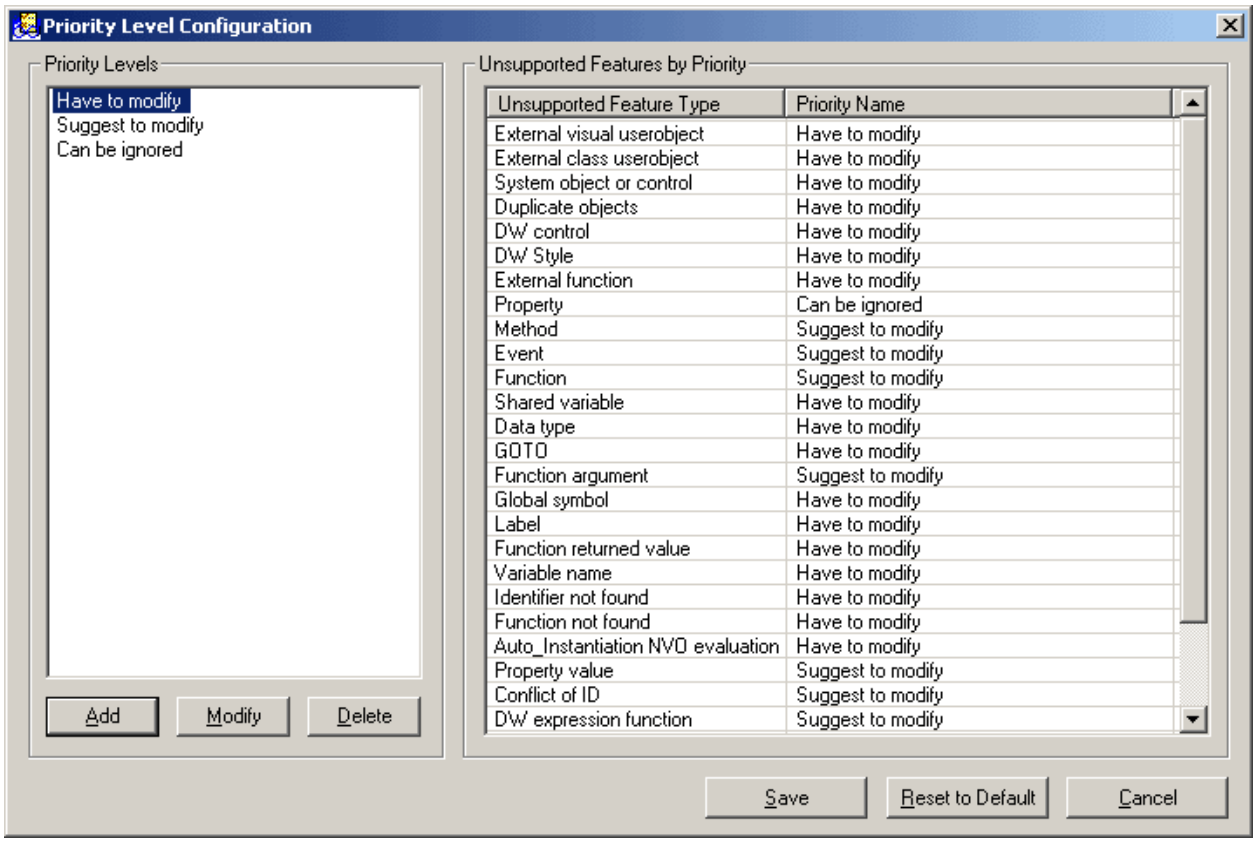
By default, Appeon divides all unsupported features into three priority levels:

- 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 | Priority menu in the UFA Report Window. The Priority Level Configuration window is displayed, as shown in Figure 5-13.

Figure 5-13: Priority Level Configuration dialog box



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

Step 3 – Change priority levels of unsupported feature types by modifying the priority in the Priority Name column.

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

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

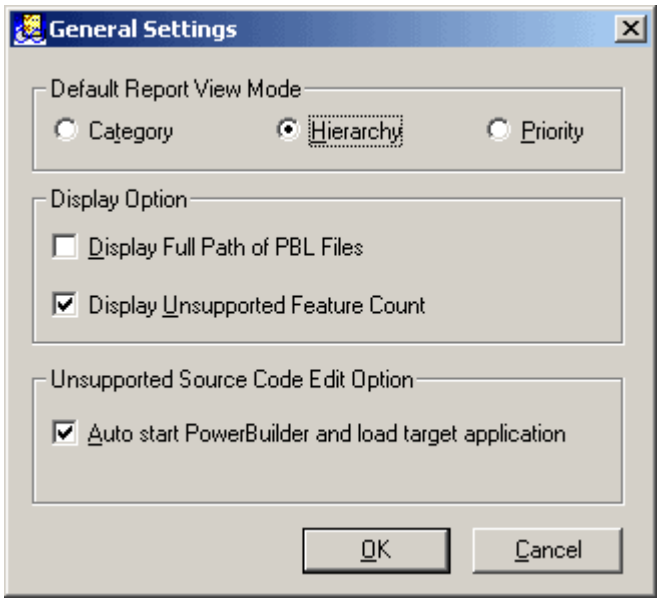
5.3.2.e Customize the general settings of the UFA Report

You can define your preference settings of the UFA Report Window. The new settings will be saved as the default settings.

The Tools | General menu provides settings for the default report view mode, display formats of PBLs, and feature summaries, and the unsupported source code edition.

Step 1 – Select the Tools | General menu in the UFA Report Window. The General Settings window is displayed, as shown in Figure 5-14.

Figure 5-14: General Settings dialog box



Step 2 – Set the default view mode of the UFA Report in the *Default Report View Mode* group box.

- Category – View the unsupported features in the following categories: Unsupported Objects, Unsupported PowerScript, Unsupported Embedded SQL and Others.
- Hierarchy – View the unsupported features in the following hierarchical order: PBL, Object, Control, Method (Event/Function).
- Priority – View the unsupported features by specific priority: Have to modify, Suggest to modify, Can be ignored.

Step 3 – Select whether to display the full path of PBLs and the total number of the unsupported features.

- 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.

Step 4 – Specify the unsupported source code edition options.

Auto start PowerBuilder and load target application – If this option is checked, PowerBuilder automatically starts and loads the target application when you click the *Edit* or *Edit Source* button in the UFA Report Window.

5.3.3 Limitations of UFA Reports

5.3.3.a 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 Unsupported Features Analysis 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 for Appeon Xcelerator* or *for Pure-JavaScript*, which documents the types of unsupported features that may not be detected.

5.3.3.b UFA Report for ACF applications

If you perform an Unsupported Features Analysis (UFA) on Appeon-compliant Framework (ACF) PBLs, the UFA will report a large number of unsupported features because Appeon ACF supports 85% of the PowerBuilder PFC features and the UFA report has caught the 15% of the PFC features that are unsupported. Although "15%" seems to be a pretty large number, this 15% actually only covers the features that are most rarely used in PFC and the absence of these features usually does not affect the overall functionality of the application.

This section gives you guidance on how to read useful information from UFA Reports for ACF applications and how to preserve the “15%” of PFC features for PowerBuilder applications although you will lost them on the Web.

Reading useful information from UFA Reports for ACF applications

Although the UFA report is long, we can ignore the features reported for ACF framework and only focus on the features that are used in the application source code. The following are two ways to get the useful information from the report.

1. Ignore the features that are reported for Appeon ACF libraries. You can immediately skip a feature in the report if it belongs to a "pfc..." library, because the names of Appeon ACF libraries start with "pfc...". For example, the following features in the UFA Report can be ignored:

```
c:\appeon\pfcmain\pfcmain.pbl\pfc n tr\of distinctvalues(string,st
ring,string appeon array appeon )  describe describe sqlsa into
sqlda

c:\appeon\pfcmain\pfcmain.pbl\pfc n tr\of distinctvalues(string,st
ring,string appeon array appeon ) fetch fetch c values cursor
using descriptor sqlda
```

2. Pay special attention to the features that do not belong to a "pfc…" library. For example, if your application contains a library named "testappeon.pbl", make sure you study all the features in the report that are related to "testappeon.pbl".

Preserving the “15%” of PFC features for PowerBuilder applications

The “15%” of PFC features that ACF does not support will definitely not work in the deployed applications, but you can preserve them in PowerBuilder applications. The methodology is to use a function to check the runtime environment of the application, PowerBuilder or Web, and then use an IF THEN statement to execute different codes for different environments. Appeon provides two functions, of_iscs and AppeonGetClientType, for you to implement the methodology.

Table 5-7 shows you how of_iscs or AppeonGetClientType are defined in Appeon, and provides two samples on using the functions to execute different code for PowerBuilder or Web application.

Table 5-7: Executing different code for PowerBuilder or Web application

Function	Description of the Function	Sample

of_iscs()	Defined in pfc_n_cst_environment of pfc_appeon.pbl
	Returns TRUE if it is executed in a PowerBuilder application, and returns FALSE if it is executed in a Web application.
		IF of_iscs() THEN
		Unsupported ACF Code
		END IF

AppeonGetClientType	• Provided in the %Developer%\appeon_workarounds\JS\appeon_workarounds_js.pbl and the %Developer%\appeon_workarounds\AX\appeon_workarounds_ax.pbl
	• Returns “PB” if it is executed in a PowerBuilder application, and returns “WEB” if it is executed in a Web application.
		IF appeongetclienttype() = "PB" THEN
		Unsupported ACF Code
		ELSEIF appeongetclienttype() = "WEB" THEN
		ACF Code
		END IF

5.4 Working with Heavy Window Report

You can access the Heavy Window Report from the UFA Report Window, or from the Information Manager on the following conditions:

1. The *Generate Heavy Window Report during Application Analysis or Deployment* option is enabled in the Application Profile Configuration window
2. The PBLs or objects that are analyzed contain one or more heavy windows

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.

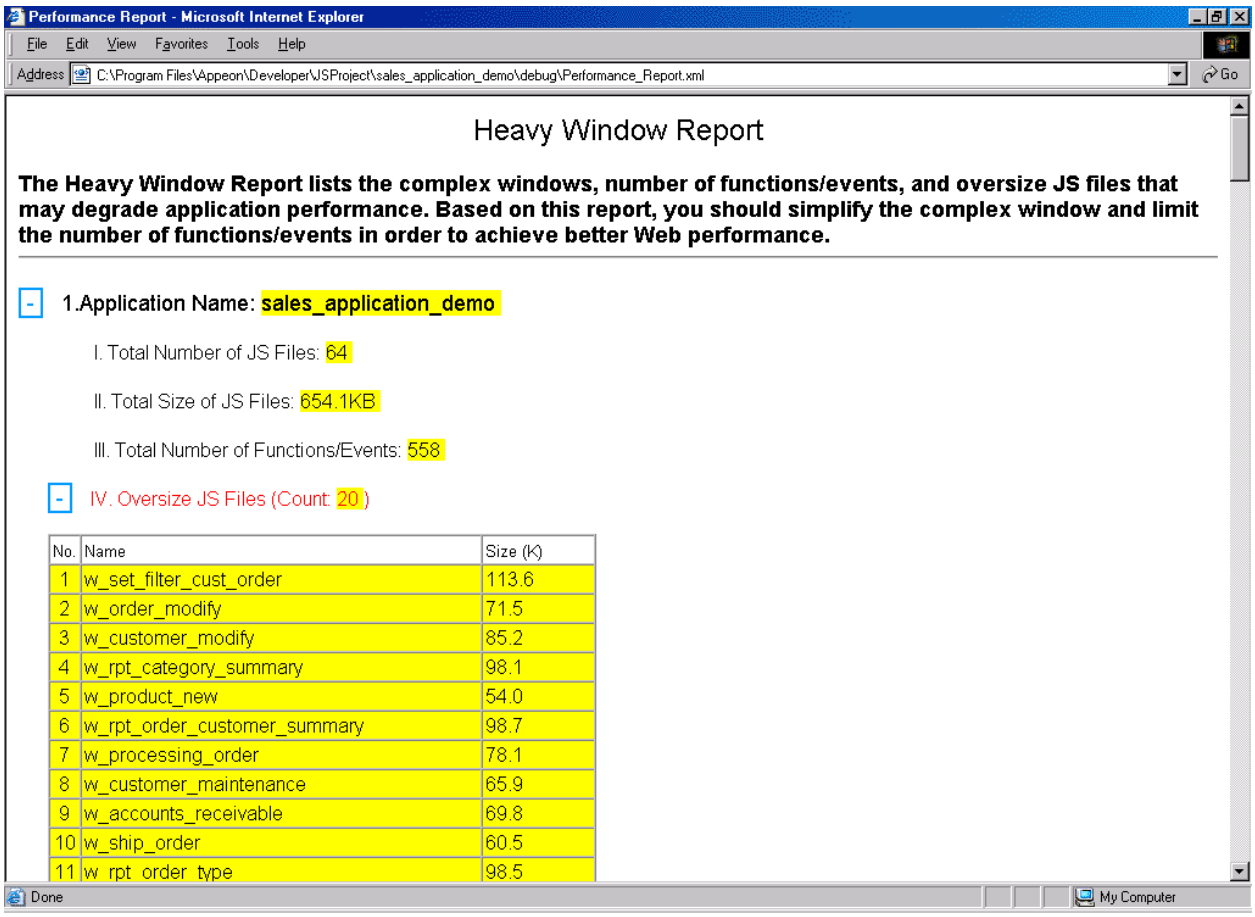
The weight of the different control types, as well as the default minimum load value, are all stored in the PerformanceStatConfig.xml file in the %Appeon%/Developer path. You can change the settings in the XML file based on actual requirements. (Example: change the minimum load value to 6.0 to get a 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 5-15. This helps you measure each window's performance.

Figure 5-15: Heavy Window Report



5.4.3 Making heavy windows lighter

For instructions on how to make heavy windows lighter, please 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 for Appeon Xcelerator* or *for Pure-JavaScript*.

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 DataWindows to Appeon Server.
5. Transfers local Web files to the Web Server by either file copy or FTP.

Note: 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 an EAServer Web Server as an SSL Web Server, refer to Section *Tutorial: Using SSL* in the *EAServer Security Administration and Programming Guide*.

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 Section 3.1: *System Requirements* 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 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 6-1: Deployment duration

Deployment Task	Elapsed Time
Task 1: Application Source Code Export	8 seconds
Task 2: Web File Generation	27 seconds
Task 3: Web Deployment	15 seconds (for a local deployment)
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.

6.2.2 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 6-2 shows you which type of build options you should select for the application.

Table 6-2: Build options

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

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 () in the Appeon Developer toolbar. The Appeon Application Deployment Wizard is displayed.

Figure 6-1: Appeon Application Deployment Wizard

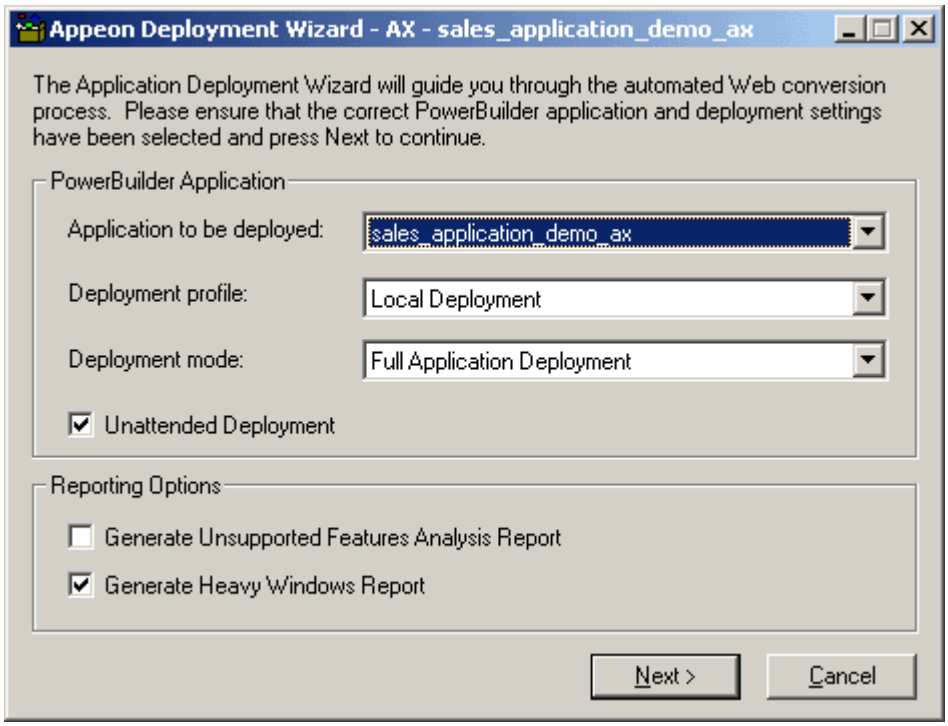


Table 6-3 describes how to specify the deployment settings for an application:

Table 6-3: Appeon Deployment Wizard settings

Setting	Description
Application to be deployed list box	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. The deployment wizard title displays the name of the selected application and the deployment option (Pure-JavaScript or Appeon Xcelerator). The Pure-JavaScript and Appeon Xcelerator deployment processes are identical.
Deployment Profile list box	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.
Deployment Mode list box	Select one of the three available deployment modes: • Full Application Deployment • Incremental Application Deployment • Deploy Already Generated Application
Unattended Deployment check box	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.

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. Checking or unchecking this option will automatically check or uncheck the *Generate Heavy Window Report during Application Analysis or Deployment* option on the *Misc Settings* tab of the application profile, and vice versa.

6.3.2.a Selecting the deployment mode

Table 6-4 describes the three different deployment modes and guides you in choosing the proper deployment mode.

Table 6-4: Deployment mode

Deployment Mode	What It Does	Elapsed Time	When To Use It
Full deployment	Exports all PowerBuilder application objects and code, generates the corresponding Web files for the application, and deploys all Web files to the server.	Depends on the size and complexity of the PowerBuilder application.	The first time an application is deployed to the Web.
Incremental deployment	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.	Up to 80% less time than a full deployment.	After making changes to an already fully-deployed application's source code or features in the application profile.
Deploy Already Generated Application	Bypasses the object exporting and file generation tasks and only re-deploys an application whose Web files have already been generated.	Much less time than a Full or Incremental Deployment.	When you intend to: 1) deploy the application to a different server. 2) refresh the set of Web files on the server after a Full or Incremental Deployment has been cancelled during Web deployment (Task 3).

Appeon Developer will automatically set the default deployment mode by detecting whether a folder named after the application exists in the %Appeon%\Developer\AX(JS)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 6-5: Situations when the folder exits

The folder will exist if you have …	In this case, choose …
Performed an unsupported feature analysis on the application before it has undergone a full deployment. The folder is created during analysis.	Incremental deployment mode.
Undeployed an Appeon deployed application from servers using the Application Undeployment Wizard. The folder is preserved.	Full deployment mode.
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.	Full deployment mode.

6.3.2.b Enabling runtime report

To use the runtime reports to debug and fine-tune a deployed Web application, before you perform the application deployment, select the *Enable Web Debug Report and Runtime Tracing Report* check box in either of the following two windows:

- In the Performance Tracing Configuration window. To access this window, click the *Trace Configure* button () in the Appeon Developer toolbar.
- In the Web Debugging Configuration window. To access this window, click the *Debug Configure* button () in the Appeon Developer toolbar.

The *Enable Web Debug Report and Runtime Tracing Report* check box on these two windows is selected or unselected simultaneously. For more information of how to configure the runtime report after deployment, refer to Section 8.2: *Generating Runtime Reports*.

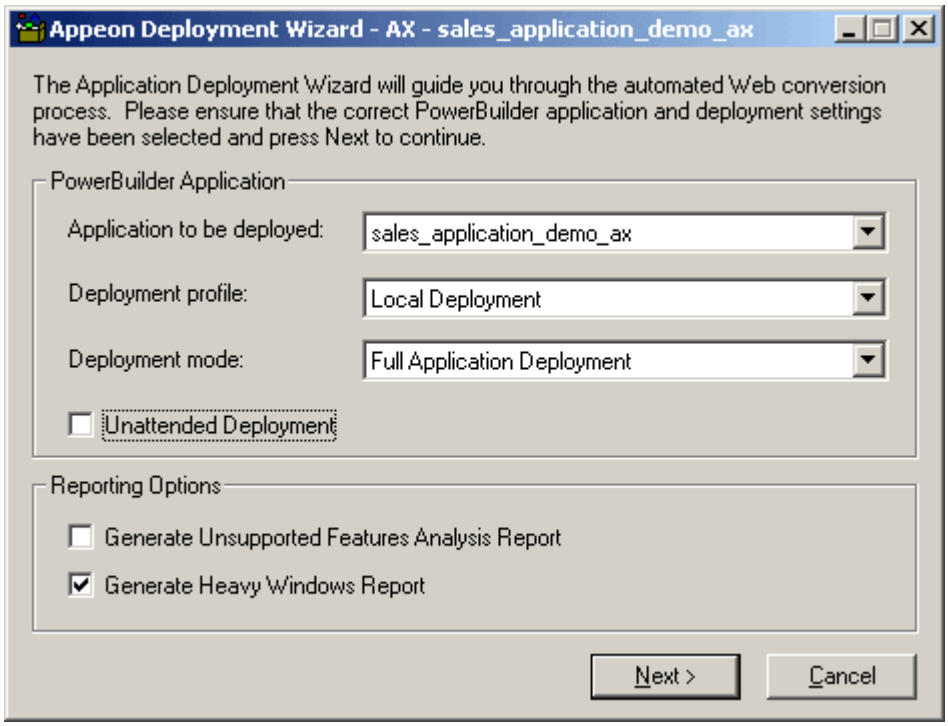
6.3.3 Deploying the PowerBuilder application

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

Table 6-6: Different tasks performed during each deployment mode

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

Figure 6-2: Full application deployment

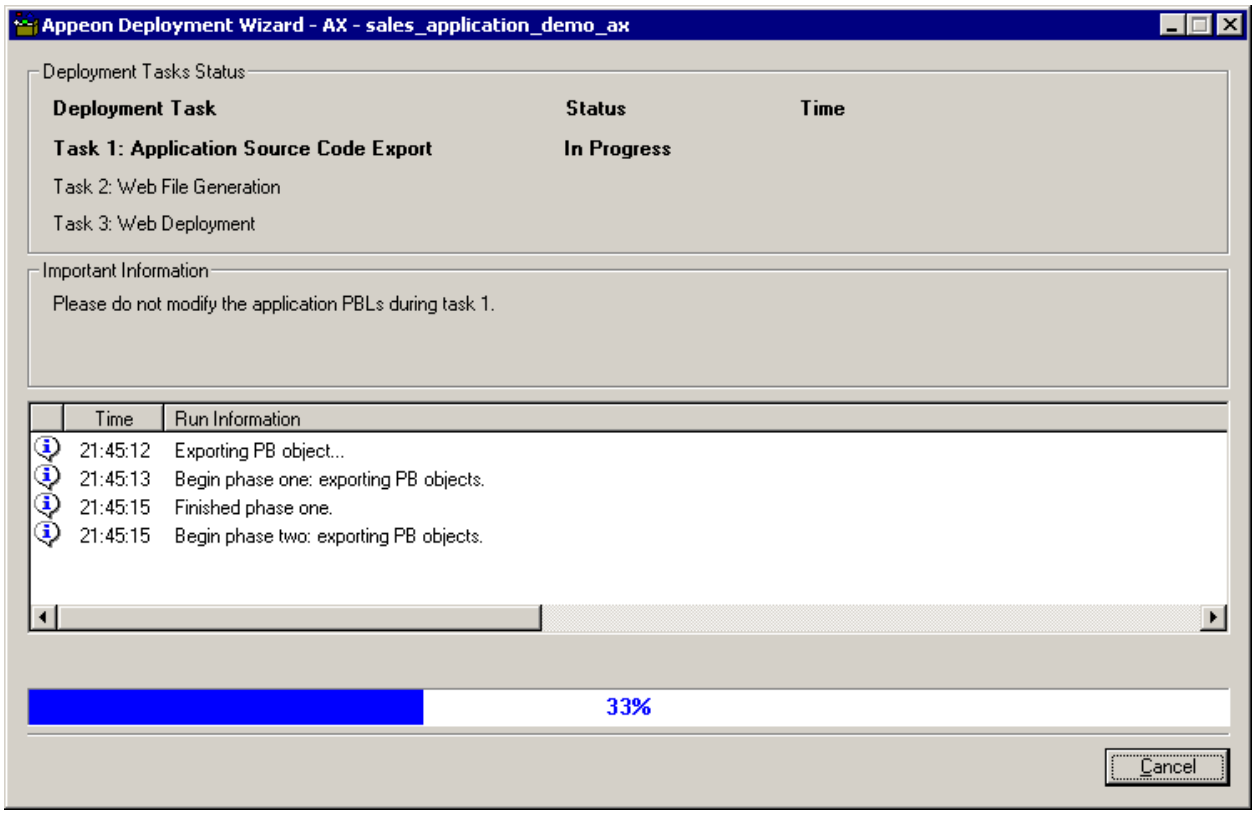


As shown in Figure 6-2, the application to be deployed is the *sales_application_demo_ax*, the deployment option selected is Appeon Xcelerator, and the deployment mode selected is Full Application Deployment.

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

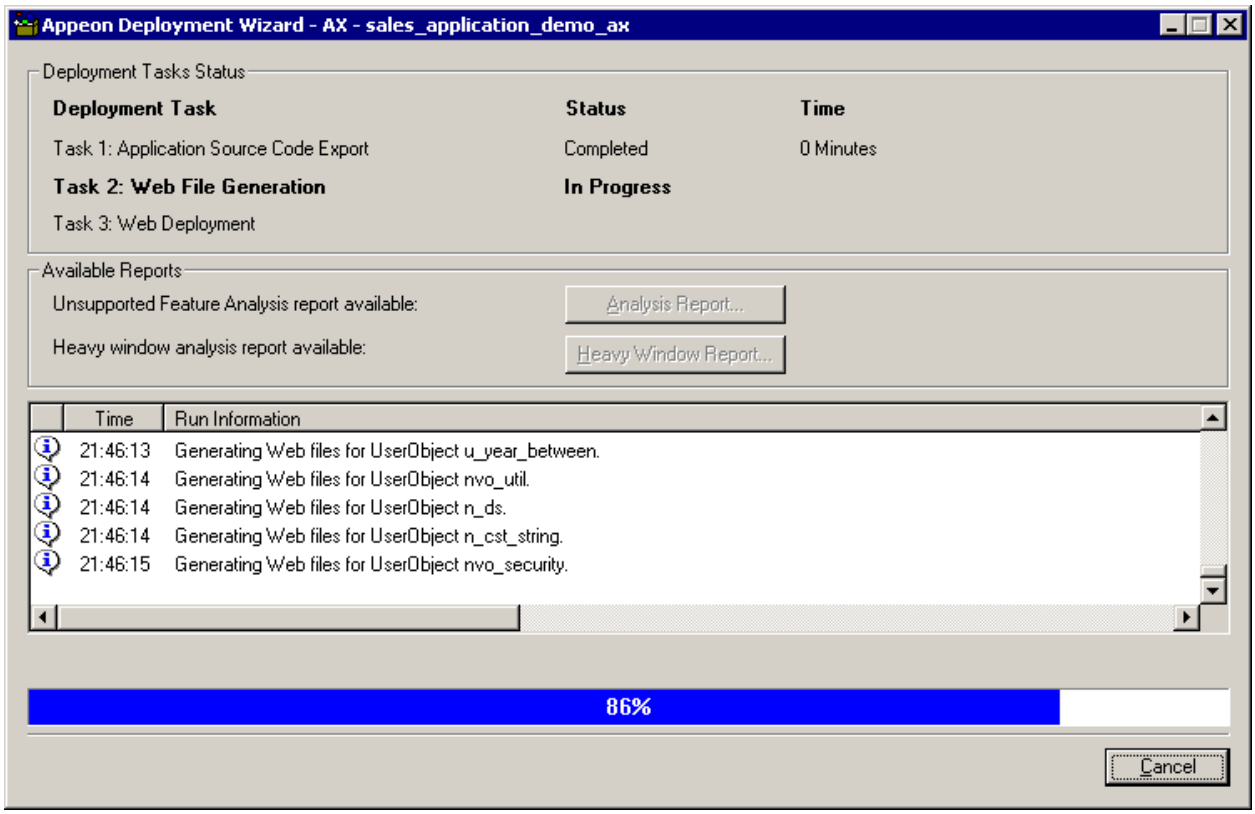


Click *Cancel* or press Esc to terminate the entire deployment process.

The *View Log* button will be available when Task 1 is completed. Click it to view the error messages and status information generated during Task 1. The log file can be managed in the Information Manager. For detailed instructions, refer to Chapter 10: *Using Information Manager*.

Step 2 – After Task 1 is completed, click *Next* at the bottom of the Appeon Deployment Wizard to begin Task 2, Web File Generation, as shown in Figure 6-4.

Figure 6-4: Task 2: Web File Generation



Click *Cancel* or press the Esc key if you want to terminate the entire deployment process.

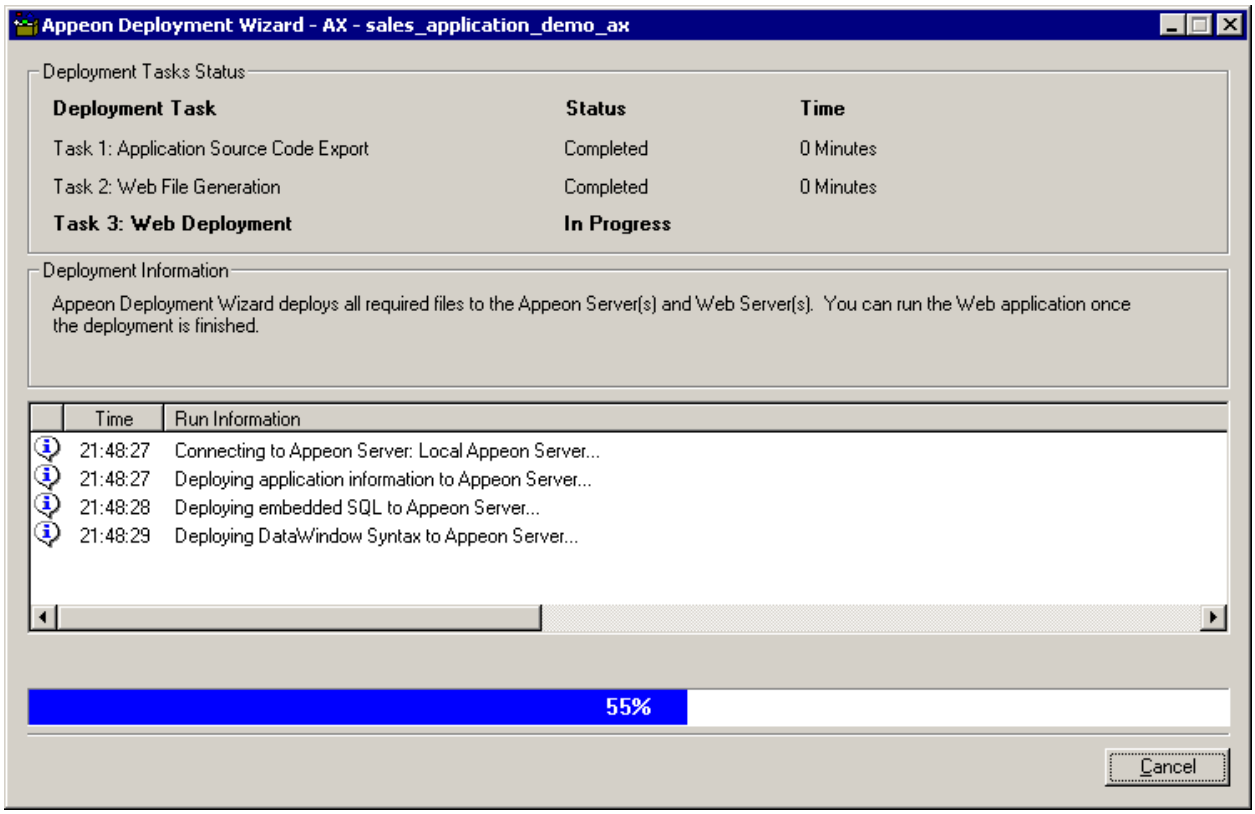
When Task 2 is complete, you can click the following buttons to view the report in XML format in an Internet Explorer window:

- *Analysis Report* button – This button is available if the *Generate Unsupported Feature Analysis Report* check box 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 5.3.2: *Manipulating the UFA Report in the UFA Report Window*.
- *Heavy Window Report* button – This button is available if the *Generate Heavy Window Report* check box on the Appeon Deployment Wizard or the *Generate Heavy Window Report during Application Analysis or Deployment* option on the *Misc Settings* tab of the Application Profile is selected before deployment.
- *View Log* button – This button is available when Task 2 is completed. The log file records error messages and status information generated during Task 1 and 2. The log file can be managed in the Information Manager. For detailed instructions, refer to Chapter 10: *Using Information Manager*.

Step 3 – After Task 2 is completed, click the *Next* button at the bottom of the Appeon Deployment Wizard to begin Task 3, Web Deployment, as shown in Figure 6-5.

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.

Figure 6-5: Task 3 Web Deployment



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 *Full Deployment* or *Deploy Already Generated Application* as the deployment mode the next time you attempt to deploy the application.

The *View Log* button will be available when Task 3 is completed. Click it to view the error messages and status information generated during the entire deployment process. The log file can be managed in the Information Manager. For detailed instructions, refer to Chapter 10: *Using Information Manager*.

Step 4 – 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

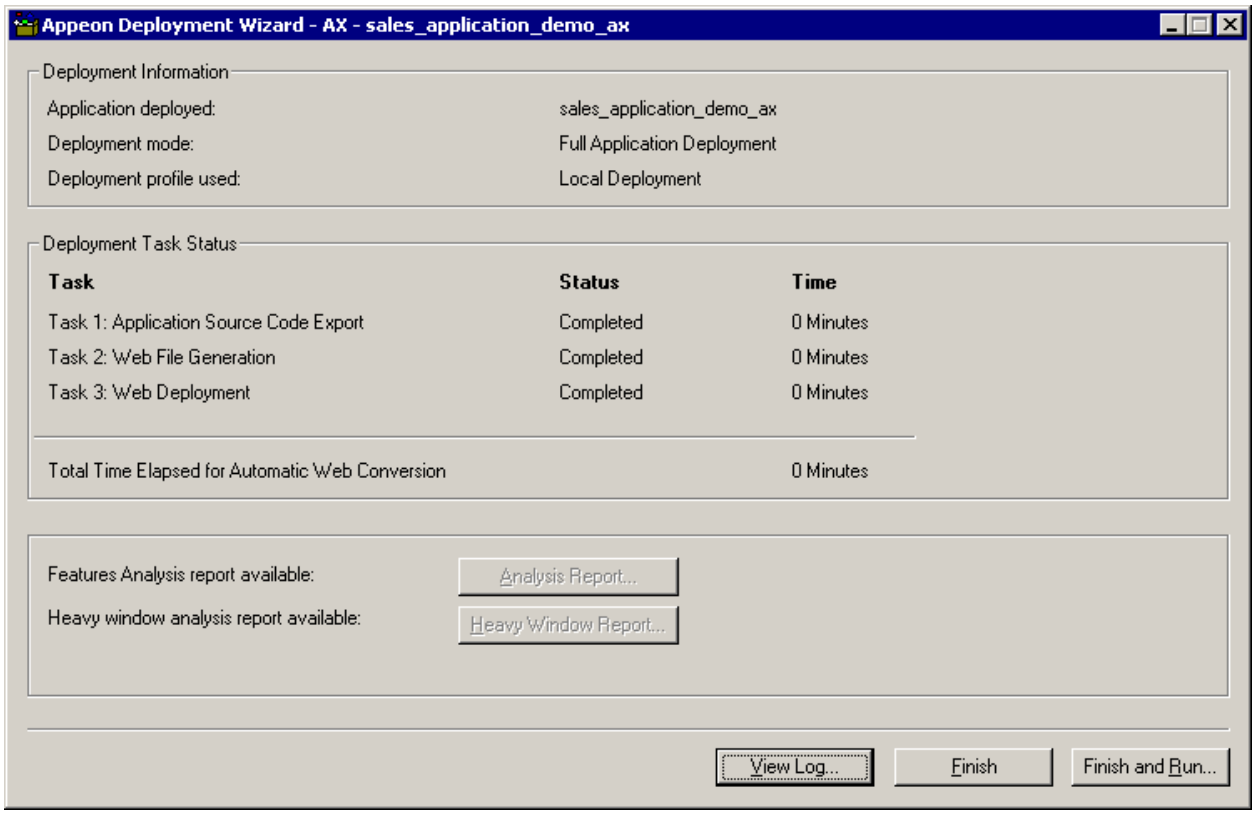


Table 6-7: Deployment report page

Element	Description
Deployment Information	Displays the name of the application deployed, the deployment mode used, and the deployment profile used.
Deployment Task Status	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.
Available Reports	Displays the reports (Analysis Report and Heavy Window Report) generated from each of the three tasks. After you click the *Analysis Report* button, the UFA Report Window will be displayed. For detailed instructions, refer to Section 5.3.2: *Manipulating the UFA Report in the UFA Report Window*.

Step 5 – Click the *Finish* button to close the Appeon Deployment Wizard, or click the *Finish and Run* button to close the Appeon Deployment Wizard and automatically launch the newly deployed Web application.

7 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.

7.1 Requirements

Before launching AEM, verify that:

1. The Appeon Server hosting AEM has been started.
2. You have correctly configured AEM's settings in the Appeon Server Profile Configuration page.

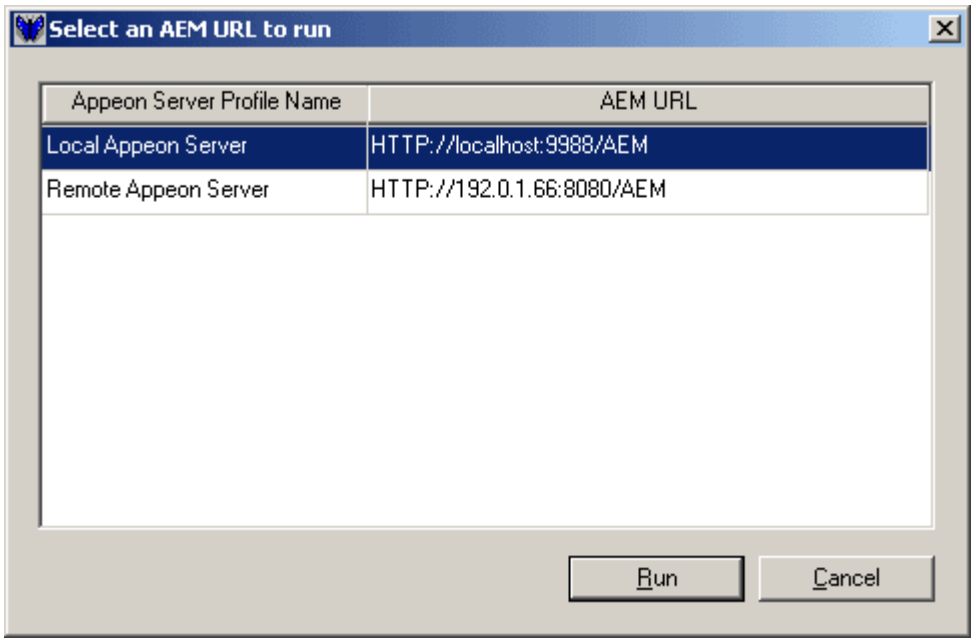
The configuration of AEM is a vital step in the entire PowerBuilder-to-Web conversion process. Before running the deployed Web applications, you must launch AEM and refer to the *Appeon Enterprise Manager User Guide* for instructions on how to configure the transaction object mappings for the Web application.

7.2 Launching Appeon Enterprise Manager

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

If more than one AEM URL is configured to link to Appeon Server profiles, the "Select an AEM URL to run" page is displayed, as shown in Figure 7-1.

Figure 7-1: Select an AEM URL to run

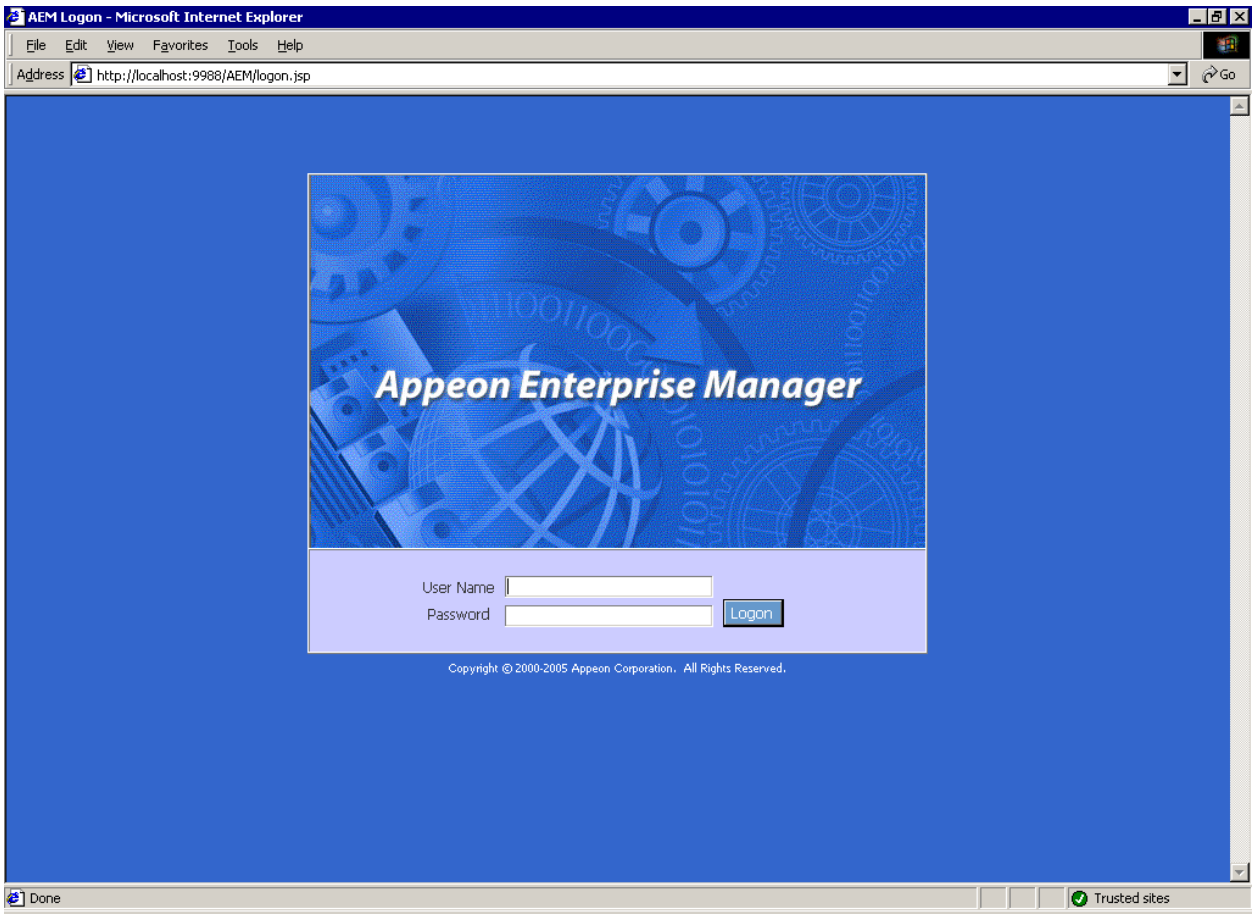


All Appeon Server profiles and corresponding configured AEM URLs are listed. By default, the Appeon Server Profile names are listed in alphabetical order. The Appeon Server configured as the default server is highlighted and selected as the default.

Step 2 – Select an AEM URL and click *Run*, as shown in Figure 7-1.

The entry page of AEM is loaded in an Internet Explorer page, as shown in Figure 7-2.

Figure 7-2: Appeon Enterprise Manager entry page



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 Enterprise Manager User Guide* for more information.

8 Generating and Using Runtime Reports

8.1 Overview

Runtime reports refer to both the Web Debug Report and the Web Performance Runtime Tracing Report that are generated when the application is run. You can configure and generate the runtime reports to help you debug and fine-tune an application during the development and debug stage:

Table 8-1: Web Debug Report and Web Performance Runtime Tracing Report

	Web Debug Report	Web Performance Runtime Tracing Report
What it is for	For debugging the Web application.	For improving the Web application's runtime performance.
What's recorded	Execution information for the Web application: 1) input parameters 2) function return value 3) Appeon comments 4) unsupported code 5) variable/object information.	Runtime performance information of the Web application: • the executed functions/events • the control and root object that contains the functions/events • the time at which functions/events are called • the duration of each call.

8.2 Generating Runtime Reports

The following sections describe how to generate runtime reports for an application, including instructions on how to:

1. Enable runtime report generation before deployment. (See Section 8.2.1)
2. Configure runtime reports after deployment. (See Section 8.2.2 and 8.2.3)
3. Run the application from the Developer machine to generate runtime reports. (See Section 8.2.4)

Note that the runtime report can be generated only when you are running the Web application from the Developer machine where the application is deployed and configured. Both reports are generated and stored on the Developer machine.

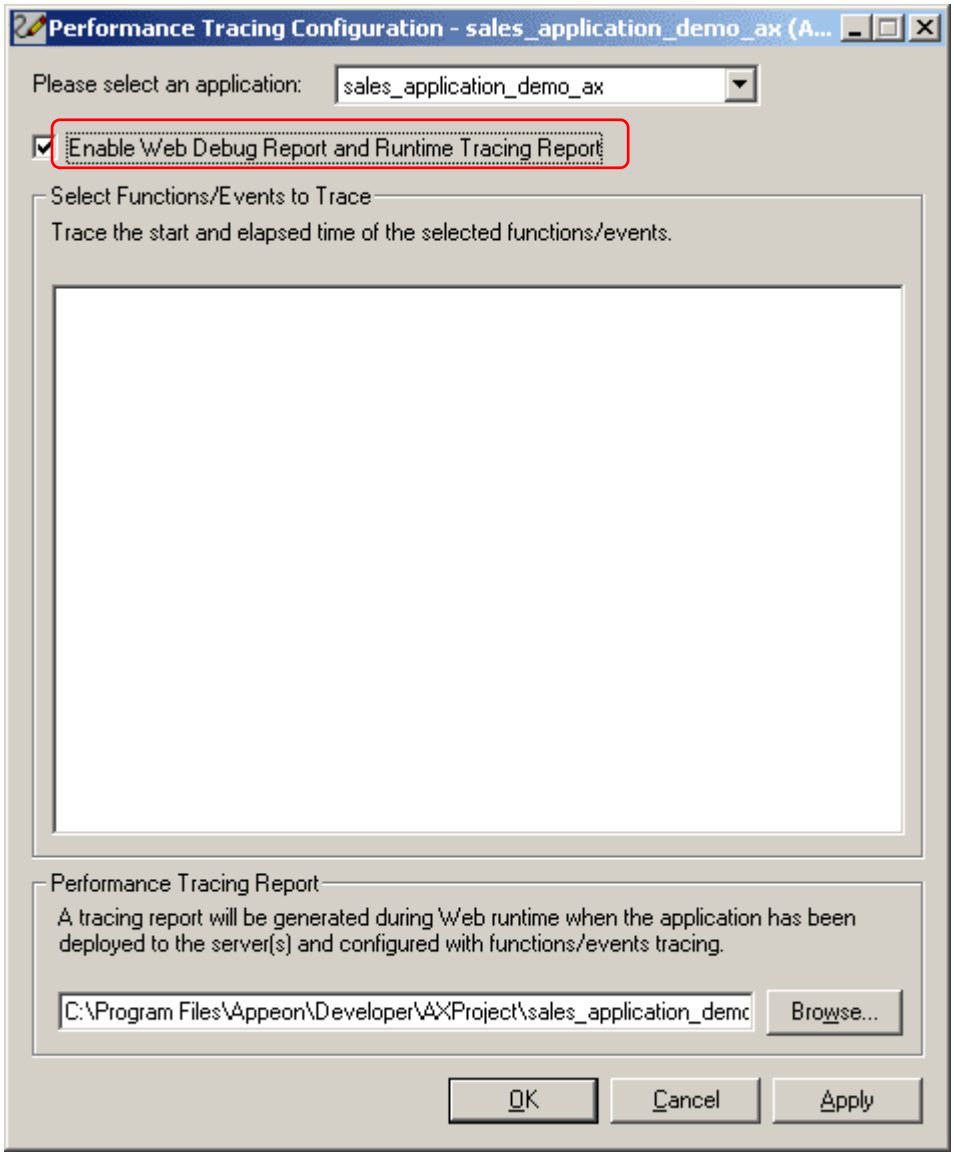
8.2.1 Enabling runtime report

To enable runtime report for an application, select the *Enable Web Debug Report and Runtime Tracing Report* check box in either of the following two windows before the application's full deployment:

- In the Performance Tracing Configuration window as shown in Figure 8-1. To access the Performance Tracing Configuration window, click the *Trace Configure* button () in the Appeon Developer toolbar.
- In the Web Debugging Configuration window. To access the Web Debugging Configuration window, click the *Debug Configure* button () in the Appeon Developer toolbar.

The *Enable Web Debug Report and Runtime Tracing Report* check box on these two windows is selected or unselected simultaneously. Every time you check or uncheck this option for an application, you must perform a full deployment on that application to make the setting effective.

Figure 8-1: Enable Runtime Reporting



The runtime report generation should be enabled only during the development and debug stage. For a fine-tuned application, you should turn off this option and do a full deployment to produce the best runtime performance.

8.2.2 Configuring Web Debug Report

After the application is deployed with the runtime report generation enabled but before the application is run, configure the Web debug report settings in the Web Debugging Configuration window to select the specific functions and events that will be tracked.

To access the Web Debugging Configuration window, click the *Debug Configure* button ( ) in the Appeon Developer toolbar.

Figure 8-2: Web Debugging Configuration window

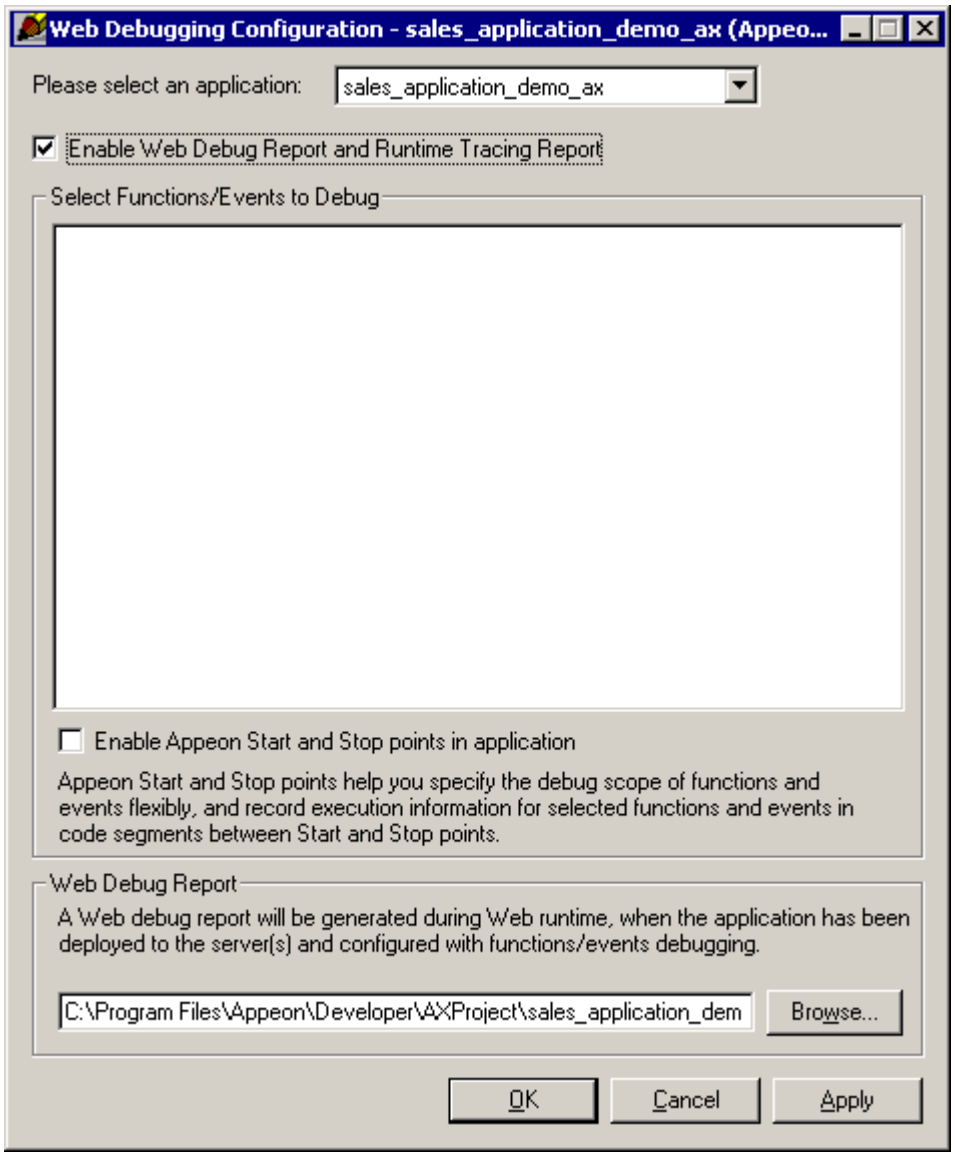


Table 8-2: Web Debugging Configuration window

Option	Description
Application dropdown listbox	Lists the application profiles configured in Application Profiles. The default application profile is selected.
Enable Web Debug Report and Runtime Tracing Report check box	Enables runtime report generation. It must be selected before application deployment and configuration.
Select Functions/Events to Debug treeview	Displays all functions and events defined in the application for you to select which functions/events to track and debug. If the *Enable Web Debug Report and Runtime Tracing Report* check box is not selected before deployment, the treeview will be empty because no function or event list is generated during deployment.
Enable Appeon Start and Stop Points in application	Makes the manually added Appeon Log Start and Stop Points in the application effective.

Browse	Allows you to select a location for storing the Web Debug Report on the Developer machine.

8.2.2.a How to debug with Appeon Log Start and Stop Points

To debug a function/event under specific situations, you can define one or multiple code segments of a selected function/event by manually adding the Appeon Log Start and Stop points to the PowerBuilder source code before deploying the application.

The Appeon Log Start and Stop points in the PowerBuilder source code will be effective when

- You have deployed the application with Appeon Log Start and Stop points;
- You have selected 1) the *Enable Appeon Start and Stop Points in application* check box; and 2) the functions/events contained in the Appeon Log Start and Stop Points in the *Select Functions/Events to Debug* treeview before you run the application.

Use the following Appeon Log Start and Stop Points to specify the code segment. Place the Appeon Log Start Point at the beginning of the code segment and the Appeon Log Stop Point at the end of it.

Appeon log start point

The log start point is marked with "//@@Appeon@@ LOG BEGIN".

Appeon log stop point

The log stop point is marked with "//@@Appeon@@ LOG END".

There must be only one space between //@@Appeon@@ and LOG, between LOG and BEGIN, and between LOG and END. Either upper or lower case text can be used.

Every time you modify, add, or remove the Appeon Log Start and Stop Points in the PowerBuilder source code, you must perform a full or incremental deployment on the application to make the changes effective.

8.2.2.b How to monitor the variables/objects

To monitor specific variables and objects, you can insert a script in the relevant PowerBuilder source code:

Start of the script: "/*@@Appeon@@ OUTPUTVARIABLE BEGIN"

End of the script: "@@Appeon@@ OUTPUTVARIABLE END*/"

There must be only one space between /*@@Appeon@@ and OUTPUTVARIABLE, between OUTPUTVARIABLE and BEGIN, and between OUTPUTVARIABLE and END. Either upper or lower case can be used.

The variables and objects contained in the above script will be tracked, if

- You have deployed the application with the above inserted script;
- You have selected the function/event that contains the monitored variables and objects in the *Select Functions/Events to Debug* treeview before you run the application.

For example, to track the "*ls_variable*" variable, add the following script to your PowerBuilder application's source code:

/*@@Appeon@@ OUTPUTVARIABLE BEGIN

ls_variable

@@Appeon@@ OUTPUTVARIABLE END*/

While running the application, the state of *ls_variable* is recorded into the report the moment the script is executed.

Remember that every time you modify, add or remove the script, you must perform a full or incremental deployment on the application to make the changes effective.

8.2.3 Configuring Web Performance Runtime Tracing Report

After the application is deployed with the runtime report generation enabled, but before the application is run, configure the Web performance runtime tracing report settings on the Performance Tracing Configuration window to select the specific functions and events that will be tracked.

To access the Performance Tracing Configuration window, click the *Trace Configure* button () in the Appeon Developer toolbar.

Figure 8-3: Performance tracing configuration window

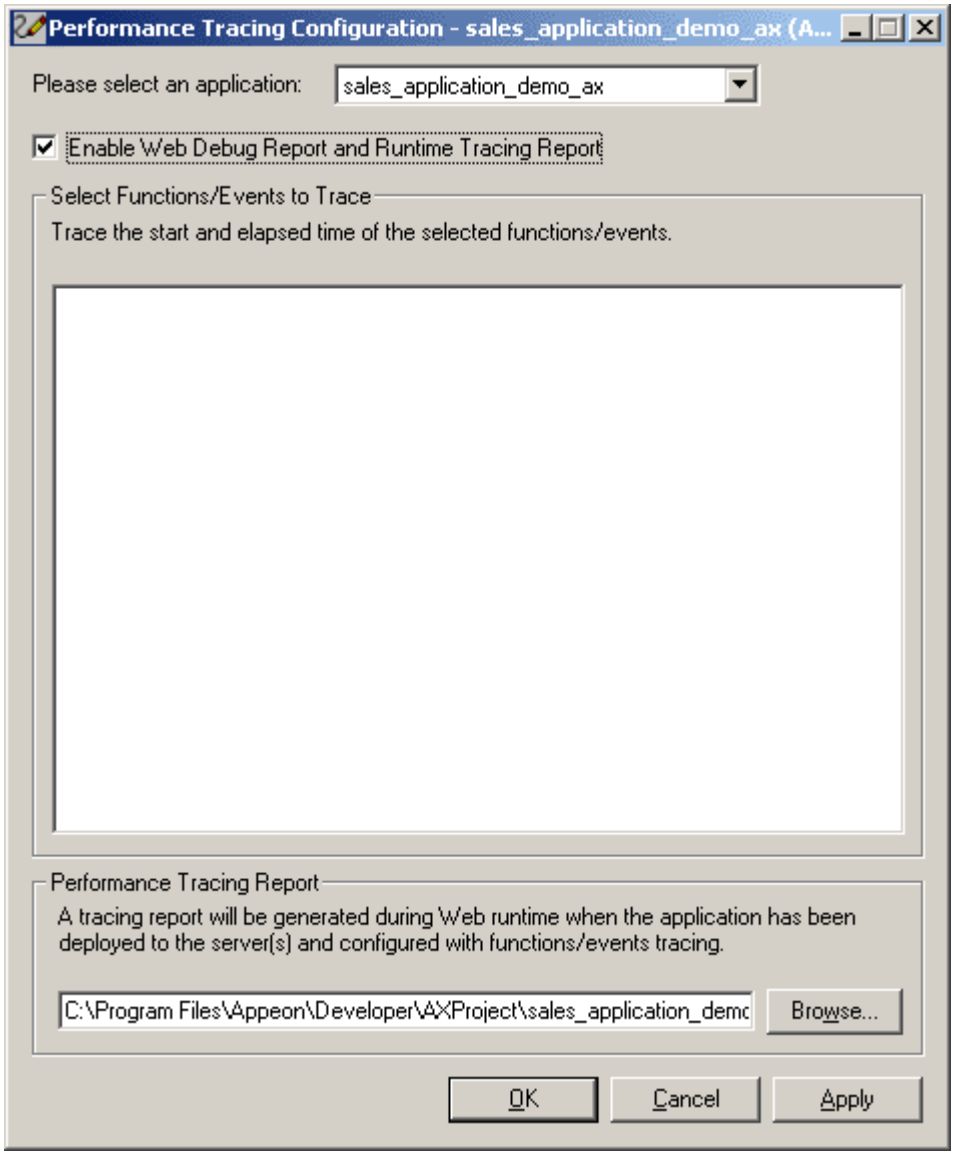


Table 8-3: Performance tracing configuration window

Option	Description
Application dropdown listbox	Lists the application profiles configured in Application Profiles. The default application profile is selected.
Enable Web Debug Report and Runtime Tracing Report check box	Enables the runtime report generation. It must be selected before the application deployment and configuration.
Select Functions/Events to Trace treeview	Displays all functions and events defined in the application for you to select which functions/events to track. If the *Enable Web Debug Report and Runtime Tracing Report* check box is not selected before deployment, the treeview will be empty because no function or event list is generated during deployment.
Browse	Allows you to select a location for storing the Web Performance Runtime Tracing Report on the Developer machine.

8.2.4 Running the application to generate runtime reports

After you have selected the functions and events that you want to track on the Web Debugging Configuration window or the Performance Tracing Configuration window, you can run the application to generate the runtime reports.

Access the Web application from the Developer machine where the application is deployed and configured, so that the runtime reports can be successfully generated on the Developer machine.

Before you run the application, verify the following settings in Application Profiles:

- the application is selected as the default application;
- the deployment option (Pure-JavaScript or Appeon Xcelerator) for the default application is identical to the deployment option used in both deployment and configuration;
- the debug mode (Debug PS/JS, Debug JS, or Encrypted) for the default application is identical to the debug mode used in both deployment and configuration.

You can now run the application to generate the runtime reports. For detailed instructions on running Web applications, refer to Chapter 9: *Running Web Applications*.

8.3 Working with Runtime Reports

8.3.1 Accessing runtime reports

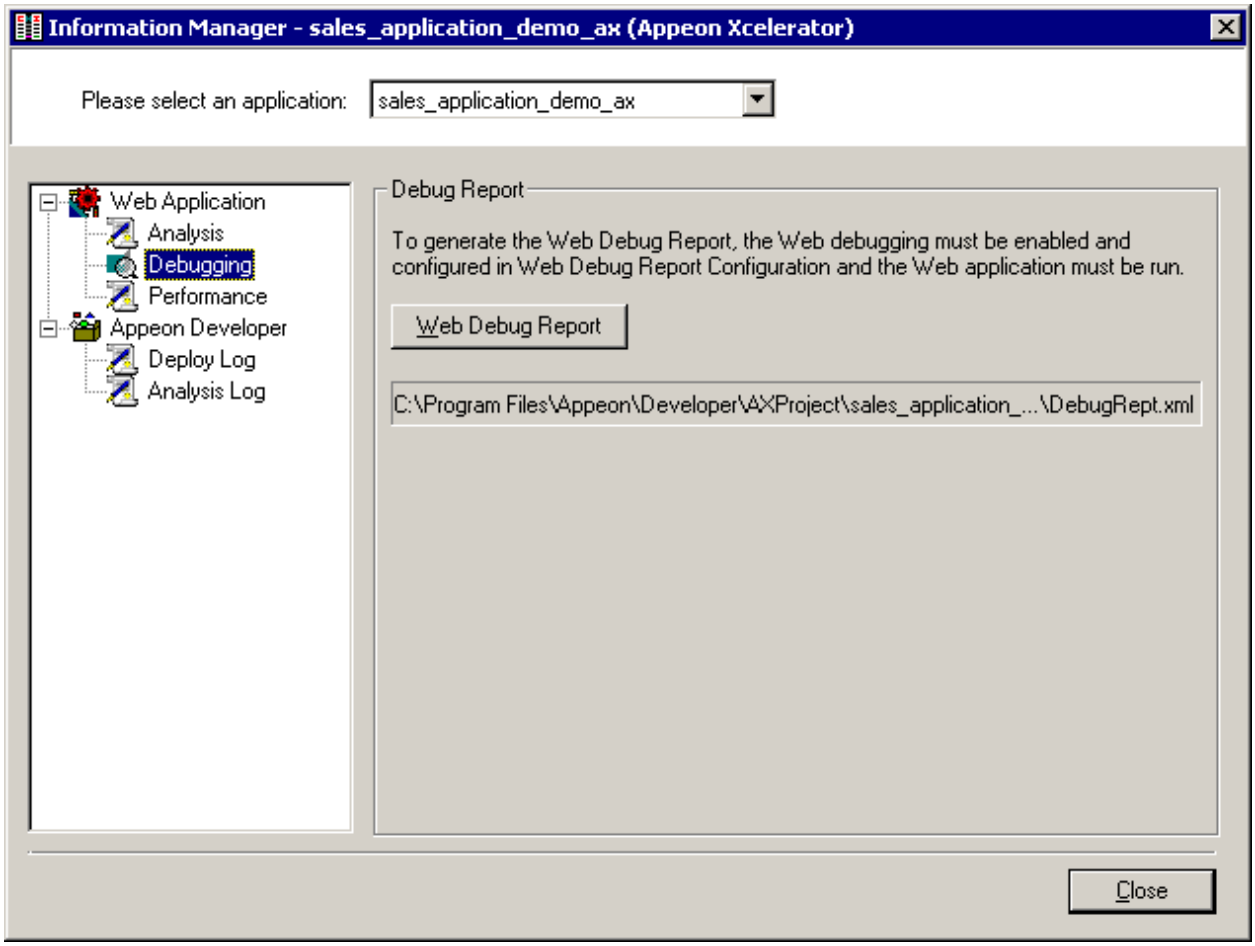
The generated runtime reports are saved in the specified location on the Developer machine. You can use the Information Manager to view them.

Step 1 – Click the *Information* button () on the Appeon Developer toolbar.

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

Step 3 – Click *Debugging* in the left treeview and the *Web Debug Report* button will be displayed on the right of the page, as shown in Figure 8-4, or click *Performance* in the left treeview and the *Web Performance Report* button will be displayed on the right of the page.

Figure 8-4: Information Manager

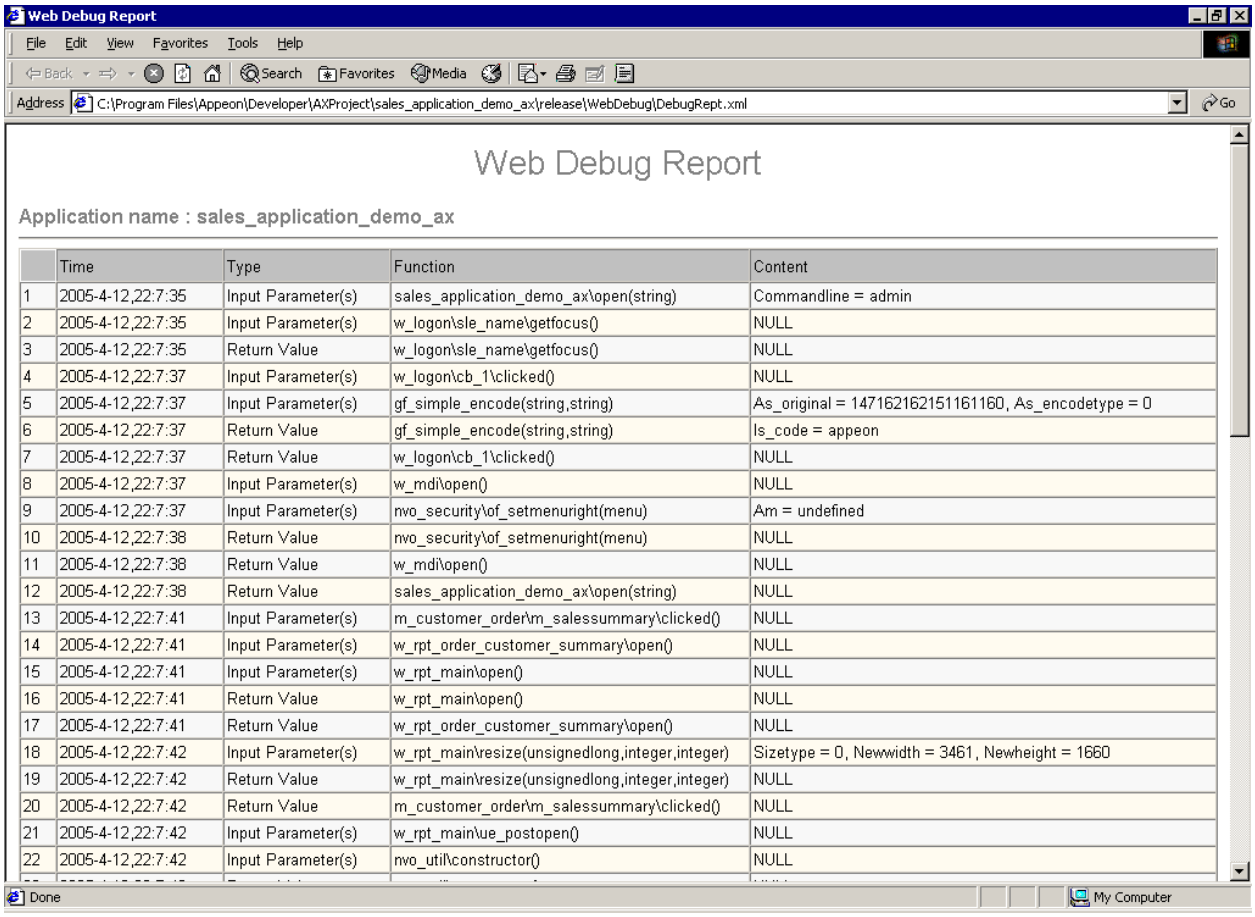


Step 4 – Click the *Web Debug Report* button or the *Web Performance Report* button to open the report.

8.3.2 Working with Web Debug Report

The Web Debug Report shows detailed execution information of a function or event in the application, as shown in Figure 8-5.

Figure 8-5: Web Debug Report



	Time	Type	Function	Content
1	2005-4-12,22:7:35	Input Parameter(s)	sales_application_demo_ax\open(string)	Commandline = admin
2	2005-4-12,22:7:35	Input Parameter(s)	w_logon\sle_name\getfocus()	NULL
3	2005-4-12,22:7:35	Return Value	w_logon\sle_name\getfocus()	NULL
4	2005-4-12,22:7:37	Input Parameter(s)	w_logon\cb_1\clicked()	NULL
5	2005-4-12,22:7:37	Input Parameter(s)	gf_simple_encode(string,string)	As_original = 147162162151161160, As_encodetype = 0
6	2005-4-12,22:7:37	Return Value	gf_simple_encode(string,string)	ls_code = appeon
7	2005-4-12,22:7:37	Return Value	w_logon\cb_1\clicked()	NULL
8	2005-4-12,22:7:37	Input Parameter(s)	w_mdi\open()	NULL
9	2005-4-12,22:7:37	Input Parameter(s)	nvo_security\of_setmenuright(menu)	Am = undefined
10	2005-4-12,22:7:38	Return Value	nvo_security\of_setmenuright(menu)	NULL
11	2005-4-12,22:7:38	Return Value	w_mdi\open()	NULL
12	2005-4-12,22:7:38	Return Value	sales_application_demo_ax\open(string)	NULL
13	2005-4-12,22:7:41	Input Parameter(s)	m_customer_order\m_salessummary\clicked()	NULL
14	2005-4-12,22:7:41	Input Parameter(s)	w_rpt_order_customer_summary\open()	NULL
15	2005-4-12,22:7:41	Input Parameter(s)	w_rpt_main\open()	NULL
16	2005-4-12,22:7:41	Return Value	w_rpt_main\open()	NULL
17	2005-4-12,22:7:41	Return Value	w_rpt_order_customer_summary\open()	NULL
18	2005-4-12,22:7:42	Input Parameter(s)	w_rpt_main\resize(unsignedlong,integer,integer)	Sizetype = 0, Newwidth = 3461, Newheight = 1660
19	2005-4-12,22:7:42	Return Value	w_rpt_main\resize(unsignedlong,integer,integer)	NULL
20	2005-4-12,22:7:42	Return Value	m_customer_order\m_salessummary\clicked()	NULL
21	2005-4-12,22:7:42	Input Parameter(s)	w_rpt_main\ue_postopen()	NULL
22	2005-4-12,22:7:42	Input Parameter(s)	nvo_util\constructor()	NULL

Table 8-4: Web Debug Report

Column	Description
Time	Displays the time at which the function/event is called.
Type	Displays the type of the recorded information: • Input Parameters • Unsupported Code • Appeon Comments • Variable Information • Object Information • Function Return Value
Function	Displays the location and the name of the events and functions that are executed.
Content	Displays the recorded information according to the Type: • parameter name and value • unsupported script • Appeon comments • variable name and value • object name and class name • function return value You can use the content to help debug the application.

8.3.3 Working with Web Performance Runtime Tracing Report

The Web Performance Runtime Tracing Report window automatically refreshes every 15 seconds, dynamically displaying the latest runtime performance of a running Web application.

Figure 8-6: Web Performance Runtime Tracing Report

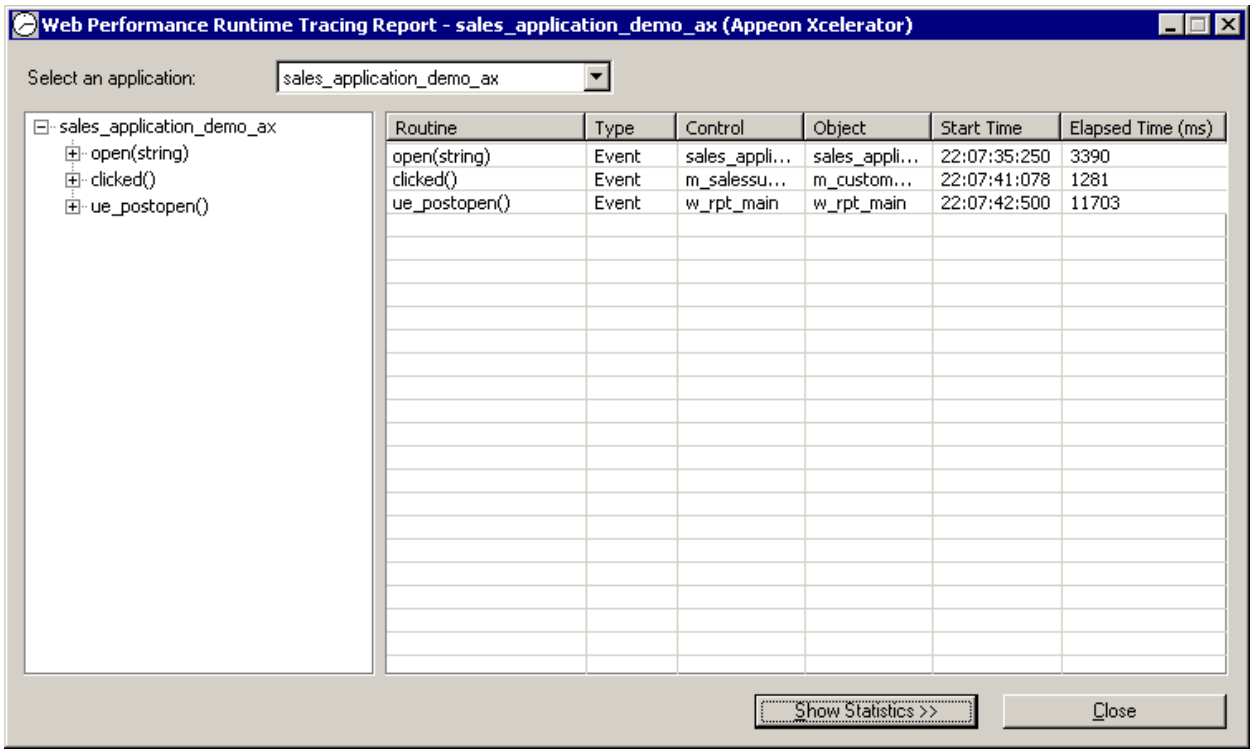


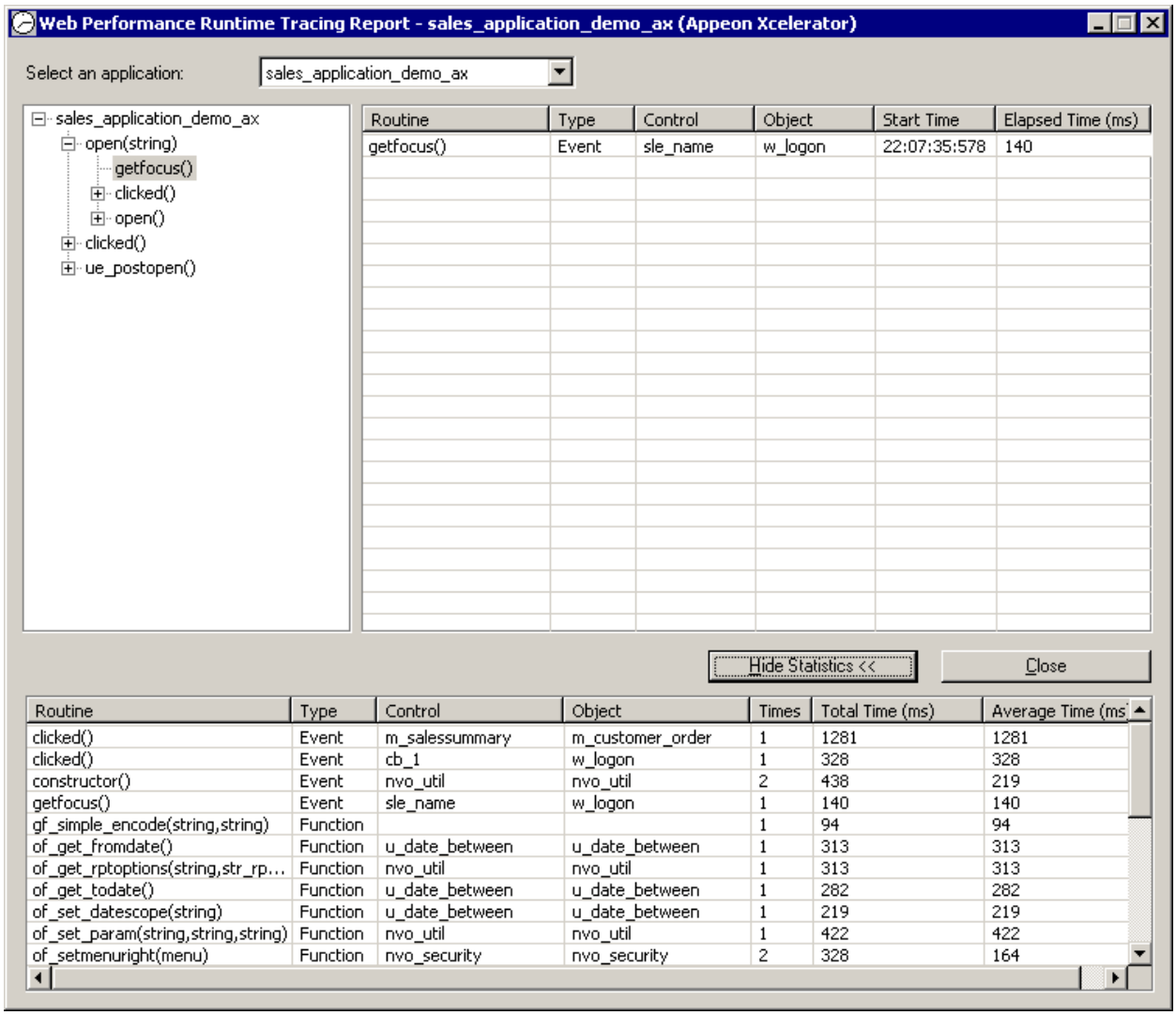
Table 8-5: Web Performance Runtime Tracing Report

Option	Description
Application list box	Lists the application profiles configured in Application Profiles. The default application profile is selected.
Triggered event treeview	Displays the triggered events in a treeview structure. You can expand and select the items in the treeview to view detailed information of the executed events/functions in the right box.
Routine	Displays the name of the event/function that is triggered.
Type	Displays the type of the triggered routine, event or function.
Control	Displays the control that contains the routine.
Object	Displays the root object that contains the routine.
Start Time	Displays the time at which the routine is called.
Elapsed Time	Displays the elapsed time used to complete the execution of a routine.
Show/Hide Statistic	Click this to show or hide the summary of each routine: the number of calls, the total elapsed time, and average time used in each call.

For example, click the getfocus() event under the open(string) in the tree view. Detailed information is displayed in the right box as shown in Figure 8-7. The getfocus() event of

sle_name control is called at 22:07:35:578, and takes 140 milliseconds to complete the call. From the Elapsed Time column, you can find out which routines impede performance.

Figure 8-7: Web Performance Runtime Tracing Report



Expand the report by clicking the *Show Statistics* button at the bottom of the window. The statistics shown in the expanded box summarize the time each routine is called, and the total elapsed time and average time used in each call, as shown in Figure 8-7.

9 Running Web Applications

9.1 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*.

9.2 Requirements

9.2.1 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 need to make sure you configure Internet Explorer's settings accordingly.

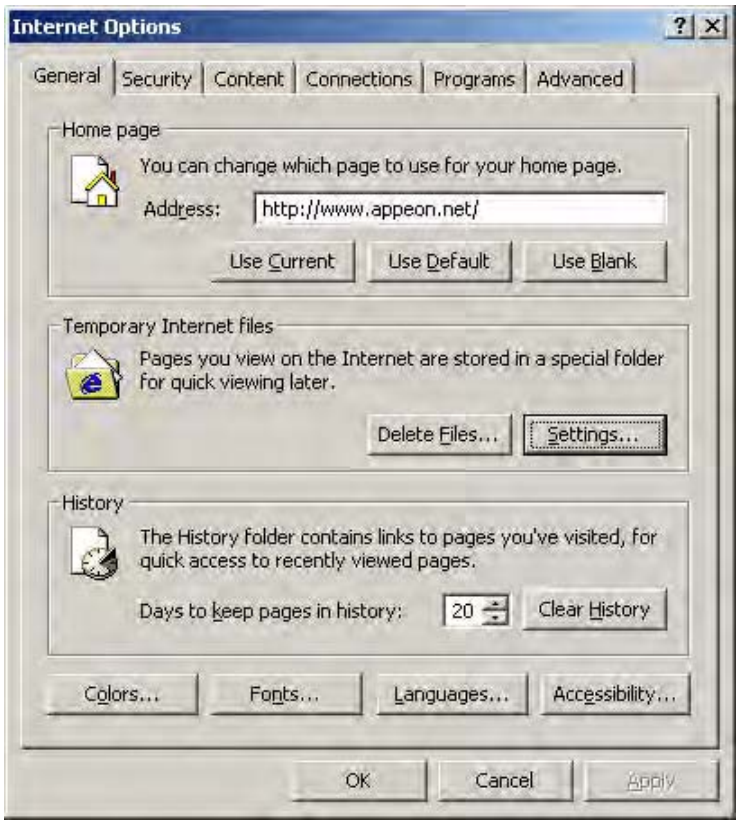
9.2.1.a Settings for temporary Internet files

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 9-1.

Figure 9-1: Internet Options in Internet Explorer



Step 2 – Click the *Settings* button in the *Temporary Internet Files* group box, as shown in Figure 9-2.

Figure 9-2: Temporary Internet File Settings

Step 3 – Select the radio button to *Automatically* check for newer versions of stored pages, as shown in Figure 9-2.

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

9.2.1.b Advanced settings for temporary Internet Explorer files folder

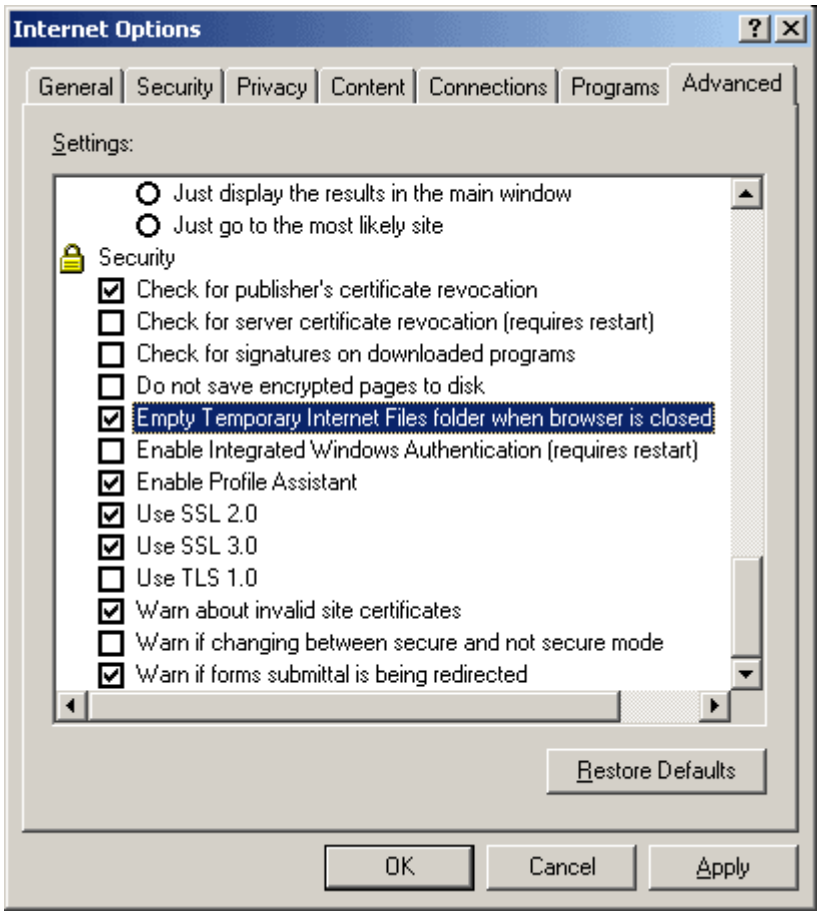
During development, the "Temporary Internet Files folder" should be set to automatically empty each time Internet Explorer is closed. This will force Internet Explorer to re-download the entire application every time it is opened, to ensure the latest version of the application is present, and to prevent 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 9-3.

Step 2 – Check the *Empty Temporary Internet files folder when browser is closed* check box under the Security section, as shown in Figure 9-3.

Figure 9-3: 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.

9.2.1.c Advanced settings for proxy server

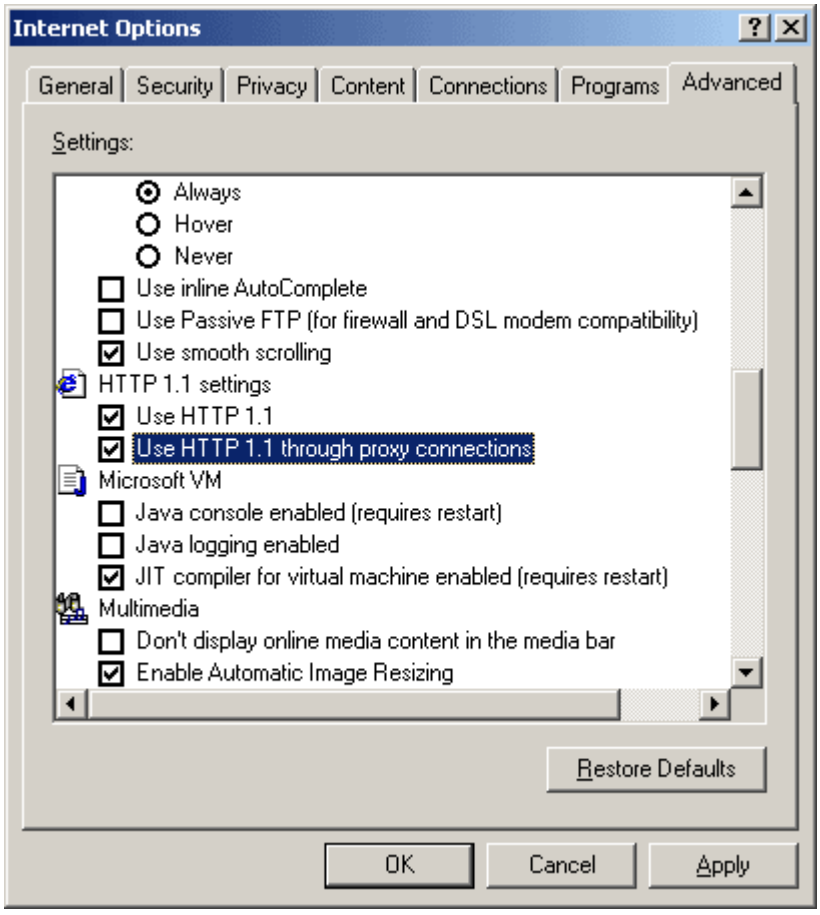
If the Client visits the Web application through a proxy server, you need to select the *Use HTTP 1.1 through proxy connections* option under the HTTP 1.1 settings section, as shown in Figure 9-4.

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 9-4.

Step 2 – Check the *Use HTTP 1.1 through proxy connections* check box under the HTTP 1.1 settings section, as shown in Figure 9-4.

Figure 9-4: 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.

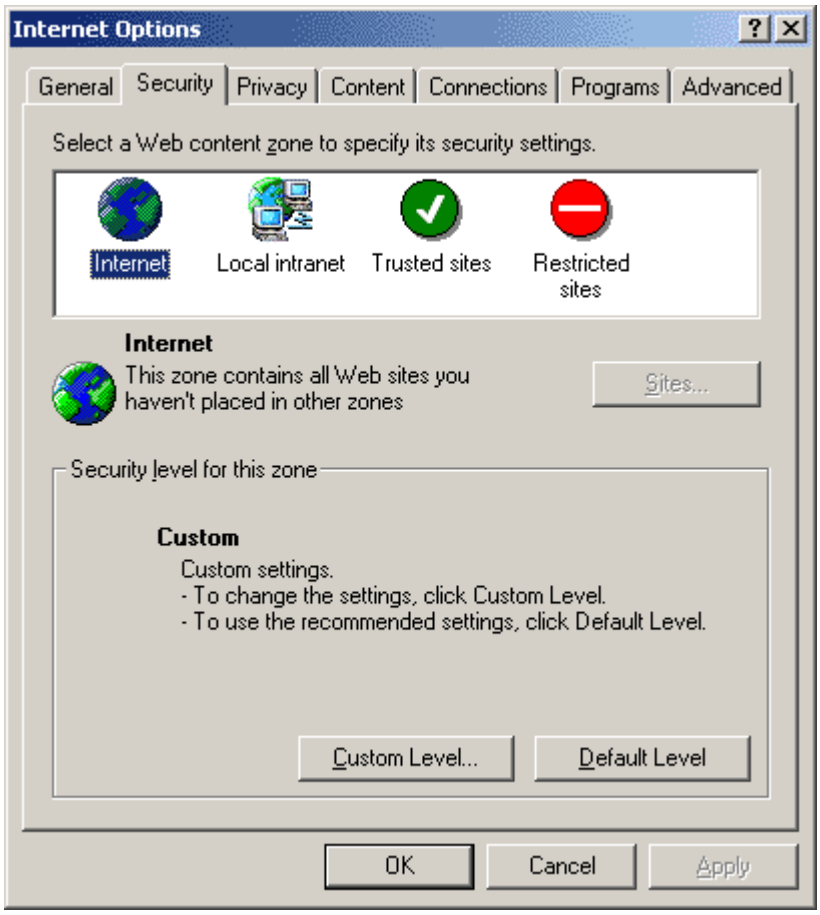
9.2.1.d Security settings for ActiveX controls and Appeon Xcelerator plug-in download

If the Web application uses any of the following features: call OLE objects, DLL files, runs executable programs on the Client, or the Web application is deployed with Appeon Xcelerator, 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) or 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.

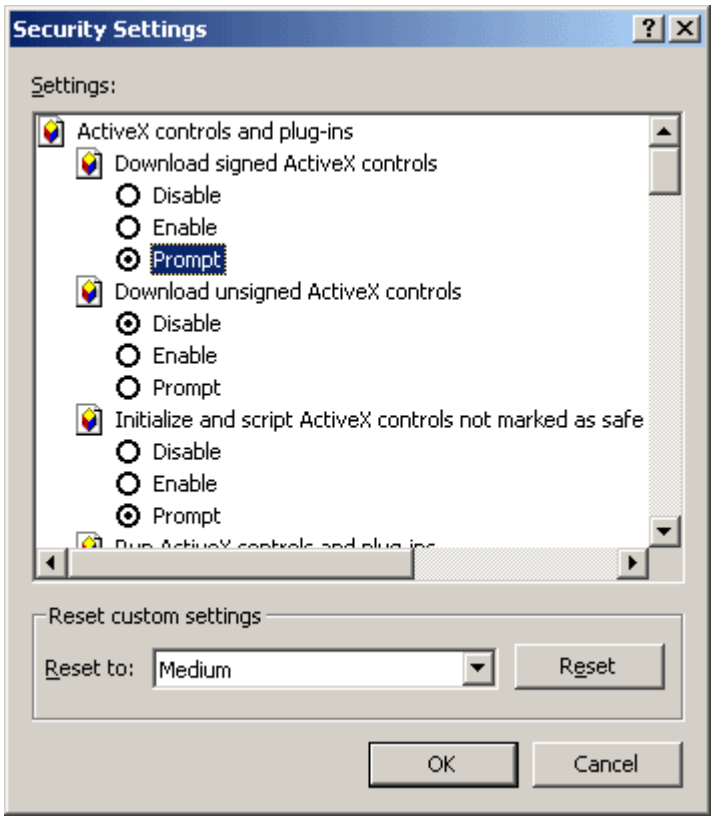
Step 2 – Verify that you selected an appropriate Web content zone, as shown in Figure 9-5.

Figure 9-5: Select a Web content zone



Step 3 – Click the *Custom Level…*button in the *Security* tab. The Security Settings page is displayed, as shown in Figure 9-6.

Figure 9-6: 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
- Initialize and script ActiveX controls not marked as safe
- Run ActiveX controls and plug-ins
- Script ActiveX controls marked safe for scripting
- File download

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.

9.2.2 Language setting requirements

The language of the data in the database used by the application should match the Windows 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 Windows operating system:

Step 1 – Select Windows Start Menu | Settings | Control Panel.

Step 2 – Click Regional Options, as shown in Figure 9-7.

Figure 9-7: 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.

9.2.3 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.

9.3 Accessing Web applications

9.3.1 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/index.html

A complete Web application URL consists of four parts:

Table 9-1: Web URL

Web URL	Description
Protocol	Uses "http://" typically. If your Web Server is configured as an SSL Web Server, use "https://".
IP address or domain name	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 Appeon Server.
HTTP port	(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.
Web file path	Specifies the path of the entry page (index.html) of the Web application, relative to the home directory of the Web server. It is the Web deployment path set for the application in Application Profiles, or "*Web deployment path*/index.html".

9.3.2 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 9.2: *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.

9.3.2.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 9-8, prompting you to select which application to run.

Figure 9-8: 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*.

9.4 Appeon Enhanced Web Features

Appeon provides a Web enhancement, the Appeon DataWindow menu, for all deployed DataWindows. When right-clicking on a deployed DataWindow, you can access the Appeon DataWindow popup menus:

- For HTML DataWindows, the popup menus are Find, Find Next, Find Previous, Sort, Filter, Save As File…, Print (only available in Appeon Xcelerator deployment), Print to PDF (or Print View in Pure-JavaScript deployments) and Image View.
- For Image DataWindows, the popup menus are Print to PDF, Save As File…, Zoom In and Zoom Out. Image DataWindows deployed in Pure-JavaScript will not have the Appeon DataWindow menu.

Note: When using the Print to PDF (or Print View), Image View, or Save As File menus, the previewed/printed/saved data is the data retrieved from the database instead of the modified data in the deployed DataWindow. If you want to preview/print out/save the modified data,

you must save the modified data in the database first and then retrieve the data from the database again.

9.4.1 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_ax.pbl* and *appeon_workarounds_js.pbl*. For detailed instructions on how to use *appeonpopmenu* and a*ppeonpopmenuon* functions in a DataWindow, refer to the *Appeon Migration Guide*.

9.4.2 Web DataWindow Menu for HTML DataWindows

Composite, Freeform, Grid, Group and Tabular HTML DataWindows have the following enhanced features: Find, Find Next, Find Previous, Sort, Filter, Print, Print to PDF (or Print View), Image View and Save As File, as shown in Figure 9-9 and Figure 9-10. The Print feature is only available on the DataWindow popup menu for applications deployed with Appeon Xcelerator. It allows users to print DataWindows directly to printers connected to the Client.

Perform the following steps to use these features in Web DataWindows deployed in Pure-JavaScript. The steps to use these features in applications deployed in Appeon Xcelerator are very similar.

Figure 9-9: Appeon DataWindow popup menu in Appeon Xcelerator deployment

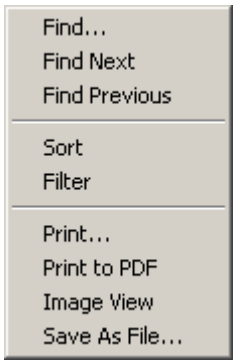


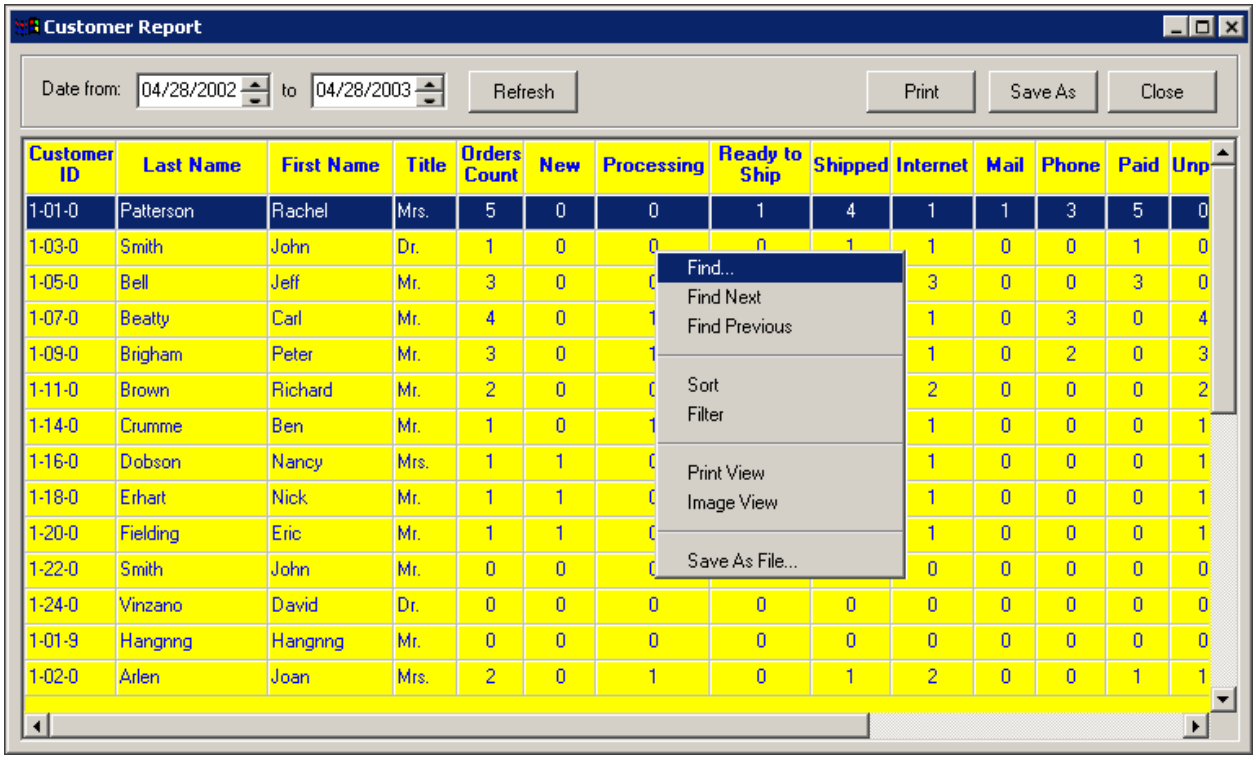
Figure 9-10: Appeon DataWindow popup menu in Pure-JavaScript deployment

9.4.2.a Find

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

Step 1 – Right-click the deployed DataWindow and select *Find* from the popup menu, as shown in Figure 9-11.

Figure 9-11: Appeon DataWindow Menu

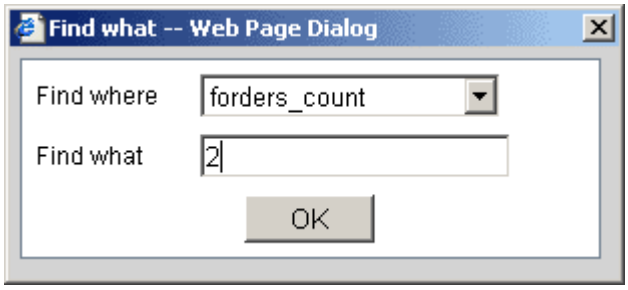


Customer ID	Last Name	First Name	Title	Orders Count	New	Processing	Ready to Ship	Shipped	Internet	Mail	Phone	Paid	Unp
1-01-0	Patterson	Rachel	Mrs.	5	0	0	1	4	1	1	3	5	0
1-03-0	Smith	John	Dr.	1	0	0	0	1	1	0	0	1	0
1-05-0	Bell	Jeff	Mr.	3	0	0			3	0	0	3	0
1-07-0	Beatty	Carl	Mr.	4	0	1			1	0	3	0	4
1-09-0	Brigham	Peter	Mr.	3	0	1			1	0	2	0	3
1-11-0	Brown	Richard	Mr.	2	0	0			2	0	0	0	2
1-14-0	Crumme	Ben	Mr.	1	0	1			1	0	0	0	1
1-16-0	Dobson	Nancy	Mrs.	1	1	0			1	0	0	0	1
1-18-0	Erhart	Nick	Mr.	1	1	0			1	0	0	0	1
1-20-0	Fielding	Eric	Mr.	1	1	0			1	0	0	0	1
1-22-0	Smith	John	Mr.	0	0	0			0	0	0	0	0
1-24-0	Vinzano	David	Dr.	0	0	0	0	0	0	0	0	0	0
1-01-9	Hangnng	Hangnng	Mr.	0	0	0	0	0	0	0	0	0	0
1-02-0	Arlen	Joan	Mrs.	2	0	1	0	1	2	0	0	1	1

Step 2 – Select the criteria for the search.

Use the Find function to find all rows where the orders count column is equal to 2, as shown in Figure 9-12.

Figure 9-12: Find



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.

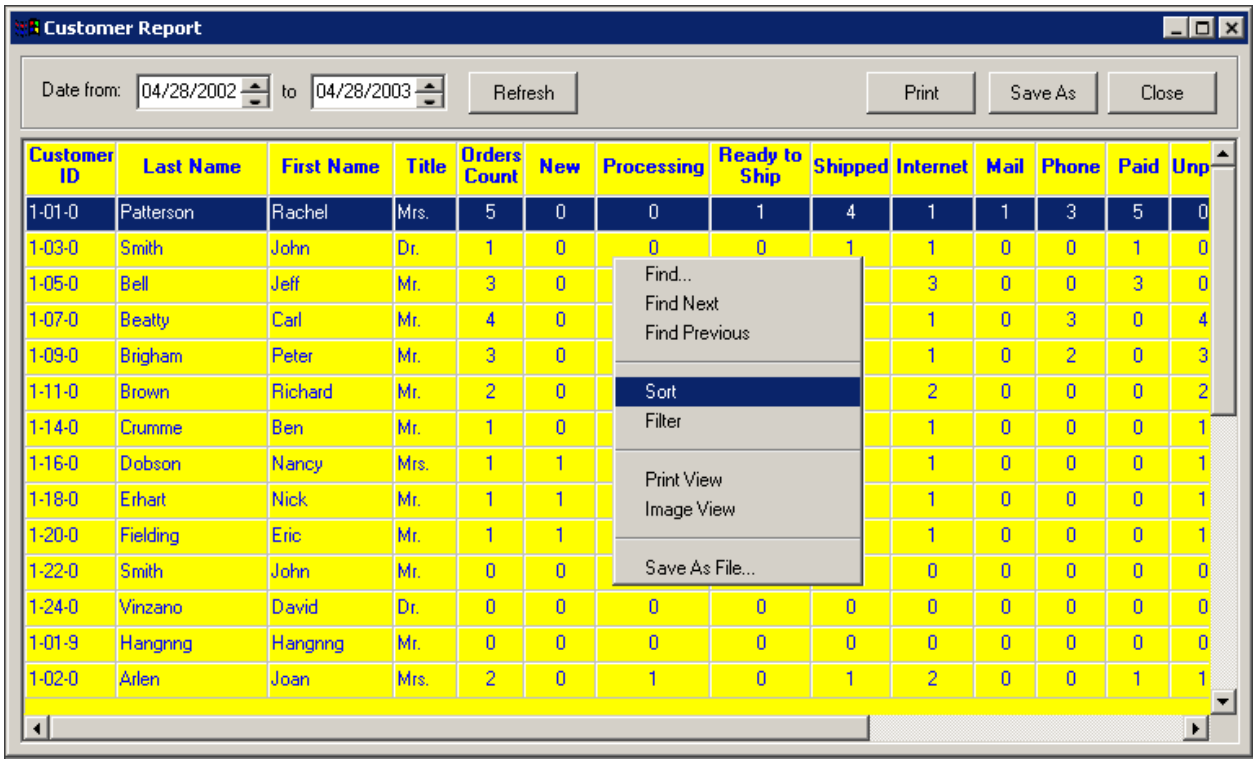
9.4.2.b Sort and filter

The user can sort data by the column in ascending or descending order, or filter data using a number of functions.

To sort data:

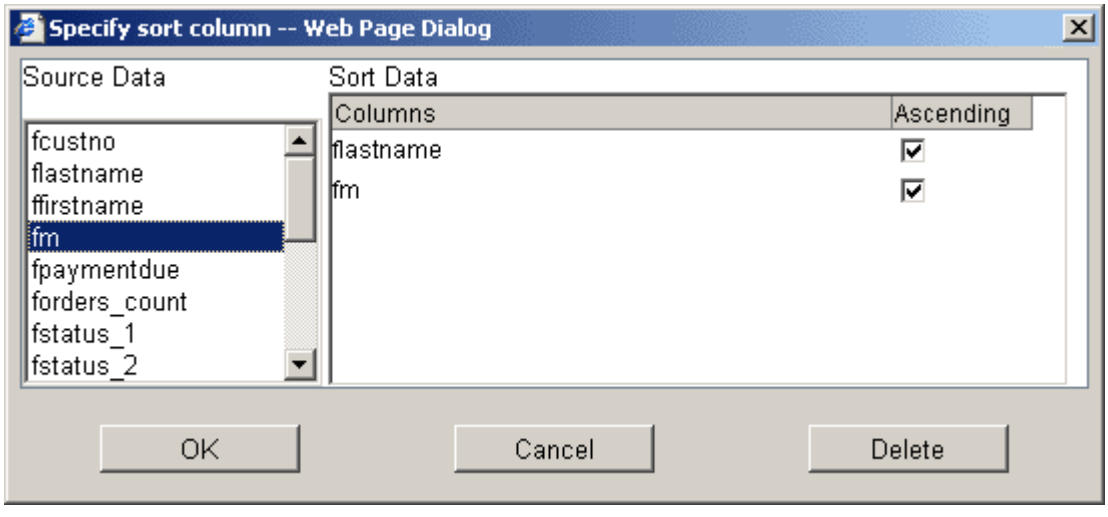
Step 1 – Right-click the deployed DataWindow and select *Sort* from the popup menu, as shown in Figure 9-13.

Figure 9-13: Sort



Step 2 – Specify the column used to sort the data in the DataWindow by clicking on the column name in the left list box. The column name appears on the right, as shown in Figure 9-14.

Figure 9-14: Specify sort column

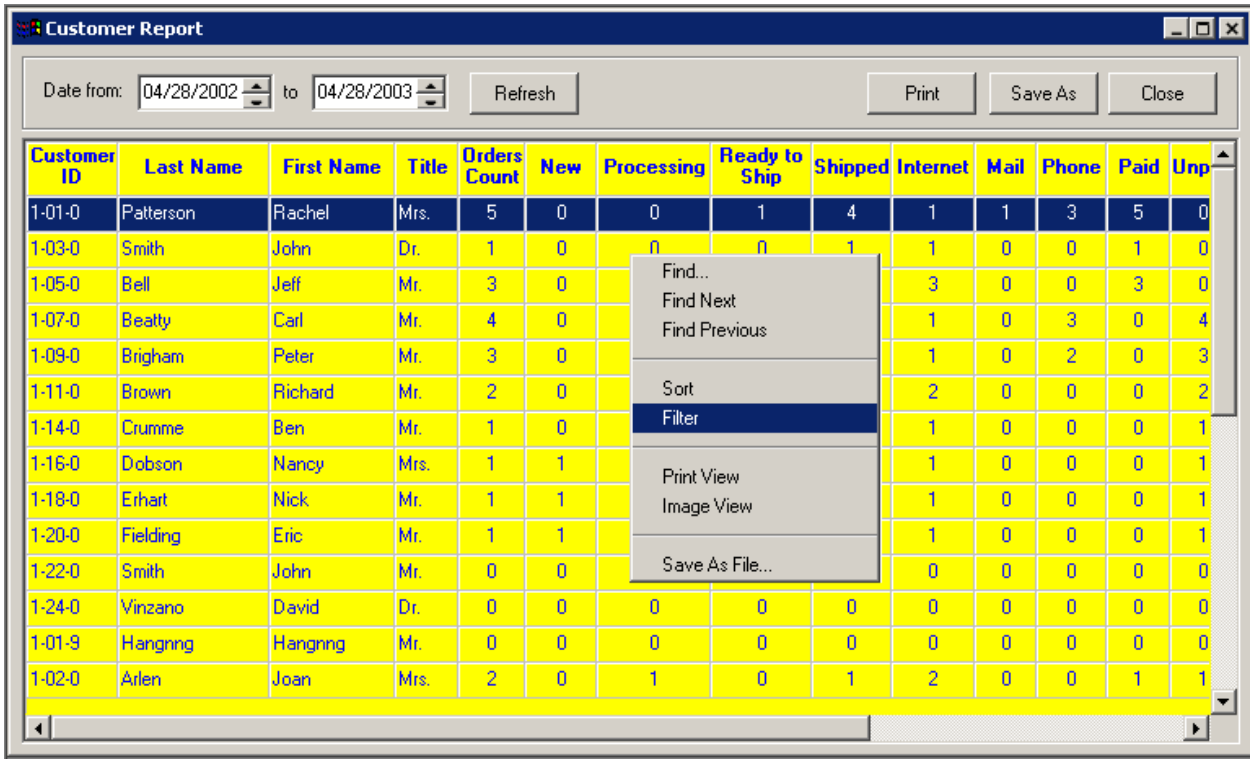


Step 3 – Check the *Ascending* check box to sort the data in ascending order or uncheck the *Ascending* check box to sort the data in descending order. Click *Delete* to remove a column from being sorted. Click *OK* to accept the changes. The DataWindow automatically refreshes and the data is sorted.

To filter data:

Step 1 – Right-click the deployed DataWindow and select *Filter* from the popup menu, as shown in Figure 9-15.

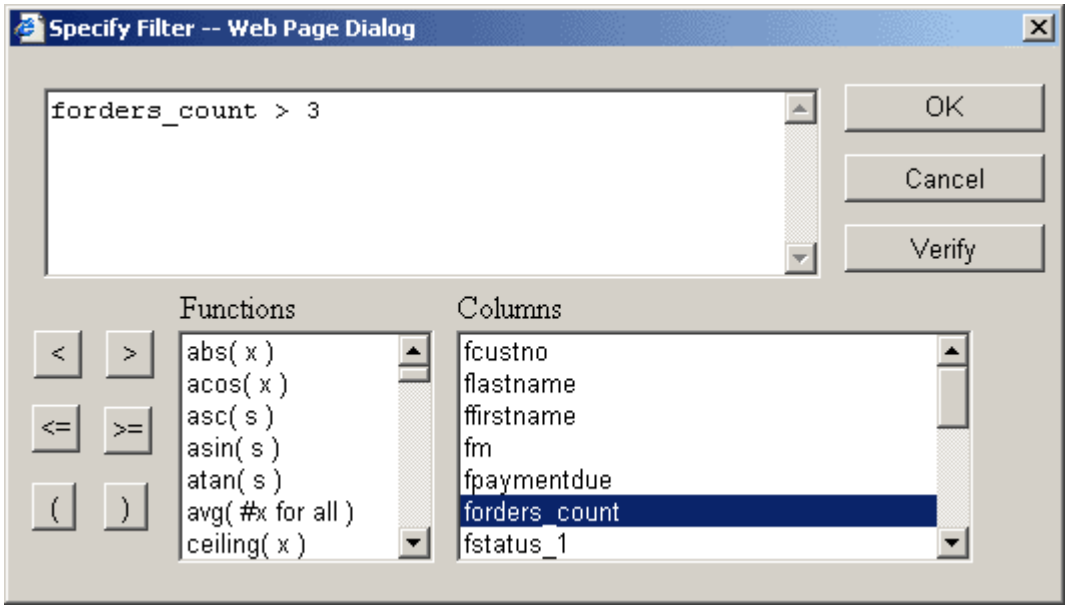
Figure 9-15: Filter



Customer ID	Last Name	First Name	Title	Orders Count	New	Processing	Ready to Ship	Shipped	Internet	Mail	Phone	Paid	Unp
1-01-0	Patterson	Rachel	Mrs.	5	0	0	1	4	1	1	3	5	0
1-03-0	Smith	John	Dr.	1	0	0	0	1	1	0	0	1	0
1-05-0	Bell	Jeff	Mr.	3	0				3	0	0	3	0
1-07-0	Beatty	Carl	Mr.	4	0				1	0	3	0	4
1-09-0	Brigham	Peter	Mr.	3	0				1	0	2	0	3
1-11-0	Brown	Richard	Mr.	2	0				2	0	0	0	2
1-14-0	Crumme	Ben	Mr.	1	0				1	0	0	0	1
1-16-0	Dobson	Nancy	Mrs.	1	1				1	0	0	0	1
1-18-0	Erhart	Nick	Mr.	1	1				1	0	0	0	1
1-20-0	Fielding	Eric	Mr.	1	1				1	0	0	0	1
1-22-0	Smith	John	Mr.	0	0				0	0	0	0	0
1-24-0	Vinzano	David	Dr.	0	0	0	0	0	0	0	0	0	0
1-01-9	Hangnng	Hangnng	Mr.	0	0	0	0	0	0	0	0	0	0
1-02-0	Arlen	Joan	Mrs.	2	0	1	0	1	2	0	0	1	1

Step 2 – Specify the filter by selecting the functions and columns. Click *Verify* to determine if the filter expression is valid. Click *OK* to accept the filter, as shown in Figure 9-16. The DataWindow automatically refreshes and the data is filtered.

Figure 9-16: Specify Filter



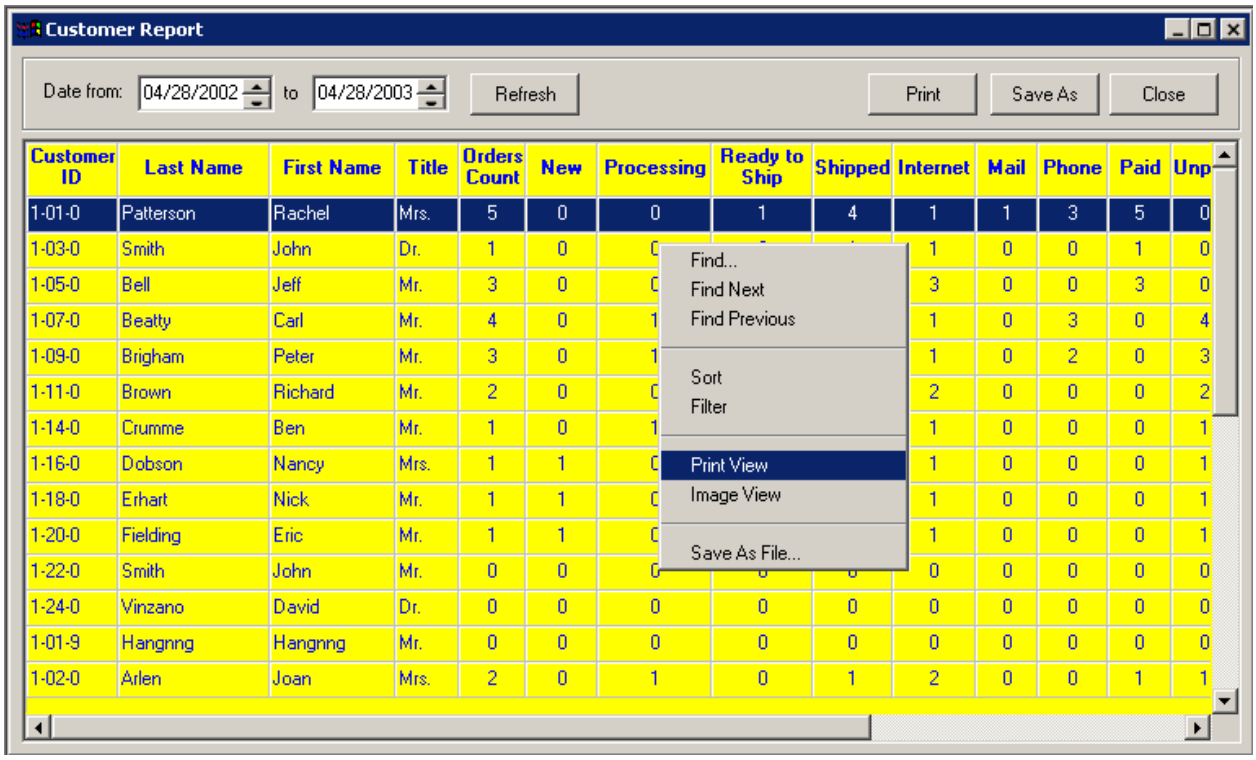
9.4.2.c DataWindow printing

You can print Tabular, Freeform, and Grid DataWindows to PDF or GIF file while viewing the application on Web.

Print View or Print to PDF

Step 1 – Right-click on a DataWindow control and select "Print View" (or "Print to PDF" in Appeon Xcelerator deployments) from the popup menu to view a deployed DataWindow in PDF, as shown in Figure 9-17.

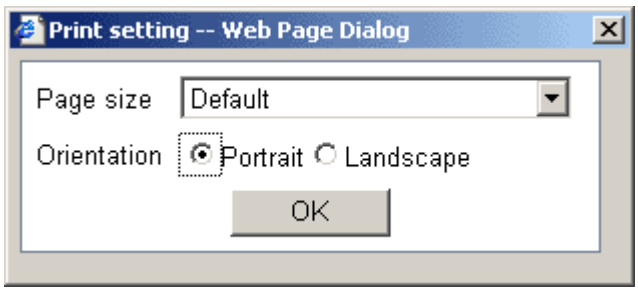
Figure 9-17: Print View



Customer ID	Last Name	First Name	Title	Orders Count	New	Processing	Ready to Ship	Shipped	Internet	Mail	Phone	Paid	Unp
1-01-0	Patterson	Rachel	Mrs.	5	0	0	1	4	1	1	3	5	0
1-03-0	Smith	John	Dr.	1	0	0			1	0	0	1	0
1-05-0	Bell	Jeff	Mr.	3	0	0			3	0	0	3	0
1-07-0	Beatty	Carl	Mr.	4	0	1			1	0	3	0	4
1-09-0	Brigham	Peter	Mr.	3	0	1			1	0	2	0	3
1-11-0	Brown	Richard	Mr.	2	0	0			2	0	0	0	2
1-14-0	Crumme	Ben	Mr.	1	0	1			1	0	0	0	1
1-16-0	Dobson	Nancy	Mrs.	1	1	0			1	0	0	0	1
1-18-0	Erhart	Nick	Mr.	1	1	0			1	0	0	0	1
1-20-0	Fielding	Eric	Mr.	1	1	0			1	0	0	0	1
1-22-0	Smith	John	Mr.	0	0	0			0	0	0	0	0
1-24-0	Vinzano	David	Dr.	0	0	0	0	0	0	0	0	0	0
1-01-9	Hangnng	Hangnng	Mr.	0	0	0	0	0	0	0	0	0	0
1-02-0	Arlen	Joan	Mrs.	2	0	1	0	1	2	0	0	1	1

Step 2 – A Web dialog box appears, prompting you to select the page size and orientation, as shown in Figure 9-18.

Figure 9-18: Print Setting

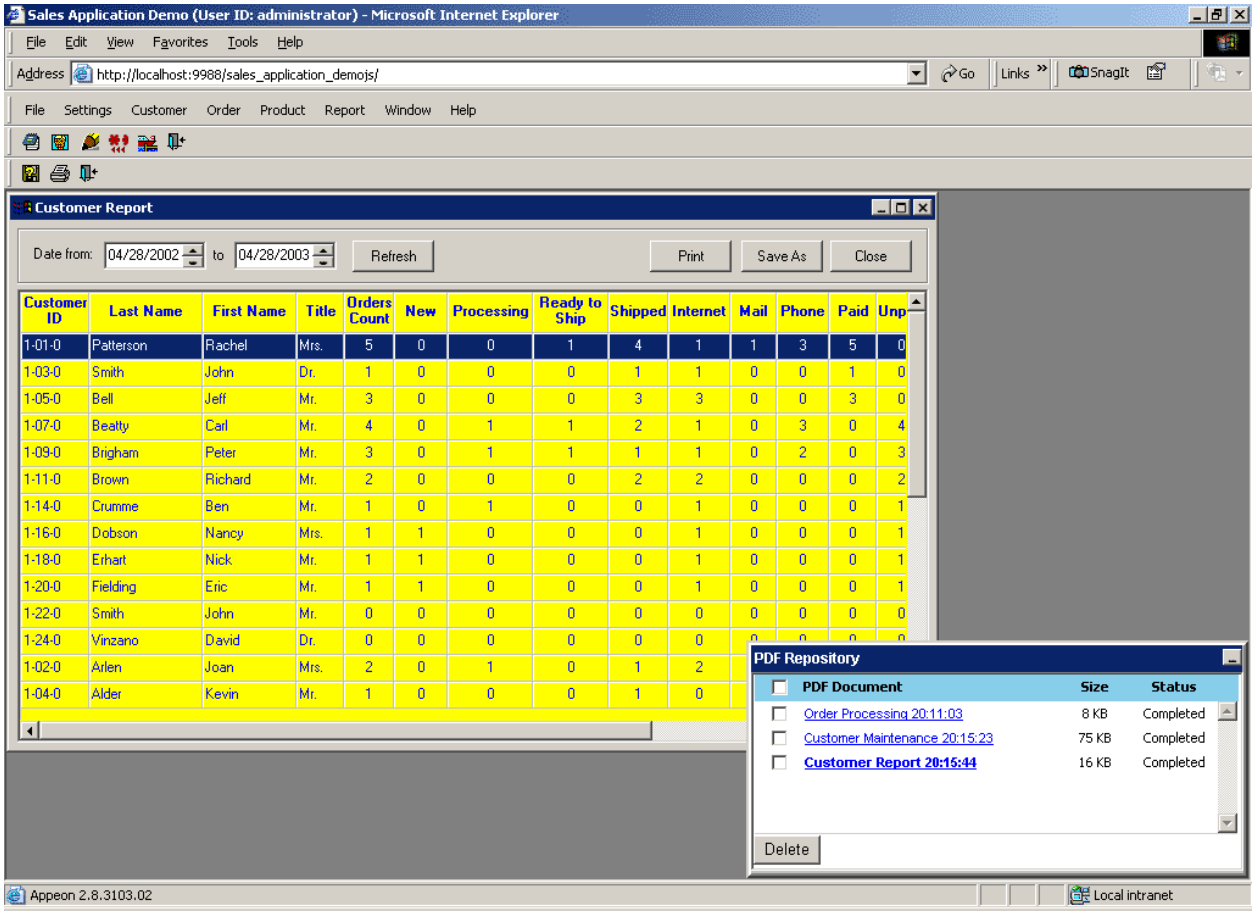


A PDF Repository window is displayed at the right bottom corner of the Internet Explorer window, as shown in Figure 9-19.

You can print several PDF files at the same time. All the printed documents are listed in order by time of creation. Click the PDF file you would like to view. It is then loaded into a separate Internet Explorer window using the Adobe Acrobat plug-in. You can view the history of all the printed documents at any time; you can also delete them from the history.

- In Pure-JavaScript deployments, the position of the PDF Repository is fixed to the bottom right corner of the browser. You can minimize the PDF Repository and display it again by clicking the text "PDF Repository".
- In Appeon Xcelerator deployments, you can move the PDF Repository window around the Internet Explorer window, minimize it to an icon in the task bar, and click the icon in the task bar to open the window again. You can also close it. When you select the "Print to PDF" menu again in the same running application, the PDF Repository window is opened with a history of each document that has been printed. You can save the printed PDF file to the local machine by right-clicking the link to the PDF file in the PDF Repository window and selecting *Save Target As* from the popup menu.
- In Pure-JavaScript deployments, if the window containing the printed DataWindow is a response window, you cannot move the focus to the PDF Repository window or select the printed file there unless you release the focus first by closing the response window.

Figure 9-19: Print PDF file

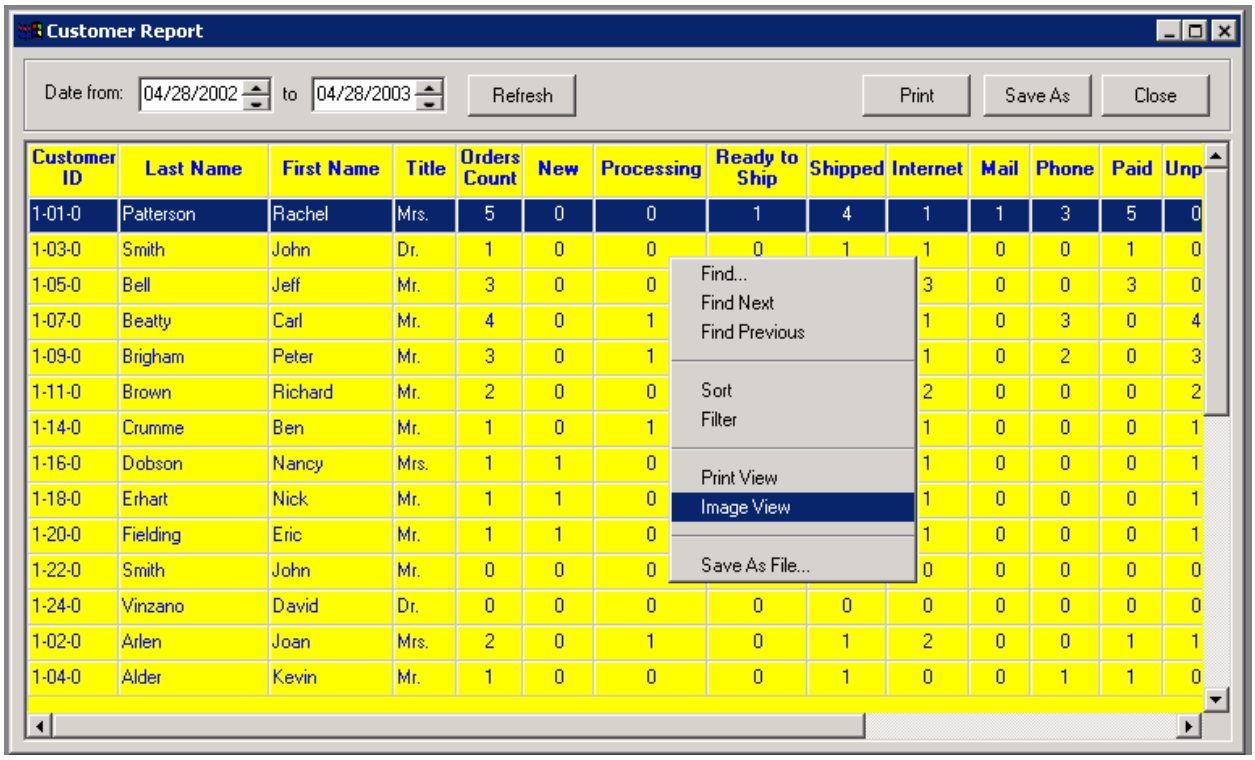


Note: If the DataWindow style is Freeform and the value of the Border property is set to Box(2), the borders of the table cells in the printed PDF DataWindow is not completely displayed.

Image View

To view a deployed DataWindow as an image in Internet Explorer, select "Image View" from the popup menu, as shown in Figure 9-20.

Figure 9-20: Print image file



Customer ID	Last Name	First Name	Title	Orders Count	New	Processing	Ready to Ship	Shipped	Internet	Mail	Phone	Paid	Unp
1-01-0	Patterson	Rachel	Mrs.	5	0	0	1	4	1	1	3	5	0
1-03-0	Smith	John	Dr.	1	0	0	0	1	1	0	0	1	0
1-05-0	Bell	Jeff	Mr.	3	0	0			3	0	0	3	0
1-07-0	Beatty	Carl	Mr.	4	0	1			1	0	3	0	4
1-09-0	Brigham	Peter	Mr.	3	0	1			1	0	2	0	3
1-11-0	Brown	Richard	Mr.	2	0	0			2	0	0	0	2
1-14-0	Crumme	Ben	Mr.	1	0	1			1	0	0	0	1
1-16-0	Dobson	Nancy	Mrs.	1	1	0			1	0	0	0	1
1-18-0	Erhart	Nick	Mr.	1	1	0			1	0	0	0	1
1-20-0	Fielding	Eric	Mr.	1	1	0			1	0	0	0	1
1-22-0	Smith	John	Mr.	0	0	0			0	0	0	0	0
1-24-0	Vinzano	David	Dr.	0	0	0	0	0	0	0	0	0	0
1-02-0	Arlen	Joan	Mrs.	2	0	1	0	1	2	0	0	1	1
1-04-0	Alder	Kevin	Mr.	1	0	0	0	1	0	0	1	1	0

The GIF image(s) are loaded into a separate Internet Explorer window.

There is a navigation bar at the top for accessing the additional DataWindow pages.

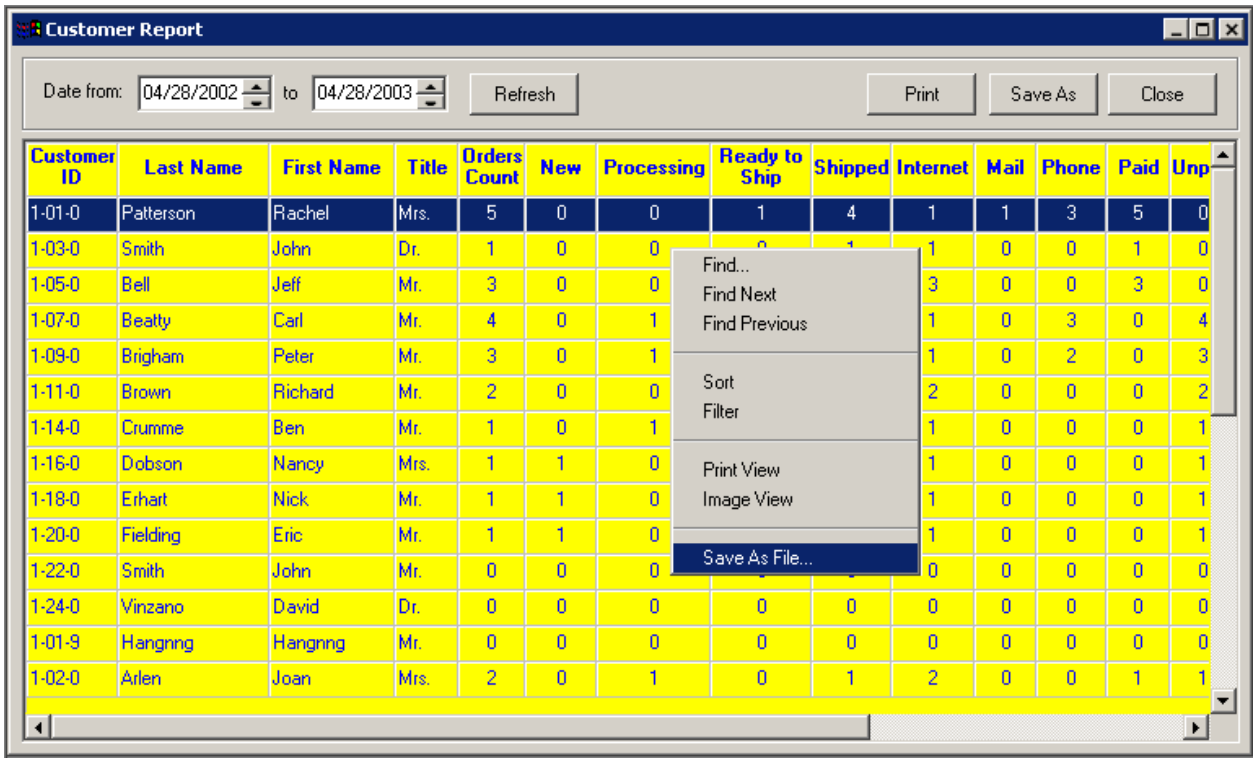
Right-click the GIF image and the same options as the GIF DataWindow will be available: Print View (or Print to PDF), Save As File, Zoom In, and Zoom Out. For more information on these options, refer to Section 9.4.3: *Web DataWindow Menu for Image DataWindow*.

9.4.2.d DataWindow SaveAs

The user can save the contents of a DataWindow in Windows Metafile, text, Microsoft Excel or HTML table formats.

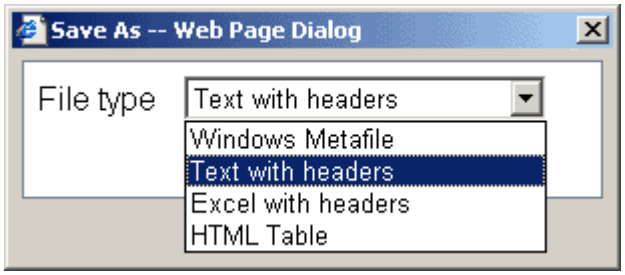
Step 1 – Right-click on a DataWindow control and select “Save As File…” from the popup menu, as shown in Figure 9-21.

Figure 9-21: Save As File…



Step 2 – Choose the file type from the popup dialog box, as shown in Figure 9-22.

Figure 9-22: Select file type in Pure-JavaScript deployment



In Appeon Xcelerator deployments, you are able to specify the file location, file name and file type in the Save As window, as shown in Figure 9-23. When clicking *Save*, the file will be saved to the specified directory on the local machine.

Figure 9-23: Save As window in Appeon Xcelerator deployment

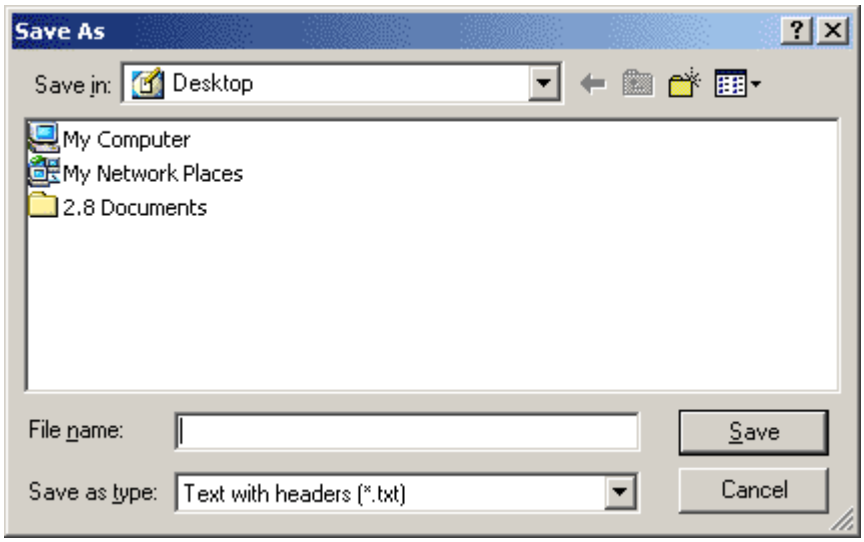


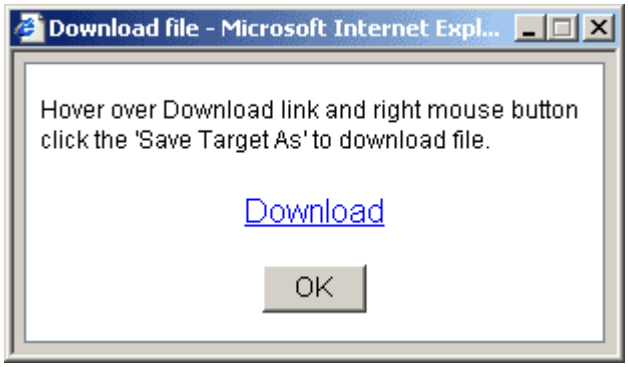
Table 9-2 lists the file formats in which DataWindow data can be saved and how those files can be accessed:

Table 9-2: File formats

File type	File saved as	Open with
Windows Metafile	.wmf	Image viewing software
Text with headers	.txt	Windows Notepad
Excel with headers	.xls	Microsoft Excel
HTML Table	.html	Web Browser

Step 3 – In Pure-JavaScript deployments, a Download File window will be displayed, as shown in Figure 9-24. Right-click on the link to Download and select Save Target As to save the file to the local machine.

Figure 9-24: Download file window in Pure-JavaScript deployment



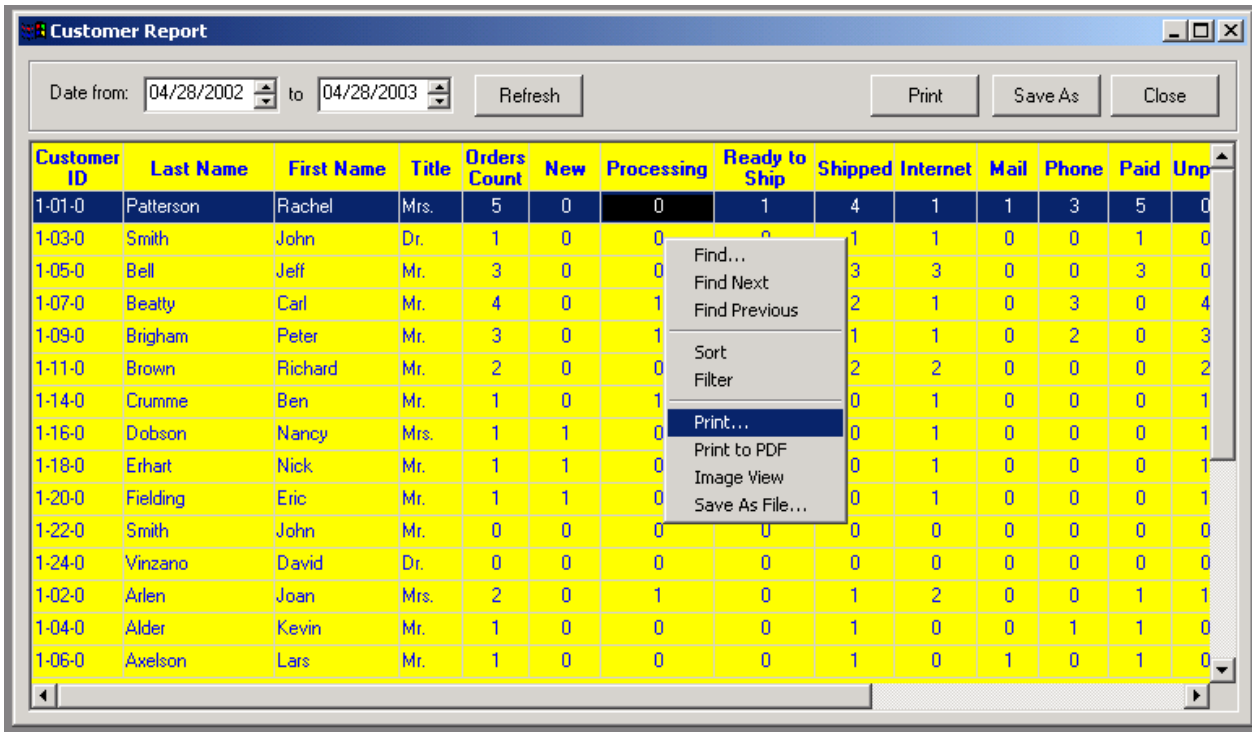
9.4.2.e Enhanced Menus for HTML DataWindows in Appeon Xcelerator deployments

In Appeon Xcelerator deployments, the user has an extra Print menu. With this menu the user can print DataWindows directly on printers connected to the Client.

Enhance Print menu

Step 1 – Select “Print” from the popup menu, as shown in Figure 9-25.

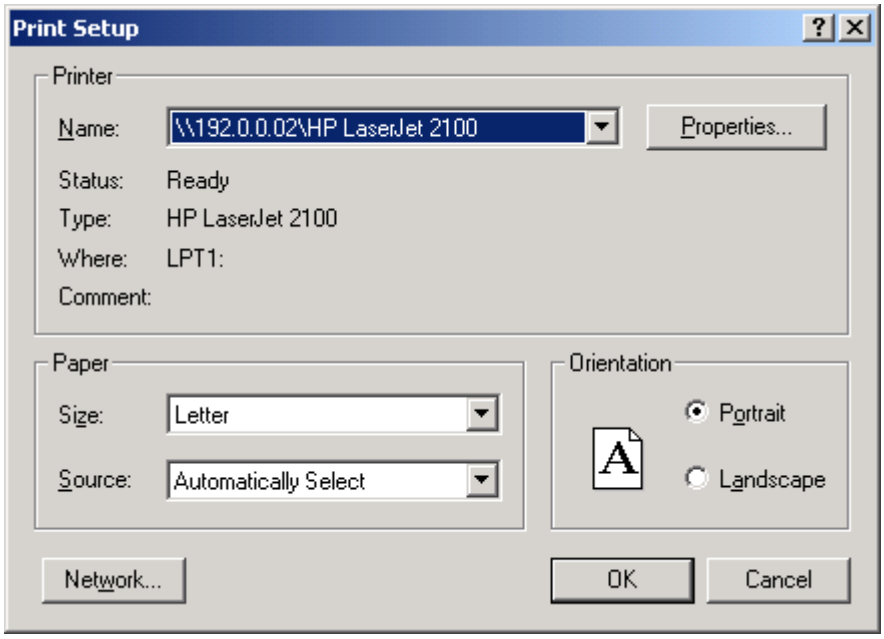
Figure 9-25: Print file with physical printers



Step 2 – Specify the printer settings and paper settings in the Print Setup window, as shown in Figure 9-26. Click *OK*.

The DataWindow will be printed directly to the local printer.

Figure 9-26: Print setup window



Text edition menus

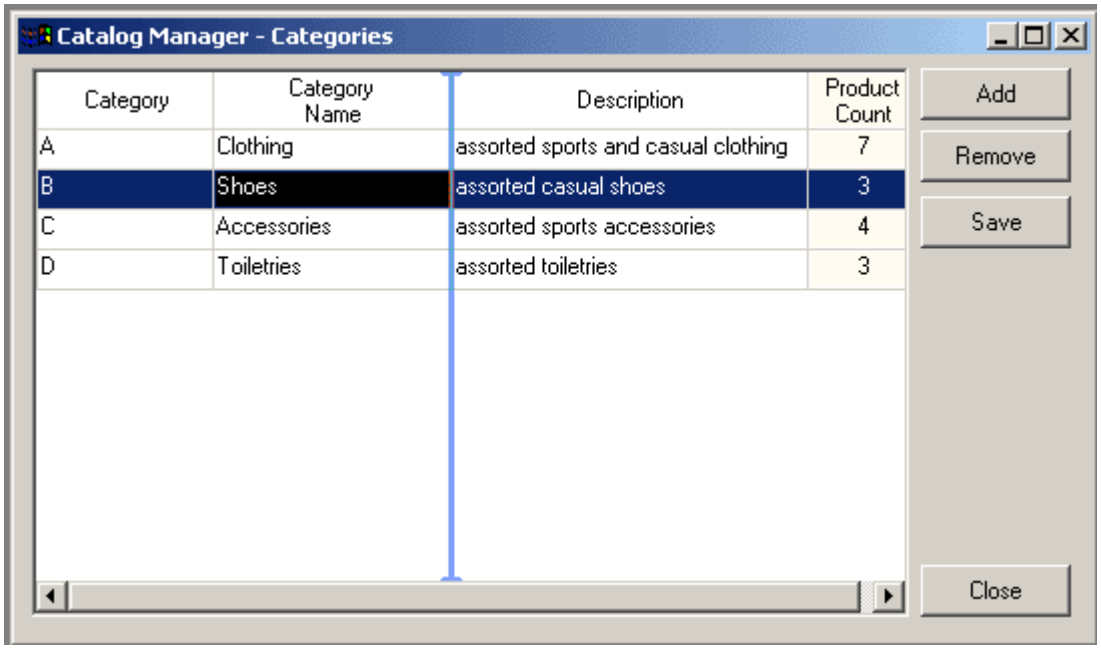
In addition to the popup menu (Find, Find Next, Find Previous, Sort, Filter, Print, Print to PDF, Image View, Save As File, Print), Appeon Xcelerator deployments offer more HTML DataWindow features for users to manage Tabular, Freeform and Grid DataWindows on the Web. Users can place DataWindow columns 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 HTML DataWindow and the selected location is highlighted, as shown in Figure 9-27.

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

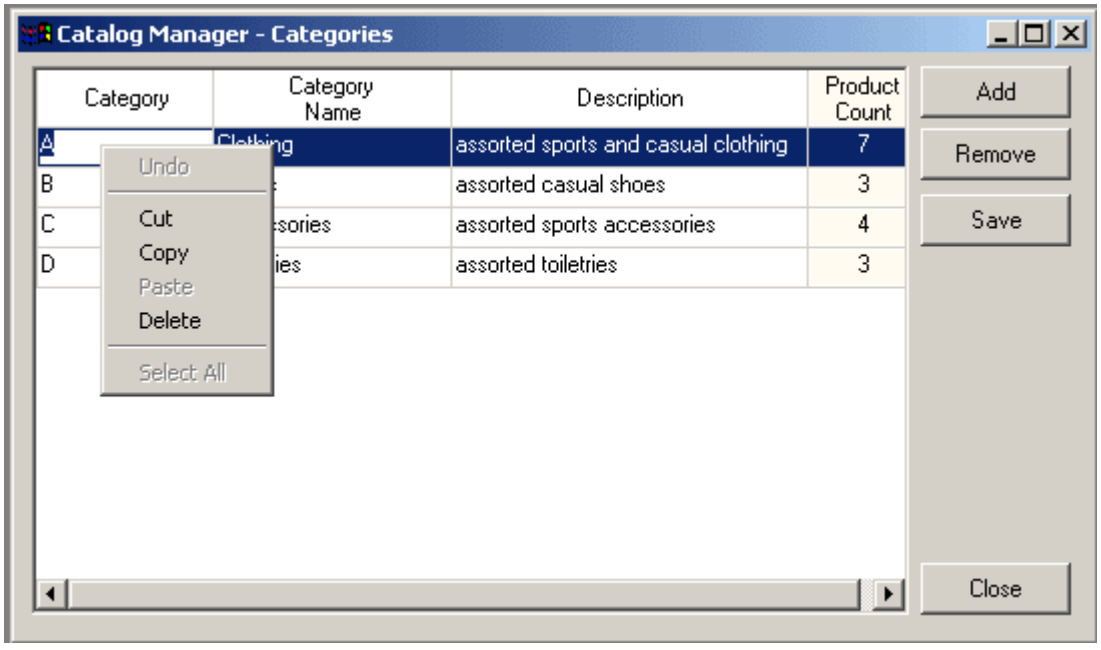
Figure 9-27: Select a column header



Step 3 – Right click on an item in an editable HTML DataWindow and a menu pops up, as shown in Figure 9-28.

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

Figure 9-28: Popup menu for quick edition



9.4.3 Web DataWindow Menu for Image DataWindow

Graph, CrossTab, Label, N-Up and Nested Image DataWindows that are deployed with Appeon Xcelerator deployment option have the following enhanced features: DataWindow printing, SaveAs and DataWindow zooming. Image DataWindows deployed in Pure-JavaScript deployment option do not have these features.

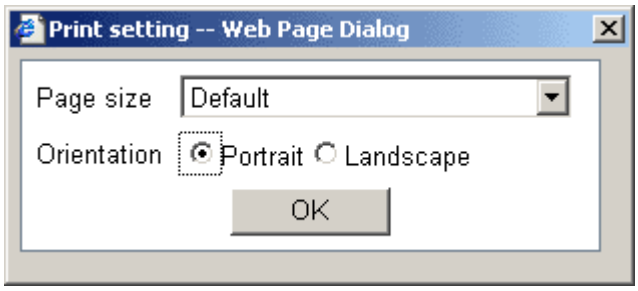
9.4.3.a DataWindow Printing

The user can print supported Image DataWindows in PDF while viewing the application on the Web by performing the following steps:

Step 1 – Right-click on a DataWindow control and select Print to PDF from the popup menu.

Step 2 – Select the desired page size and orientation settings in the Print Setting dialog box, as shown in Figure 9-29.

Figure 9-29: Print Setting



A PDF Repository window is displayed at the bottom right-hand corner of the Internet Explorer window.

You can print several PDF files at the same time. All the printed documents are listed in the order of time; you may click any of them. The PDF files are loaded into a separate Internet Explorer window using the Adobe Acrobat plug-in. You can view the history of all the printed documents at any time; you can also delete them from the history.

For detailed instructions on managing the PDF Repository window, refer to *DataWindow printing* in Section 9.4.2: *Web DataWindow Menu for HTML DataWindows*.

9.4.3.b DataWindow SaveAs

The user can save the content of an Image DataWindow in the following formats: Windows Metafile, text, Microsoft Excel or HTML table format.

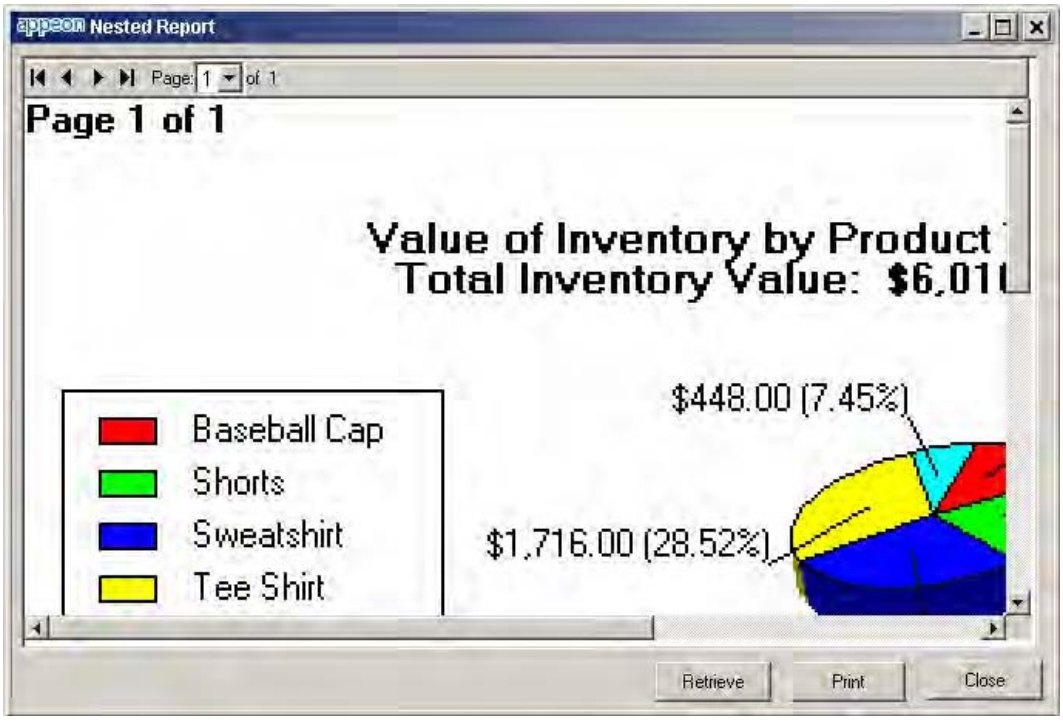
The process to use Save As File features in Image DataWindows is exactly the same as that in HTML DataWindows. For detailed instructions, refer to *DataWindow SaveAs* in Section 9.4.2: *Web DataWindow Menu for HTML DataWindows*.

9.4.3.c DataWindow zooming

While viewing an Image DataWindow, the user can right-click over the DataWindow and select *Zoom in* or *Zoom out*.

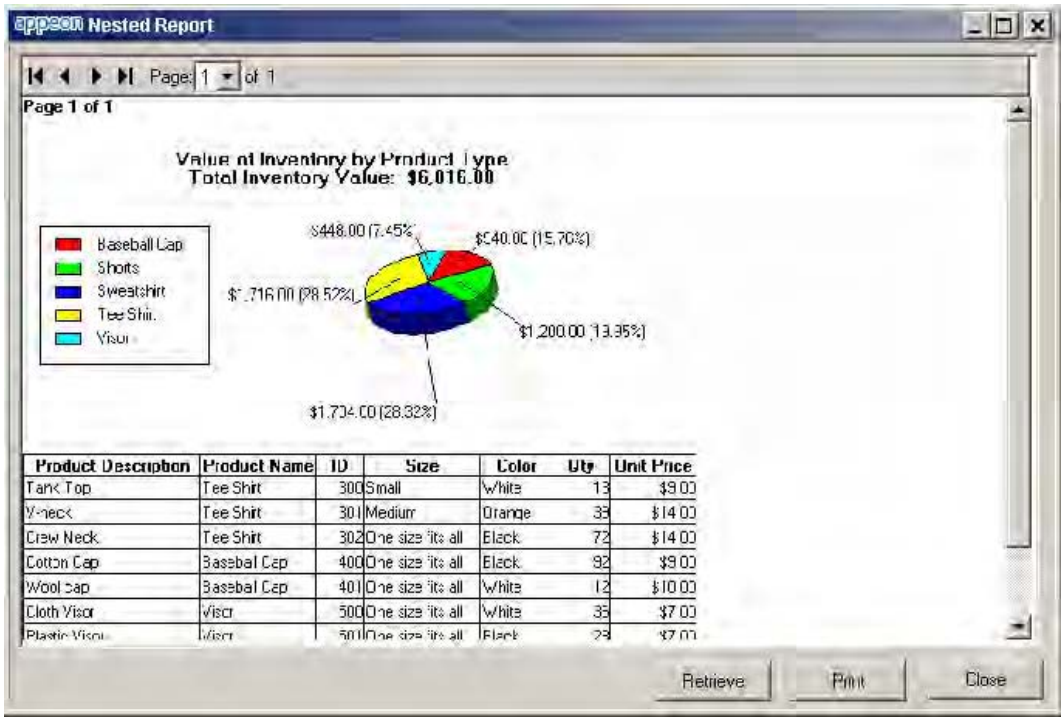
Step 1 – Right-click the DataWindow and select *Zoom in* from the popup menu to zoom in on the DataWindow, as shown in Figure 9-30.

Figure 9-30: DataWindow zoom in



Step 2 – Right-click the DataWindow and select *Zoom out* from the popup menu to zoom out on the DataWindow, as shown in Figure 9-31.

Figure 9-31: DataWindow Zoom out



10 Using Information Manager

10.1 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, Debugging, and Performance. The log files are categorized as follows: Deploy Log, and Analysis Log.

Table 10-1 provides a brief description of all the reports generated during the Web migration process and the report types they are associated with.

Table 10-1: Report Description

Report Type	Description
Analysis	**Analysis Report** 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 5.3.2: *Manipulating the UFA Report in the UFA Report Window*.
Debugging	**Web Debug Report** is generated if the Web application is run, and records the executed objects, controls, and events/functions of the Web application.
Performance	**Web Performance Report** is generated dynamically during the runtime of the Web application, and records the time details of executing the routines. **Heavy Window Report** is generated during a feature analysis or application deployment, and records the complex windows in the application.

Table 10-2 provides a brief description of all the log files generated during the Web migration process, and the log types they are associated with.

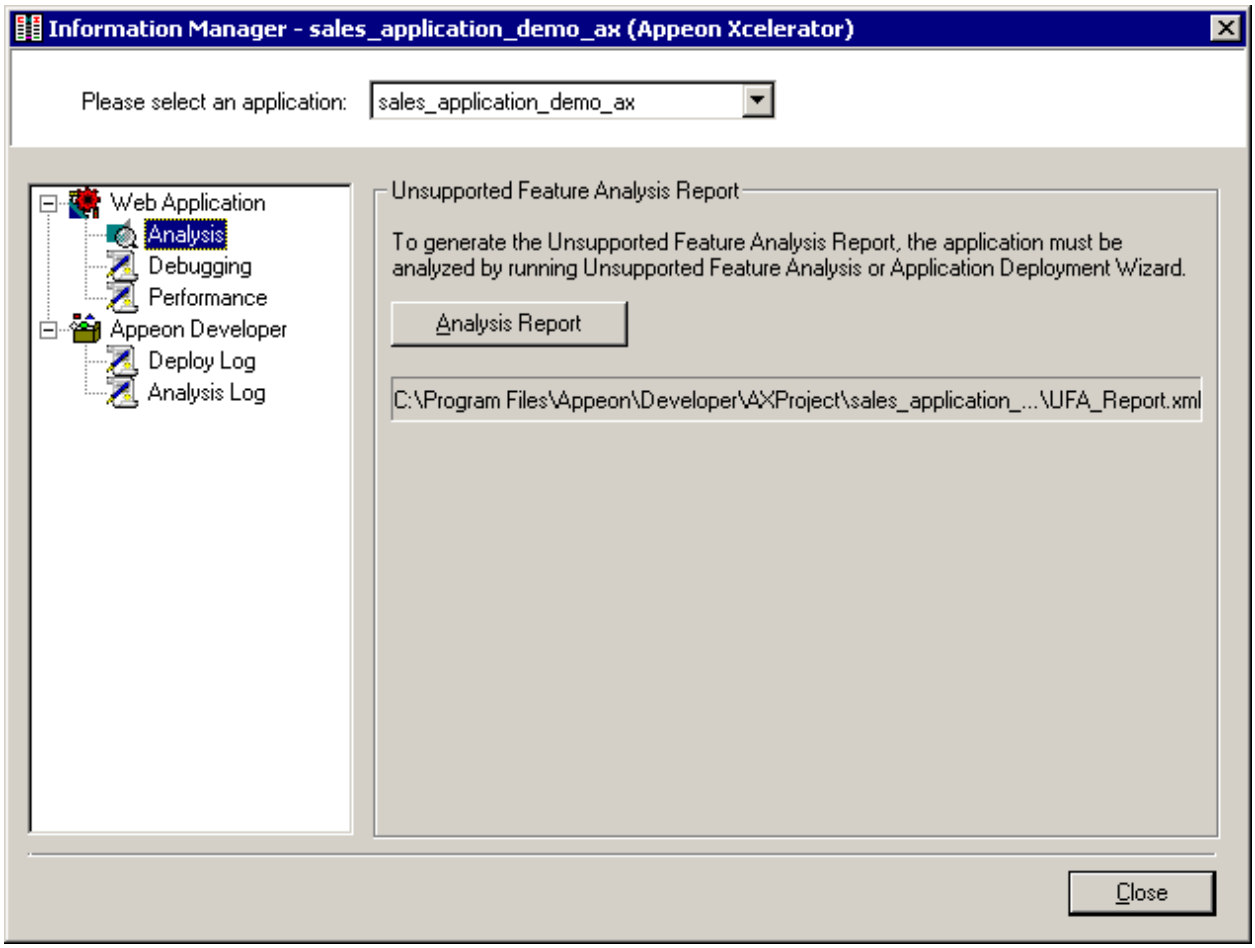
Table 10-2: Log Type Description

Log Type	Description
Deploy Log	The **Deploy Log** set is generated during application deployment, and records the deployment process status. 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.
Analysis Log	The **Analysis Log** set is generated during the feature analysis process, and records the analyzing status.

10.2 Viewing the reports and logs

Step 1 – Click the *Information* button () on the Appeon Developer toolbar and the Information Manager window is displayed, as shown in Figure 10-1.

Figure 10-1: Information Manager



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

The default application configured in Application Profiles and its current deployment option (Pure-JavaScript or Appeon Xcelerator) are displayed in the window title. If you select a different application from the dropdown listbox, the deployment option for the selected application is the current deployment option selected in the Application Profile Configuration page.

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.

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 need to 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.

11 Packaging and Installing Web Applications

11.1 Overview

The application packaging tool 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 instead of doing several individual 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 performing deployment using Appeon Developer in each area, Appeon recommends that you:

1. Deploy the target PowerBuilder application using the Appeon Deployment Wizard (See Chapter 6: *Deploying PowerBuilder Applications*) and
2. Use the packaging tool to pack all the necessary files (See Section 11.2: *Packaging a Web application*), then
3. Copy the installation package to an Appeon Server machine and install the application to any number of target servers located in the same network area. (See Section 11.3: *Installing and uninstalling a Web application*)

11.2 Packaging a Web application

11.2.1 Important requirements

Before using the packaging tool, verify that you have deployed the target PowerBuilder application as required:

- You must have performed a **full deployment** on the target PowerBuilder application on the **same Developer machine** where you will use the packaging tool to pack this application later.

  After the target PowerBuilder completes the first two deployment tasks - Exporting source code (Task 1), and Generating Web files (Task 2) - all the required files are generated at the local Developer machine and are then ready for packaging.

11.2.1.a Additional requirement for packing EAServer components

If the target PowerBuilder application uses any EAServer components, you can also use the packaging tool to pack all the necessary EAServer components and then install them to the local EAServer by using the setup packages instead of deploying them using PowerBuilder. However, be aware of the following requirement:

- All the required EAServer components must have been deployed to a component package that has the **same name** as the target application on the **local EAServer**. The packaging tool can only extract the EAServer components from the local EAServer.

Note: If you do not have EAServer installed on the local Developer machine, you can ignore this requirement, skip the prompt for specifying the local EAServer installation path during the packaging process, and then manually deploy the required EAServer components to the target EAServer using PowerBuilder.

11.2.2 Packaging instructions

Since the package wizard will only extract the files from the hard drive on the local Developer machine, there is no need to run any servers during the packaging process.

To package a Web application, take the following steps:

Step 1 – Click the *Package* () button on the Appeon Developer toolbar.

Step 2 – In the Appeon Web Application Package Wizard (Figure 11-1), select the profile of the application that you want to package.

Figure 11-1: Web Application Package Wizard

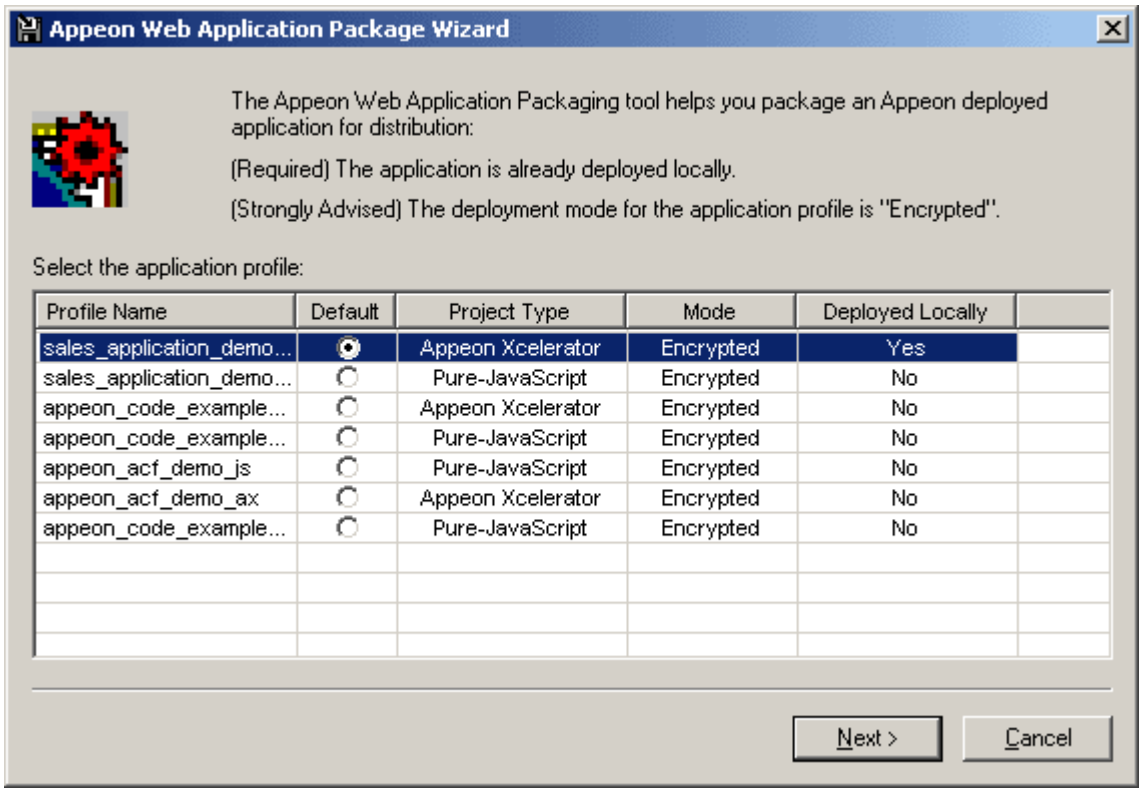


Table 11-1 gives a brief introduction of the columns:

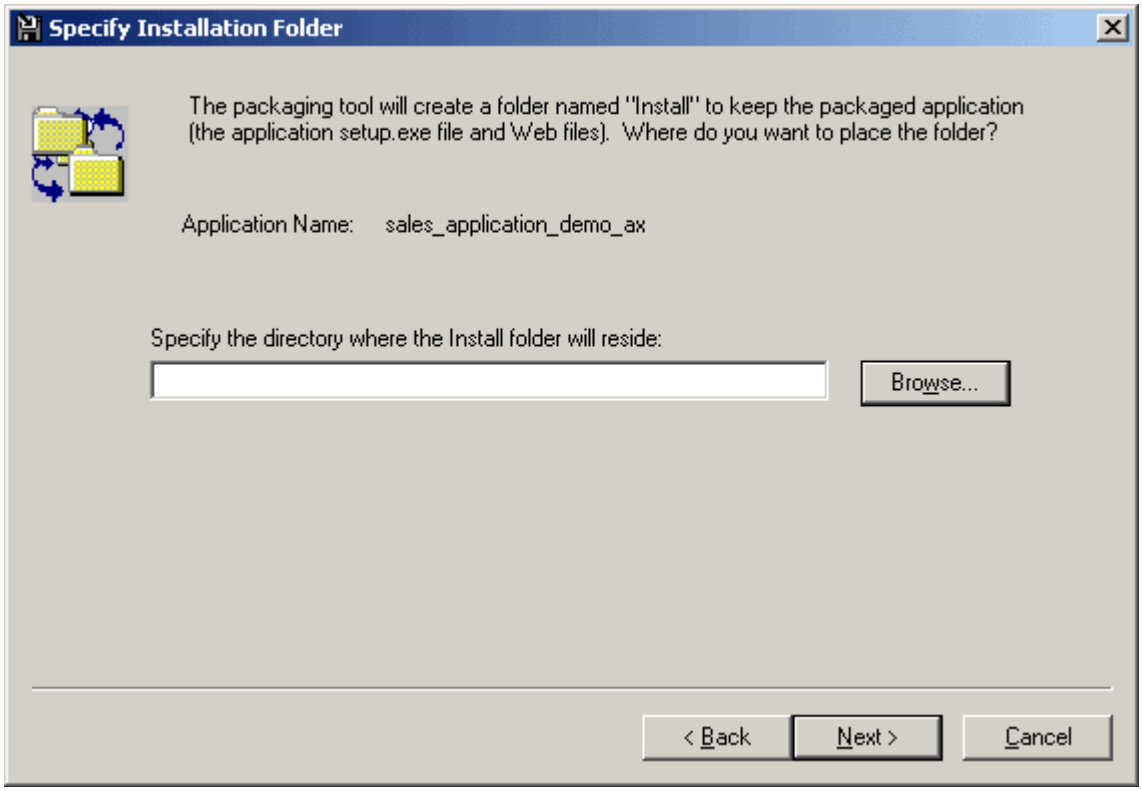
Table 11-1: Web Application Package Wizard

Column	Description
Profile Name	All the application profiles configured in the Appeon Developer Configuration window are listed in the order that the application profiles are created.
Default	Identical to the default application profile selected in the Appeon Developer Configuration window.
Project Type	The current Project Type is set to the deployment option selected in the Application Profile Configuration window.

Mode	Identical to the current mode the application profile is set to in the Appeon Developer Configuration window. Note that as the Web Application Package Wizard indicates, packaged applications should be available in deployed and encrypted mode. This mode prevents most forms of reverse engineering, and like application packaging, it protects the author's intellectual property.
Deployed Locally	Only locally deployed applications can be packaged. Select the profile that is identified with Yes in the Deployed Locally column.

Step 3 – Click *Next* and specify the installation folder, as shown in Figure 11-2.

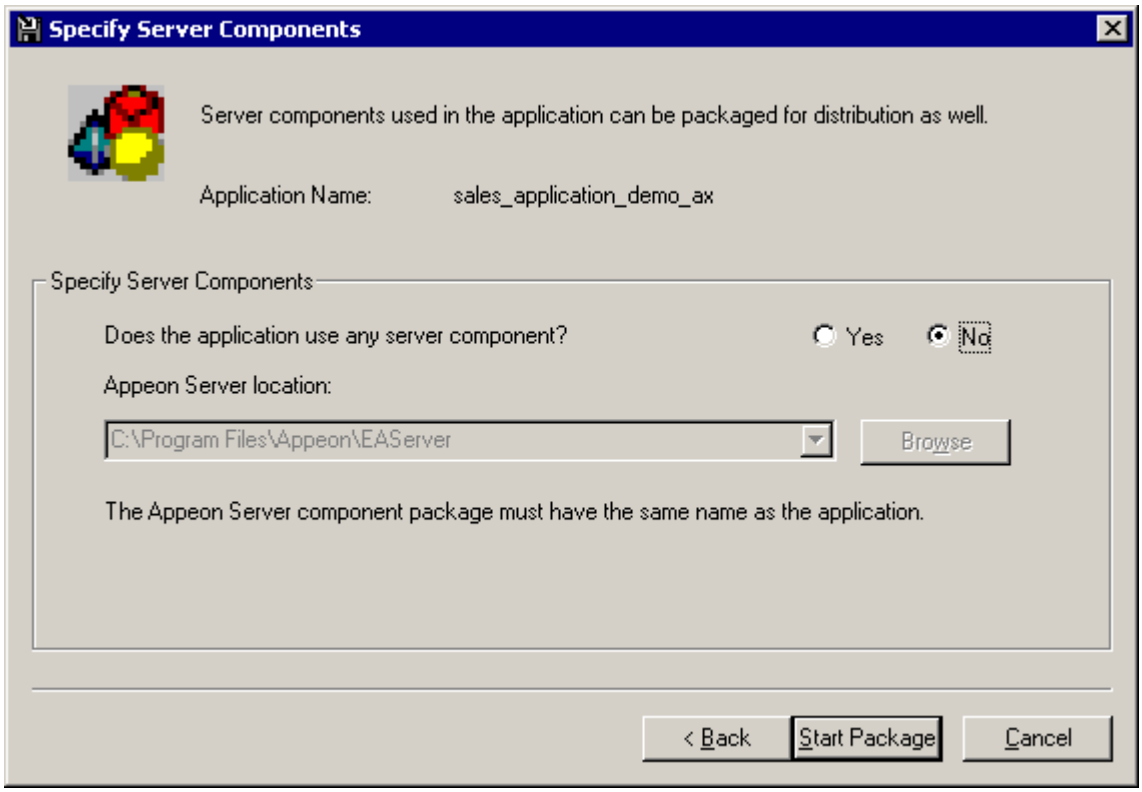
Figure 11-2: Specify installation path



Click *Browse* and select a destination folder for the packaging files.

Step 4 – Click *Next* and specify whether EAServer components are used in the application, as shown in Figure 11-3.

Figure 11-3: Specify server components



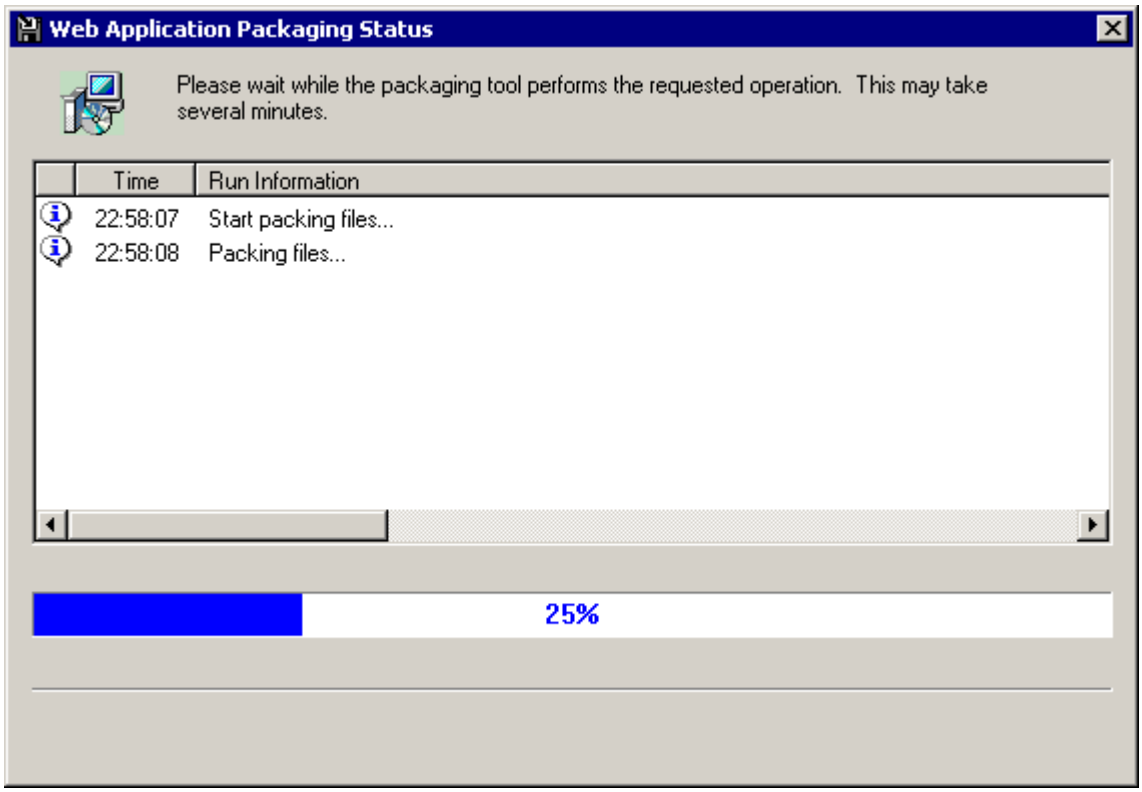
At the click of the *Yes* button, specify the EAServer installation path on the local computer.

To ensure that you can successfully package and install EAServer components, be sure to read the requirement in the *Additional requirement for packing EAServer components* section.

When the Web application is installed, the component files will be installed to the EAServer installation path you specified during the installation.

Step 5 – Click *Start Package* and the Web Application Packaging Status dialog box is displayed, as shown in Figure 11-4.

Figure 11-4: Web application packaging status window



Step 6 – Once the operation is complete, click *Finish* to exit the package wizard.

After the application is successfully packaged, an *install* folder containing all the necessary application files will be generated in the specified destination folder.

You can use the package to install the application to any Appeon Servers and Web Servers.

11.2.2.a What a package contains

The *install* folder for a Web application contains the following items:

- A **mandatory** setup.exe file to set up the Web application, and some DLL files.
- A **mandatory** Config.xml file which contains the default settings for the application, Appeon Server and Web Server.
- A **mandatory** uninstall.exe file to be copied to the local disk of Appeon Server and later used to uninstall the Web application.
- Note: Never try to use the uninstall.exe file in a package to uninstall an application. You should always use the uninstall.exe file that is copied to the local machine to uninstall an application.
- The **mandatory** Web application files.
- The **mandatory** encrypted DataWindow definition files.
- Other **mandatory** application configuration files, for example, the INI file.
- Optional image files.
- Optional EAServer components extracted from the local EAServer installation path.

The package does not contain:

- The AEM settings for the application.
- Connection cache settings in the server for the application.

You need to manually configure the AEM settings and connection cache settings for each Appeon Server.

11.3 Installing and uninstalling a Web application

11.3.1 Points to note before installation

Be aware of the following points before you install a Web application:

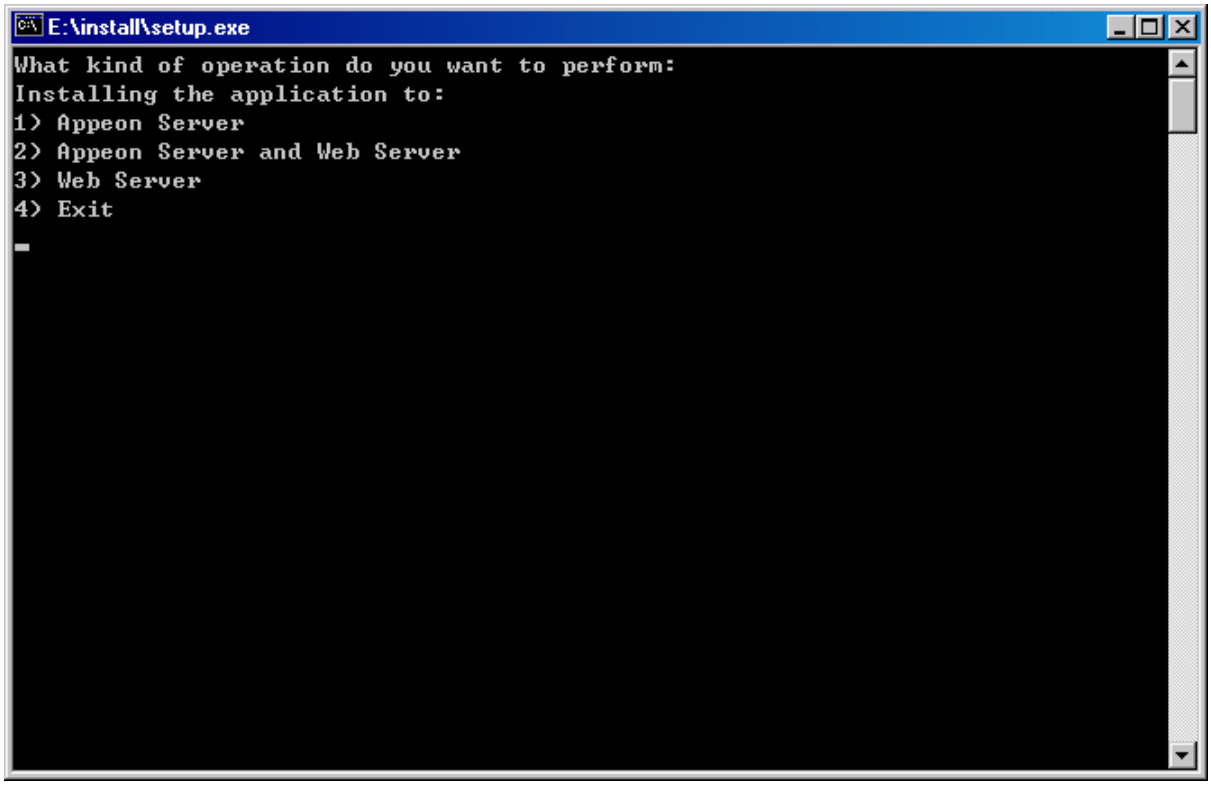
- You can install the application to one Appeon Server (either local or remote) at a time, though you can install to several Web Servers simultaneously.
- If an Appeon Server cluster is in use, the Setup program must be run at each individual Appeon Server. After the installation, configure the servers and set them as a server cluster.
- The target Web servers must have the Appeon Server Web Component correctly installed.
- The packaged EAServer components can only be installed to the local EAServer.

11.3.2 Installation instructions

Step 1 – Start the target Appeon Server and Web server(s) where you want to install the application.

Step 2 – Copy the *install* folder to any machine and then run the Setup.exe file. The installation prompt line is displayed in a command box, as shown in Figure 11-5.

Figure 11-5: Application installation



Step 3 – Select one installation type. There are three installation types.

- Press "1" to install only the DataWindow syntax files, INI files and EAServer components to a local or remote Appeon Server. EAServer components can only be installed to the local EAServer.
- Press "2" to install the whole application to the Appeon Server (either local or remote) and one or several Web servers.
- Press "3" to install only the Web files to one or several Web servers.

Step 4 – Choose whether to use the default settings for the application, Appeon Server and Web Server. The default settings are saved in the Config.xml file in the *install* folder. You can view or modify it.

The default settings were the settings used to deploy this application within the Appeon Deployment Wizard. Refer to Chapter 4: *Configuring Appeon Developer* for detailed instructions on how to configure the required settings:

Table 11-2: Installation settings

Installation Setting	Installation Type	Description
Path for the uninstall program	Type 1, 2 & 3	Input a valid path on the local machine for storing the application uninstall program.
Appeon Server settings: • IP/machine name • Port • Deployment username • Deployment password	Type 1, 2	Refer to Table 4-11 for detailed description.
EAServer installation path	Type 1, 2	If the Setup program detects that the package contains EAServer components and that the specified Appeon Server is a local Appeon Server, you will be prompted to type the full path to the local EAServer (for example: C:\Program Files\Sybase\EAServer). The EAServer components will be installed to the specified EAServer.
Web root folder	Type 2, 3	Refer to the *Specify the Web deployment path* section for detailed description.
Web Server type	Type 2, 3	Refer to Table 4-12 for detailed description.
File transfer type: • Local File Transfer, or • FTP File Transfer	Type 2, 3	Refer to Table 4-12 for detailed description.
Local Web root path	Type 2, 3	Refer to Table 4-12 for detailed description. The local Web root path is required only when you select the Local File Transfer type.

FTP settings: • IP/machine name • Port • user name • password	Type 2, 3	Refer to Table 4-12 for detailed description. The FTP settings are required only when you select the FTP File Transfer type.

If you enter "Y" to use the default settings, the Setup program tests the default settings in the Config.xml file. If the test succeeds, the installation process starts.

If you enter "N", you are prompted to enter the required settings according to the selected installation type.

Step 5 – (For installation type 2 or 3) Choose whether to configure another Web Server. You can install an application to several Web Servers simultaneously. If you choose not to configure another Web Server, the Setup program proceeds to install the Web application to the configured Appeon Server and the Web Server(s).

Step 6 – The Setup program closes the command prompt window immediately after the installation completes.

- The DataWindow syntax files and INI files are installed to the *appeondb* database on the configured Appeon Server.
- (Optional) The EAServer components are installed to the specified EAServer on the local machine.
- The Web application files are installed to the Web root path of the configured Web Server(s).

After a Web application is installed, an *appeon_uninstall* folder and an application sub-folder which contains the uninstall program for the application will be created under the specified path on the local computer.

11.3.2.a Troubleshoot using installation log file

If you encounter problems when installing a Web application, use the *appeon_install.log* file for troubleshooting. You can find this file in either of the following locations:

1. The corresponding application sub-folder under the *appeon_uninstall* folder; or
2. The Windows system temporary folder (typically C:\Documents and Settings\Administrator\Local Settings\Temp).

By default, the *appeon_install.log* file for each application is created under the corresponding application sub-folder, however, if the *appeon_uninstall* folder or the application sub-folder cannot be created during the installation process, the log file is generated in the system temporary folder.

11.3.3 Uninstallation instructions

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

Step 2 – Run the *uninstall.exe* file in the application sub-folder under the *appeon_uninstall* folder. The uninstall prompt line is displayed in a command box.

Step 3 – Confirm whether to uninstall the application. Enter "Y" to proceed.

Step 4 – The uninstall program removes all the files from the associated server(s) according to the settings in the *Uninstall.xml* file.

The *Uninstall.xml* file records the associated server(s) and their settings that were used in the installation.

Step 5 – The uninstall program closes the command box immediately after the uninstall is complete.

Step 6 – After the application has been uninstalled, restart the machine where the uninstall program was run. If you do not want to restart the machine, remove the application sub-folder under the *appeon_uninstall* folder.

12 Undeploying Web Applications

12.1 Overview

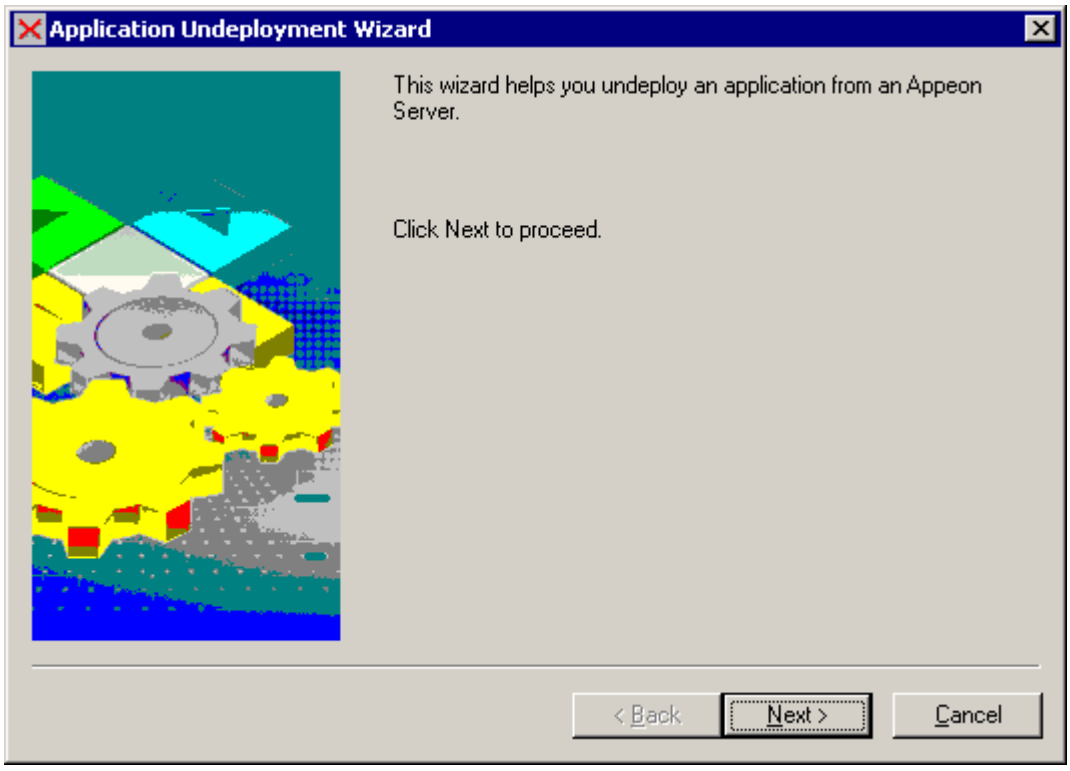
The Application Undeployment Wizard removes a deployed application from the associated Appeon Server(s) and Web Server(s), including all DataWindows from the Appeon Server(s), all Web files from the Web Server(s), and all Transaction Object mappings from AEM. This is the only way that an Appeon deployed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

12.2 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 12-1.

Figure 12-1: Application Undeployment Wizard



Step 3 – Click *Next* to continue. The Specify Undeployment Mode window is displayed, as shown in Figure 12-2, prompting you to select an undeployment mode.

Figure 12-2: Specify Undeployment Mode window

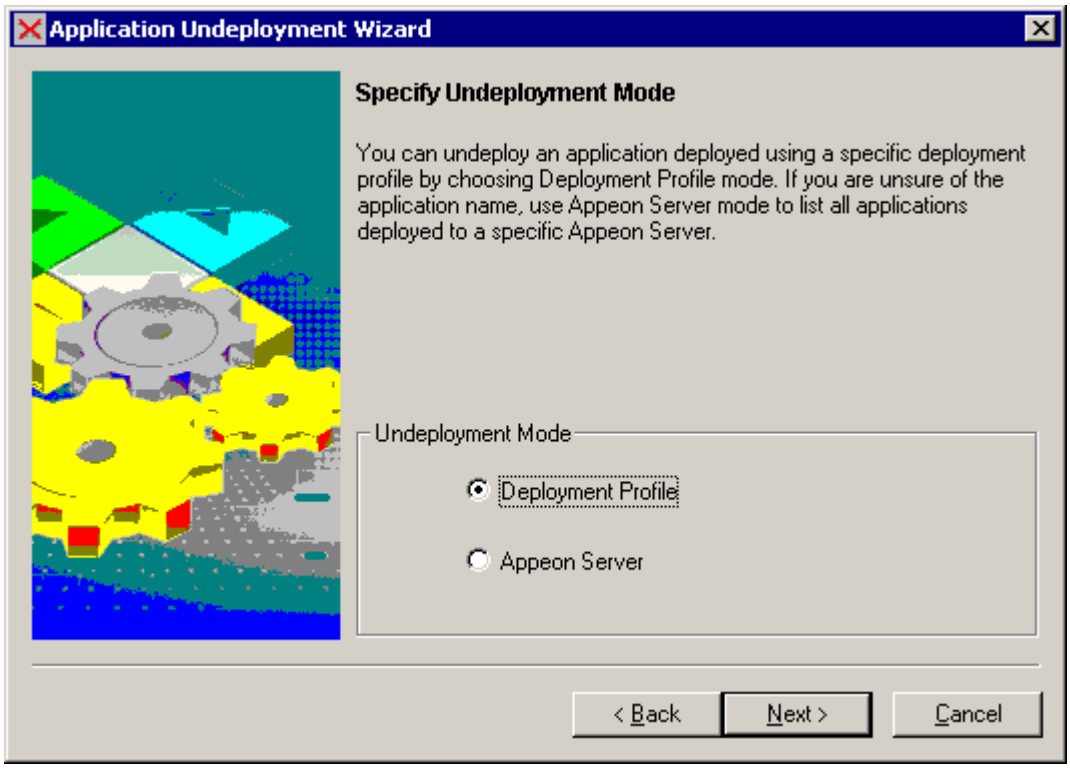


Table 12-1 shows when and how to use each mode.

Table 12-1: Undeployment modes

	When To Use It	How To Use It
Deployment Profile mode	If you clearly know which application you want to delete from which server.	Select an application to be undeployed and its associated deployment profile.
Appeon Server mode	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.	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.

12.2.1 Undeploying with the Deployment Profile mode

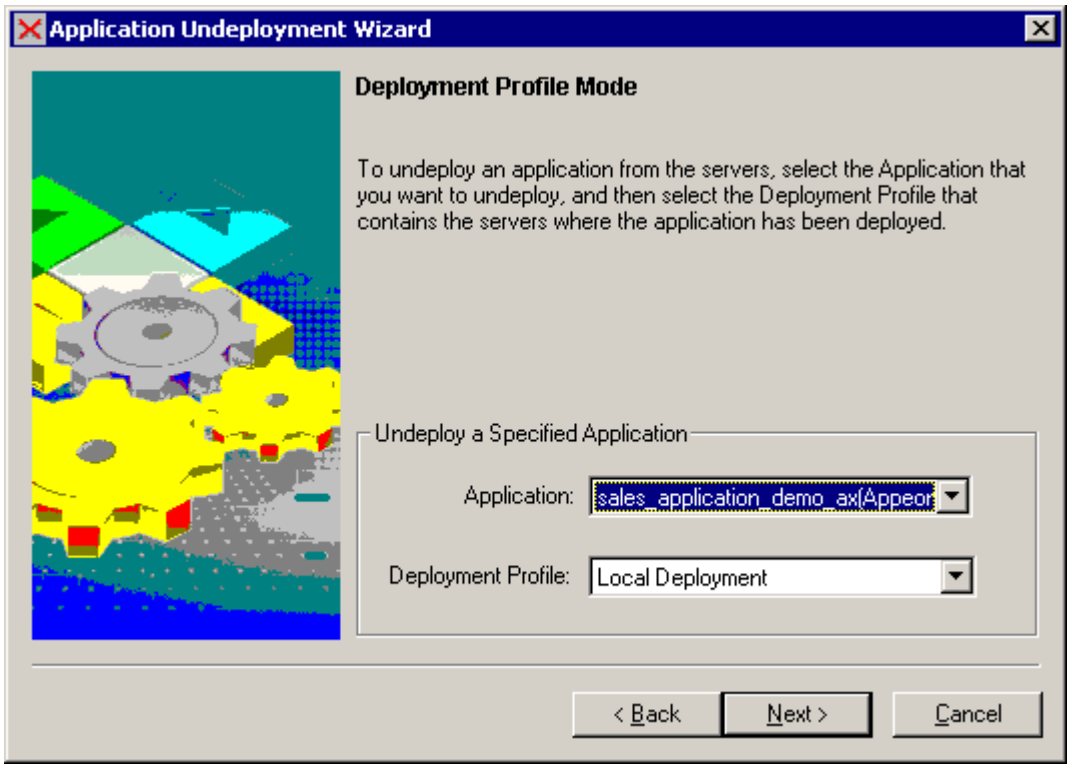
Step 1 – Select the *Deployment Profile* radio button in the Specify Undeployment Mode window as shown in Figure 12-2, and click *Next* to continue.

Step 2 – Select a Web application from the Application list box, as shown in Figure 12-3.

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.

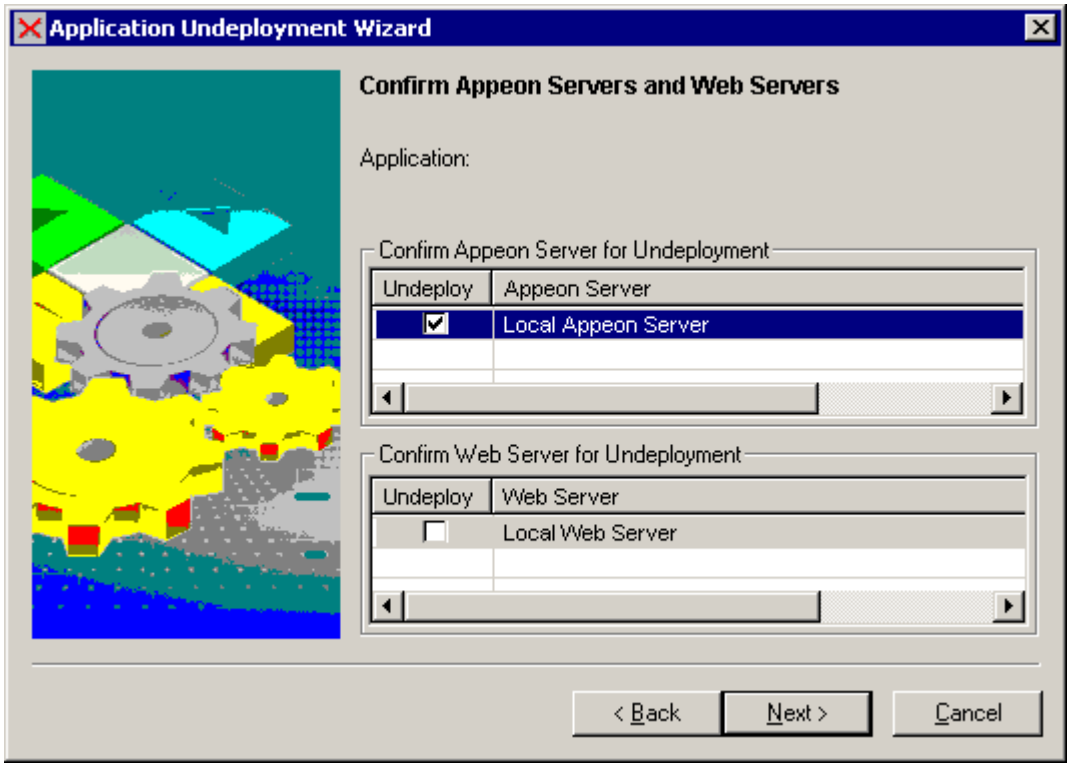
Step 3 – Select the deployment profile that is used to deploy the Web application from the Deployment Profile list box, as shown in Figure 12-3. Click *Next* to continue.

Figure 12-3: Deployment Profile Mode window



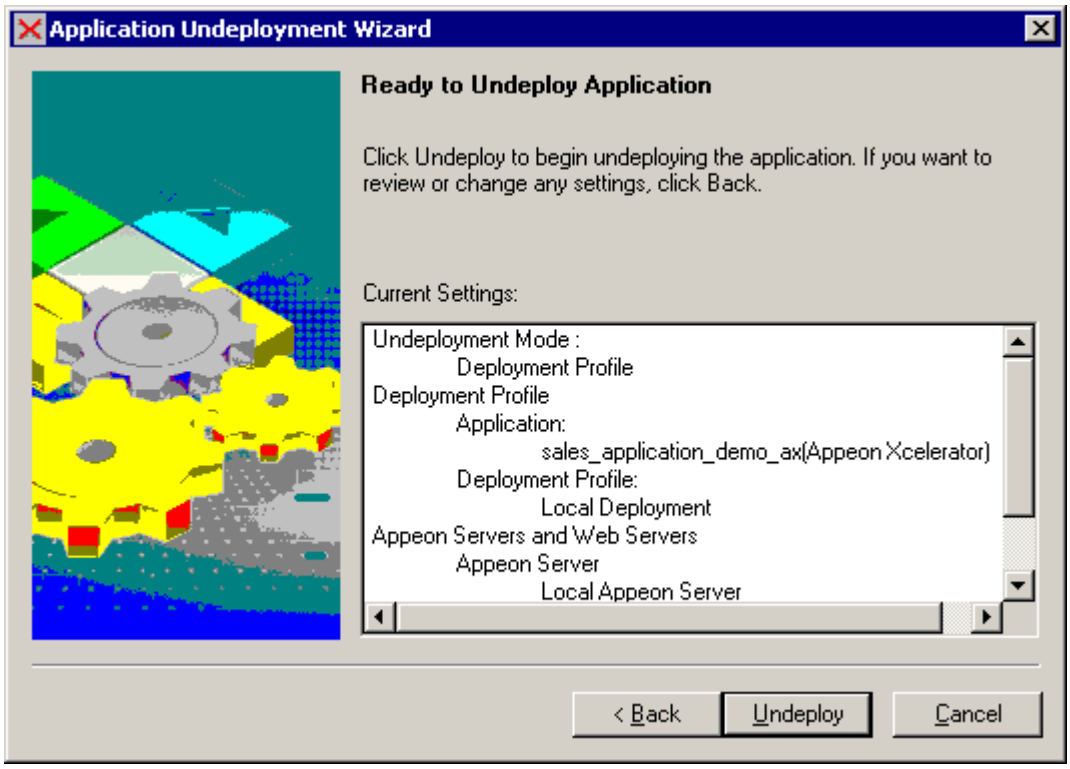
Step 4 – Select the Appeon Server(s) or Web Server(s) that are intended for undeployment, as shown in Figure 12-4. The Appeon Server(s) and Web Server(s) defined in the deployment profile selected in the previous step are listed. Click *Next* to continue.

Figure 12-4: Confirm Appeon Servers and Web Servers



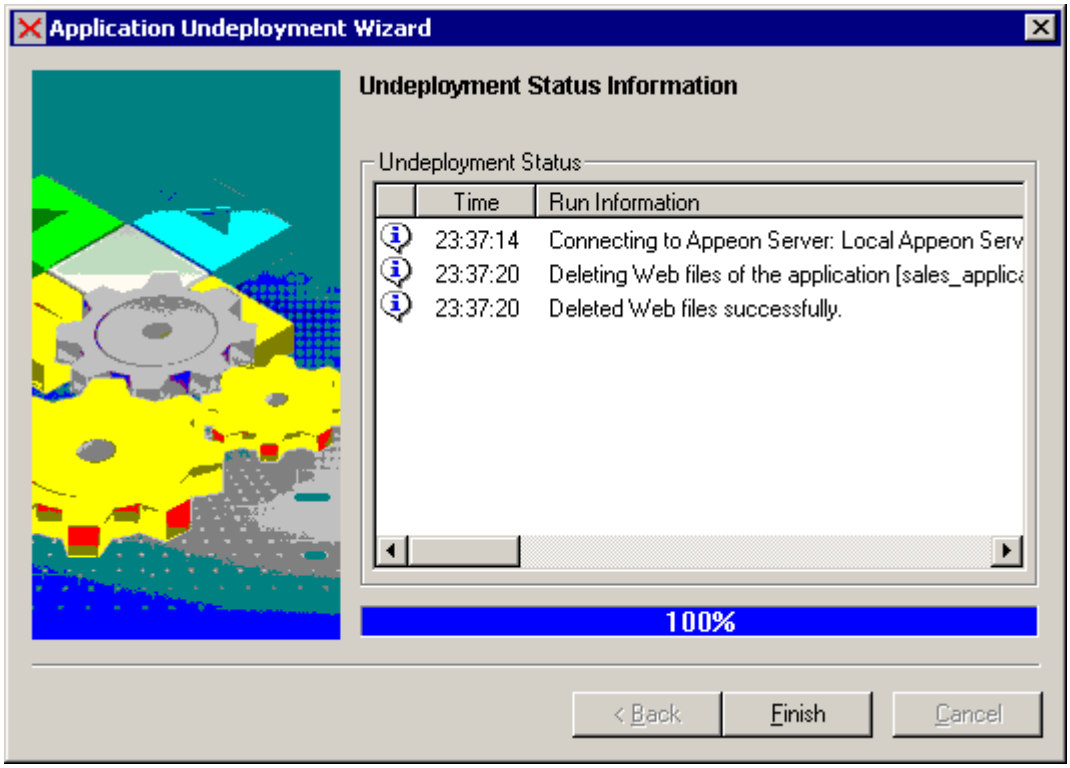
Step 5 – Click *Undeploy* to confirm the undeployment settings and start the undeployment process, as shown in Figure 12-5.

Figure 12-5: Confirm undeployment settings



The undeployment process begins, as shown in Figure 12-6.

Figure 12-6: Undeployment in Progress

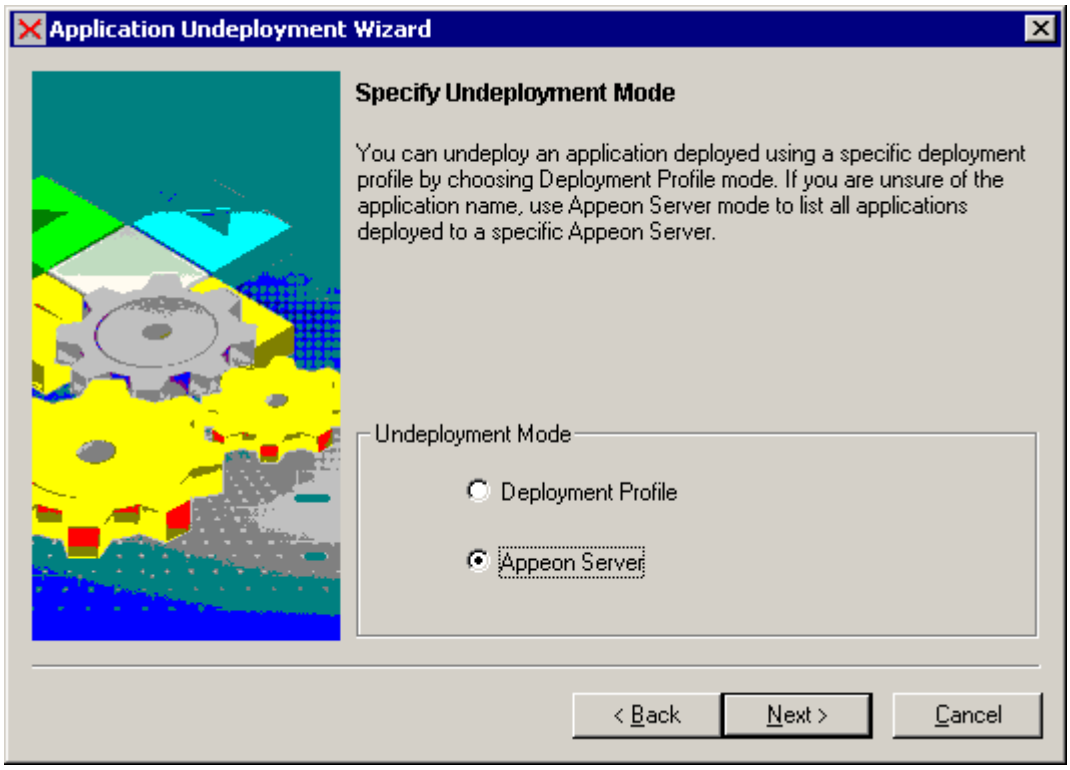


Step 6 – Click *Finish* to close the dialog box, as shown in Figure 12-6.

12.2.2 Undeploying with the Appeon Server mode

Step 1 – Select the *Appeon Server* radio button in the Specify Undeployment Mode window as shown in Figure 12-7, and click *Next* to continue.

Figure 12-7: Specify Undeployment Mode window



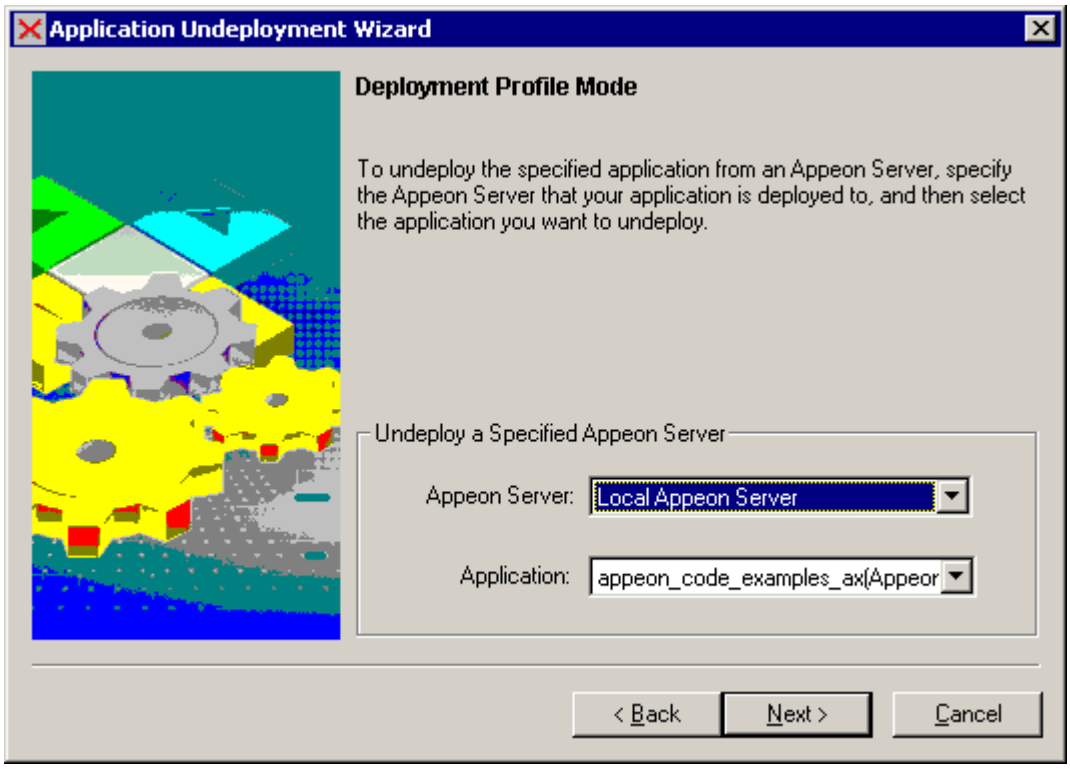
Step 2 – Select the Appeon Server that contains the Web application to be undeployed from the Appeon Server list box, as shown in Figure 12-8.

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 12-8.

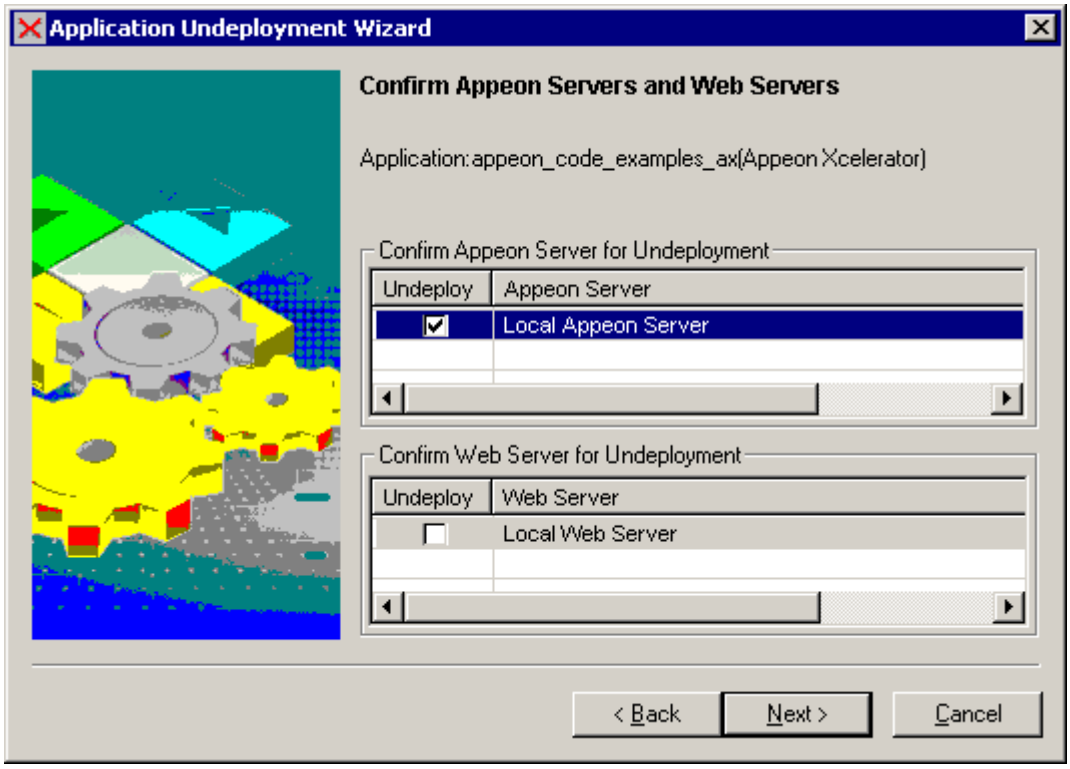
The Application list box lists all the Web applications deployed to the selected Appeon Server.

Figure 12-8: Appeon Server Mode window



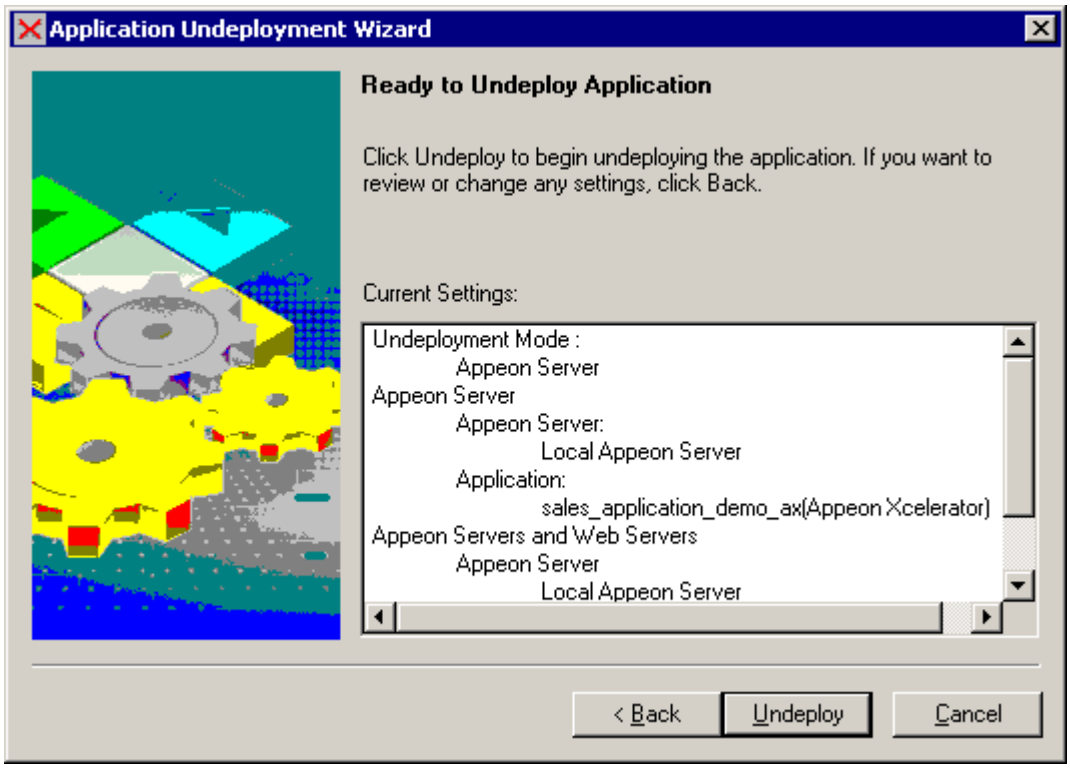
Step 4 – Select the Appeon Server(s) or Web Server(s) that are intended for undeployment, as shown in Figure 12-9. The Appeon Server(s) and Web Server(s) defined in the default deployment profile are listed. Click *Next* to continue.

Figure 12-9: Ready to Undeploy Application window



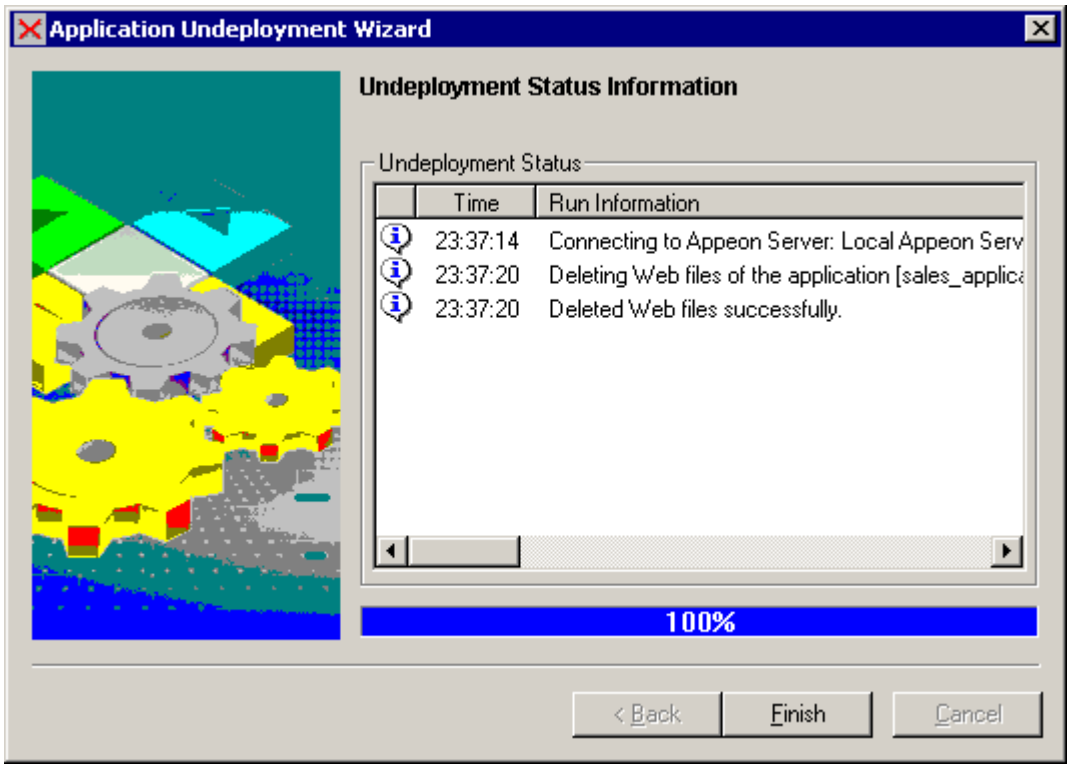
Step 5 – Click *Undeploy* to confirm the undeployment settings and start the undeployment process, as shown in Figure 12-10.

Figure 12-10: Undeployment settings



The undeployment process begins, as shown in Figure 12-11.

Figure 12-11: Appeon Web Application Undeployment



Step 6 – Click *Finish* to close the dialog box, as shown in Figure 12-11.

13 Developing with Code Insight

13.1 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 13-1 describes the differences between the PowerBuilder AutoScript and Appeon Code Insight.

Table 13-1: Comparison between AutoScript and Code Insight

	PowerBuilder AutoScript	Appeon Code Insight
Effective for	Functions, events, variables, properties, and templates for PowerBuilder DO, FOR, IF, and CHOOSE statements.	Items referred in dot notations
What pops up	A list of properties, variables, methods, or statements.	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.
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.
Priority	When Code Insight is activated, AutoScript will be automatically turned off.	

13.2 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 13-2: Status of Code Insight

Icon	Status	Description
	Disabled	Code Insight is disabled.
	Loading	Code Insight is analyzing the default PBT and generating the supported and unsupported feature list based on the analysis result.
	Off	Code Insight is enabled but ineffective.

	On	Code Insight is enabled and effective in the current application.
	Gray	This icon appears anytime the PowerBuilder IDE loses focus, even if Code Insight is disabled.

If you are using more than one PowerBuilder IDE, you should launch Code Insight in each PowerBuilder IDE.

13.2.1 Configuring Code Insight

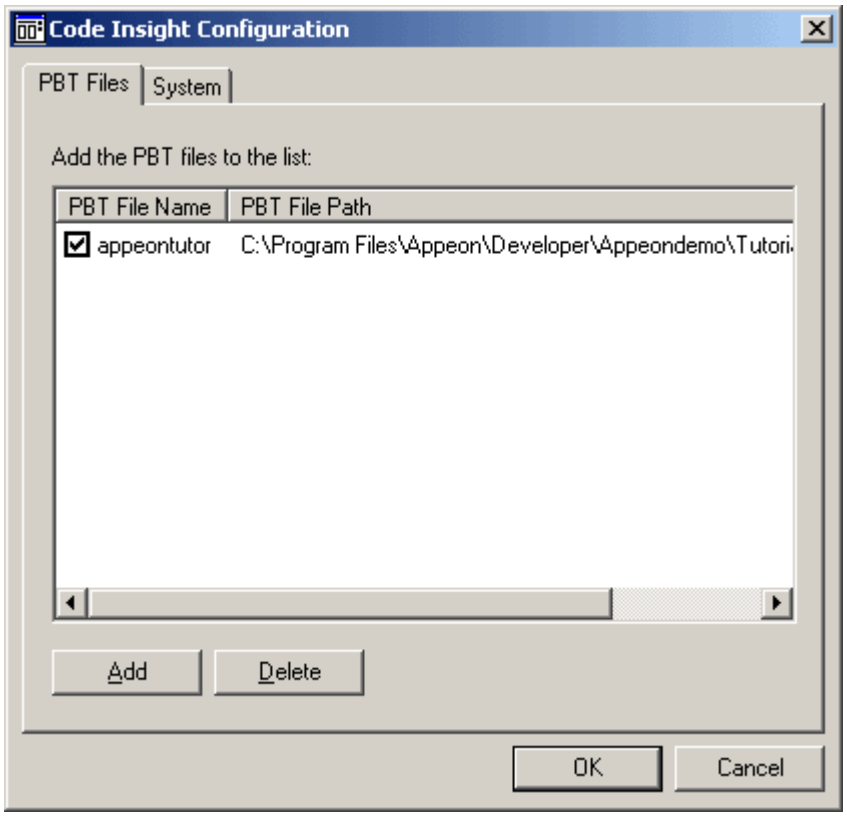
Before you can enable Code Insight for a PowerBuilder application, you need to configure the PBT file of the desired application as the default PBT file. You can also specify the Appeon feature sets and the feature types for which feature list will be displayed.

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 13-1.

Figure 13-1: The status-area-icon menu

Step 2 – Click *Config* to open the Code Insight Configuration window, as shown in Figure 13-2.

Figure 13-2: Code Insight Configuration window



The Code Insight Configuration window contains two tab pages where you can manage the PBT files and specify what displays in the list.

Table 13-3: Code Insight Configuration window

Select this tab	To do this	Detailed Steps
PBT File tab page	Add the PBT files.	1) Click *Add*. 2) Search and select the PBT file in the common dialog box. 3) Click *Open* in the common dialog box. 4) Click *OK*.
	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	Select a deployment option.	1) Select the deployment option to be used to deploy the application. The two deployment options, Appeon Xcelerator and Pure-JavaScript, support slightly different feature sets. The selected deployment option determines the feature set to be used in developing the application using Code Insight. Make sure you will deploy the application with the same deployment option selected in Code Insight.
	Specify what displays 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*.

13.2.2 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 13-3.

Figure 13-3: Enable Code Insight

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

Table 13-4: Enable Code Insight

Status	Detailed Steps
	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 ().
	Code Insight is **enabled and effective**. You can develop an application with the assistance of Code Insight.

13.3 Coding with Code Insight

When Code Insight is enabled and effective () 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.

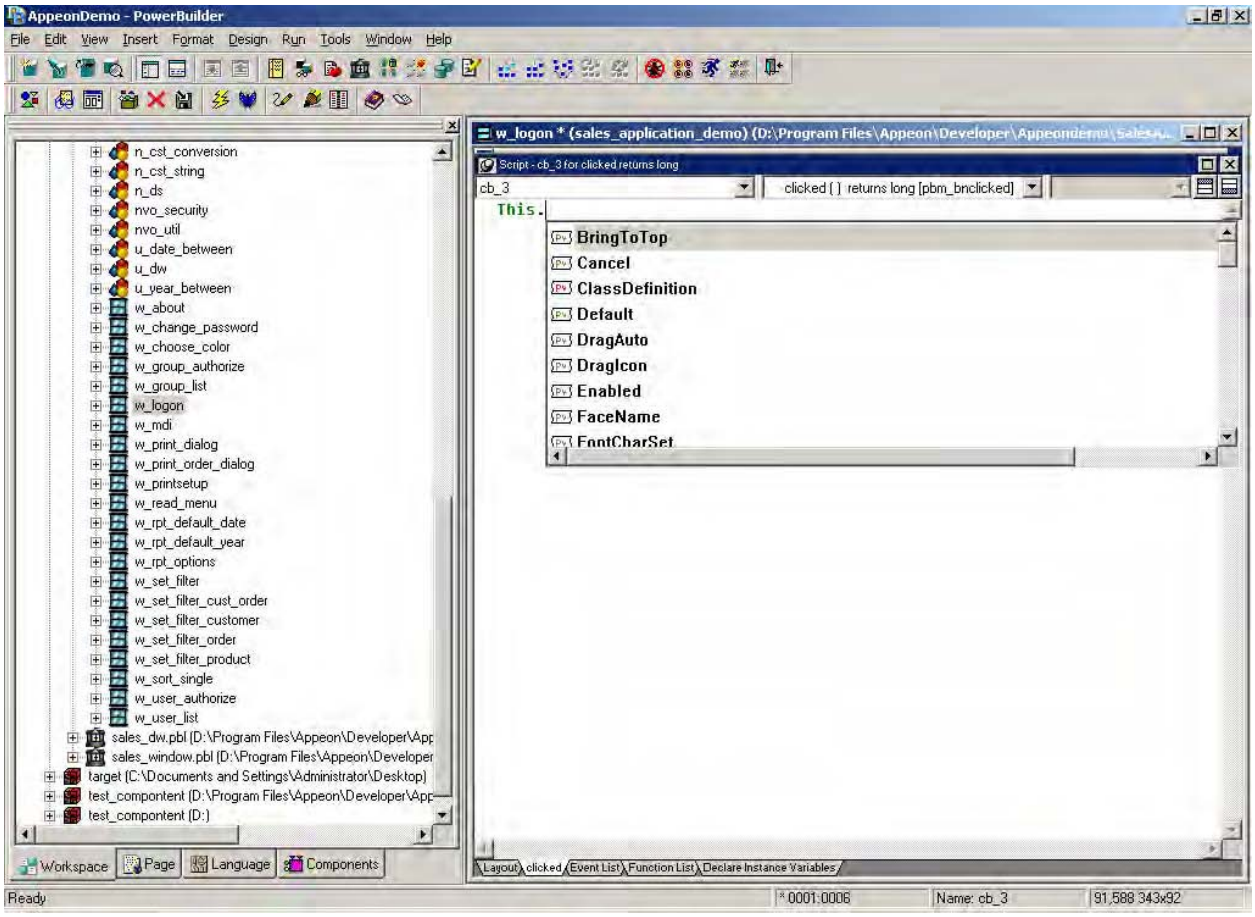
Step 2 – Save the object by clicking the *Save* button on the PowerBar.

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.").

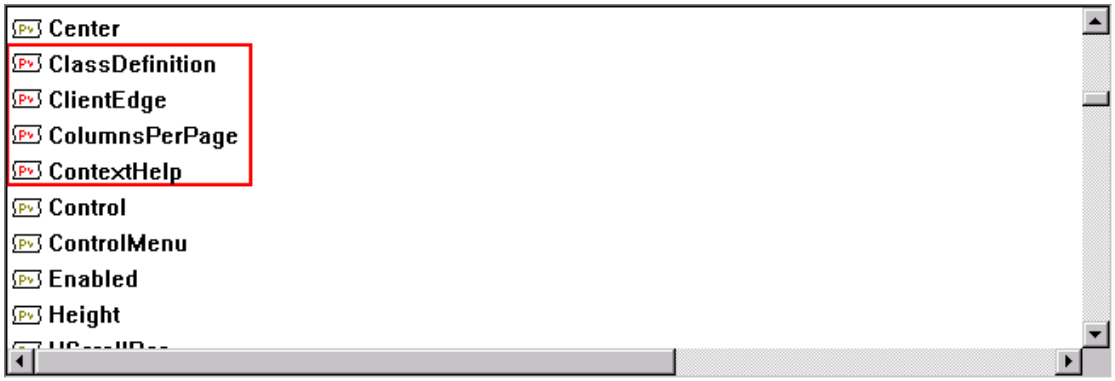
If there is more than one property, variable, method, or statement that could be inserted, the Code Insight feature list pops up for you to select a supported property, variable or method, as shown in Figure 13-4.

Figure 13-4: 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 13-5. Avoid using the unsupported features.

Figure 13-5: 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.

14 Appeon for PowerBuilder Help

Appeon provides a comprehensive HTML help system that contains:

- Working with Appeon Developer, the HTML version of the *Appeon Developer User Guide*
- Appeon Features Help for Appeon Xcelerator, the HTML version of the *Supported Features Guide* for Appeon Xcelerator Deployment
- Appeon Features Help for Pure-JavaScript, the HTML version of the *Supported Features Guide* for Pure-JavaScript Deployment
- Appeon-compliant Framework Reference, the HTML version of the *Using the PowerBuilder Foundation Class with Appeon.*

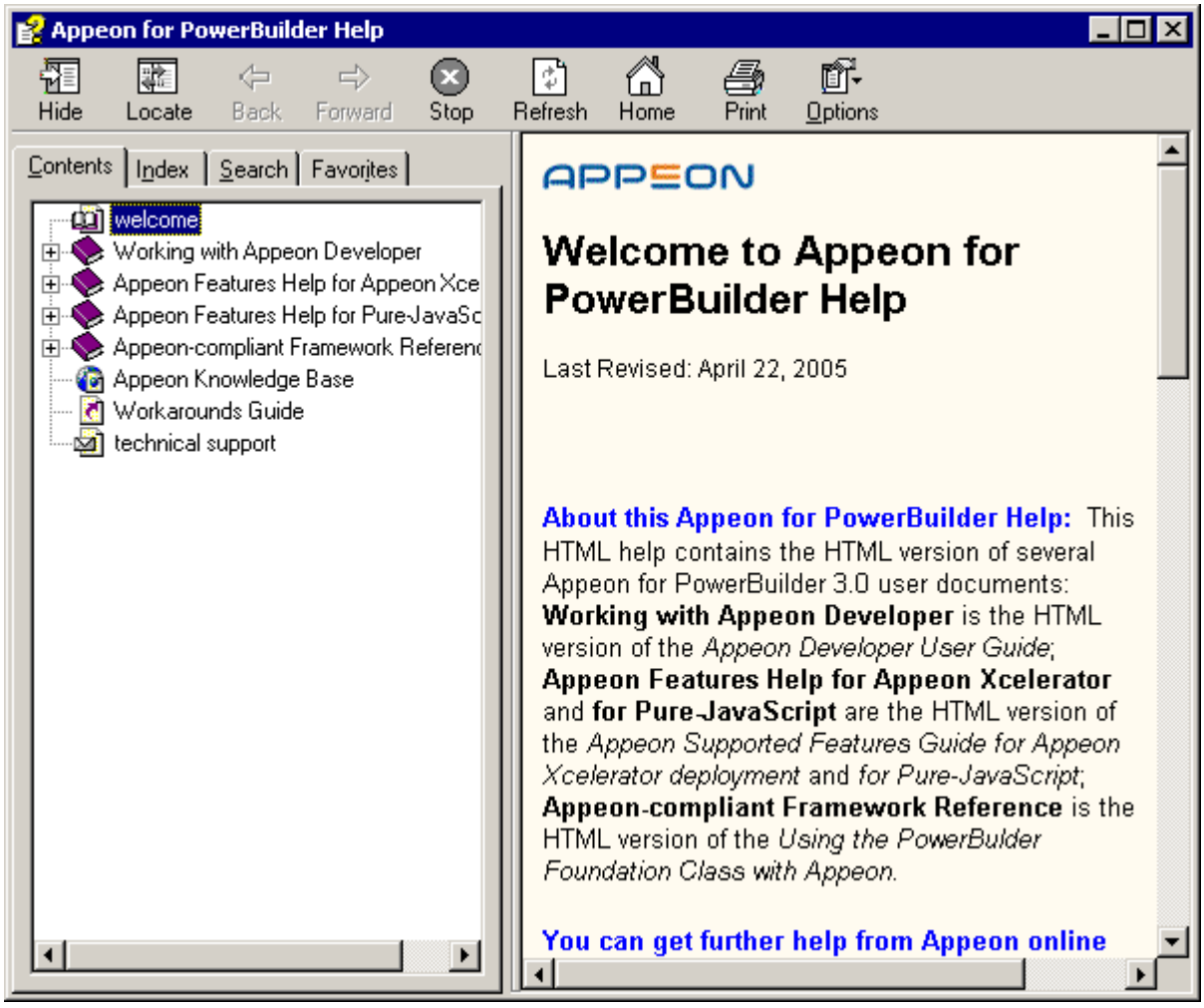
Appeon for PowerBuilder Help is in compiled HTML format, so that you can search by index or by typing keywords, navigate back and forth, or print a specified page.

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 Technical Support

Technical support information is available at http://www.sybase.com/.

When logging a case be prepared to provide the following files listed in Table 15-1 to Sybase Technical Support. This will aid debugging.

Table 15-1: Provide the following files for technical support

File Type	File Name
EAServer configuration file	%JAGUAR%\Repository\Server\Jaguar.props %JAGUAR%\bin\serverstart.bat %JAGUAR%\bin\setenv.bat
EAServer log file	%JAGUAR%\bin\Jaguar.log %JAGUAR%\bin\Jaguarhttperror.log %JAGUAR%\bin\Jaguarhttpservlet.log
Appeon Server configuration file	%JAGUAR%\appeon\config\applications-config.xml %JAGUAR%\appeon\config\server-config.xml %JAGUAR%\appeon\config\aem-config.xml
Appeon Server log file	%JAGUAR%\appeon\log\LogSystem.log %JAGUAR%\appeon\log\AppeonServer.log
Appeon Server Repository DB	%JAGUAR%\appeon\db\AppeonServer.db
Appeon license file	%JAGUAR%\appeon\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: 1) Web Server path for EAServer is usually %JAGUAR%\html 2) 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