Appeon Migration Tutorial

Appeon® 3.1 for PowerBuilder®

FOR WINDOWS

DOCUMENT ID: DC20097-01-0310-01

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 developers who want to deploy their Sybase® PowerBuilder® applications to the Web with Appeon® 3.1 for PowerBuilder®.

1.2 How to use this book

There are eight chapters in this book:

Chapter 1: About This Book

General introduction to the contents of this document.

Chapter 2: Introduction

General introduction to the tutorial process.

Chapter 3: Loading the Tutorial PowerBuilder Application

Details the steps involved in loading and running the tutorial PowerBuilder application.

Chapter 4: Configuring Appeon Developer

Details the steps involved in configuring Appeon Developer for a Web deployment.

Chapter 5: Modifying Unsupported Features

Introduces the methodology to work around the unsupported features in the tutorial.

Chapter 6: Deploying the Tutorial PowerBuilder Application

Details the steps involved in deploying the tutorial PowerBuilder application.

Chapter 7: Configuring Database Connection

Details how to create a connection cache and configure the transaction object mappings in AEM.

Chapter 8: Running the Web Application

Details the steps involved in running the converted Web application.

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 Authorized Sybase Support Partner. If you have any questions about this product, or if you need assistance during the installation process, ask the designated person to contact Sybase Technical Support or 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

This tutorial is a series of six exercises in which you convert and deploy the Appeon tutorial PowerBuilder application to the Web. By following the tutorial, you will get hands-on experience in Web migration using Appeon for PowerBuilder.

The Appeon tutorial PowerBuilder application is a small program involving database interaction. Note that it has only a small number of unsupported features for the Pure-JavaScript deployment, and no unsupported features for the Appeon Xcelerator deployment. The application is used in the tutorial as the target PowerBuilder application to be worked on for a complete Web migration, and the Pure-JavaScript deployment will be selected to produce a complete Web migration with detailed unsupported feature modification steps.

This tutorial is a simplified and practical example of the Appeon Web migration methodology laid out in Chapter 2: *Migration Process* in the *Appeon Migration Guide*. It can serve as a starting point for any developer who wants to convert their existing PowerBuilder applications. After the successful migration of the tutorial application, you can try to convert more complex, real PowerBuilder applications into a more complicated network environment.

How you will proceed:

Lesson 1	Load the Appeon tutorial application into PowerBuilder.
Lesson 2	Configure Appeon Developer.
Lesson 3	Modify the tutorial application to remove and work around the unsupported features.
Lesson 4	Deploy the tutorial application to the Web.
Lesson 5	Create a connection cache in EAServer and configure the transaction object mapping in Appeon Enterprise Manager.
Lesson 6	Run the Web application.

What you will learn:

Lesson 1	How to create a PowerBuilder workspace, load application PBL files, set up an ODBC data source, and run the application from the PowerBuilder IDE.
Lesson 2	Add the Application Profile, the Appeon Server Profile, the Web Server Profile, and the Deployment Profile in Appeon Developer Configuration.
Lesson 3	Perform unsupported feature analysis, place unsupported features in Appeon Server as NVOs, remove unsupported features, work around unsupported features, perform a Full build, and optimize the tutorial application.
Lesson 4	Deploy the tutorial application to the Web using the Appeon Deployment Wizard.
Lesson 5	How to login to Appeon Server using EAServer Manager, add Appeon Server connection caches, launch and login AEM, and add Web application database configuration.
Lesson 6	Run the deployed Web application in Internet Explorer.

How long it will take:

You can complete the entire tutorial in ninety minutes. Each lesson has also been designed so that you can stop after any lesson and continue at another time.

2.2 Preparing for the tutorial

The following preparation is required before starting this tutorial:

- One Workstation.

  This tutorial assumes the most simplified network environment. Only one physical machine is used for all the different roles in the n-Tier Web architecture: Client PC, Web Server, Application Server, Database Server and Developer PC.

  Refer to the *Appeon Installation Guide* for specific system requirements.

- Install other required software:

  Microsoft Windows 2000, Sybase PowerBuilder 8, 9 or 10, EAServer 5.2, and Internet Explorer 6.0 or above.

  Note:

  1) EAServer can be installed from the Sybase EAServer setup program or silently installed with Appeon Server during the Appeon Server installation.
  2) EAServer will be used as both the Application Server and the Web Server in this tutorial.

- Install Appeon for PowerBuilder

  Install Appeon Developer and Appeon Server.

  In order to install the demo application that the tutorial will use, you MUST either select the complete setup or choose the custom setup with the "Demo Applications" feature selected during the Appeon Developer installation.

2.3 Relevant files

The tutorial will use the following files:

File Name	*appeontutor.pbl*	*appeontutor.db*
Location	\Appeondemo\Tutorial (e.g. C:\Program Files\Appeon\Developer\Appeondemo\Tutorial)	
Description	The PowerBuilder Library (application source code) used in the tutorial.	The ASA database file used for the tutorial application.

3 Loading the Tutorial PowerBuilder Application

This tutorial starts with two files: *appeontutor.pbl* and *appeontutor.db*. The *appeontutor.pbl* file contains the source code for the Appeon tutorial PowerBuilder application. First, follow the instructions to make the tutorial application run in PowerBuilder.

In this section, you will learn how to:

- Create a new PowerBuilder workspace
- Load the tutorial PBLs into the workspace
- Configure the ODBC data source for the tutorial application
- Run the tutorial application

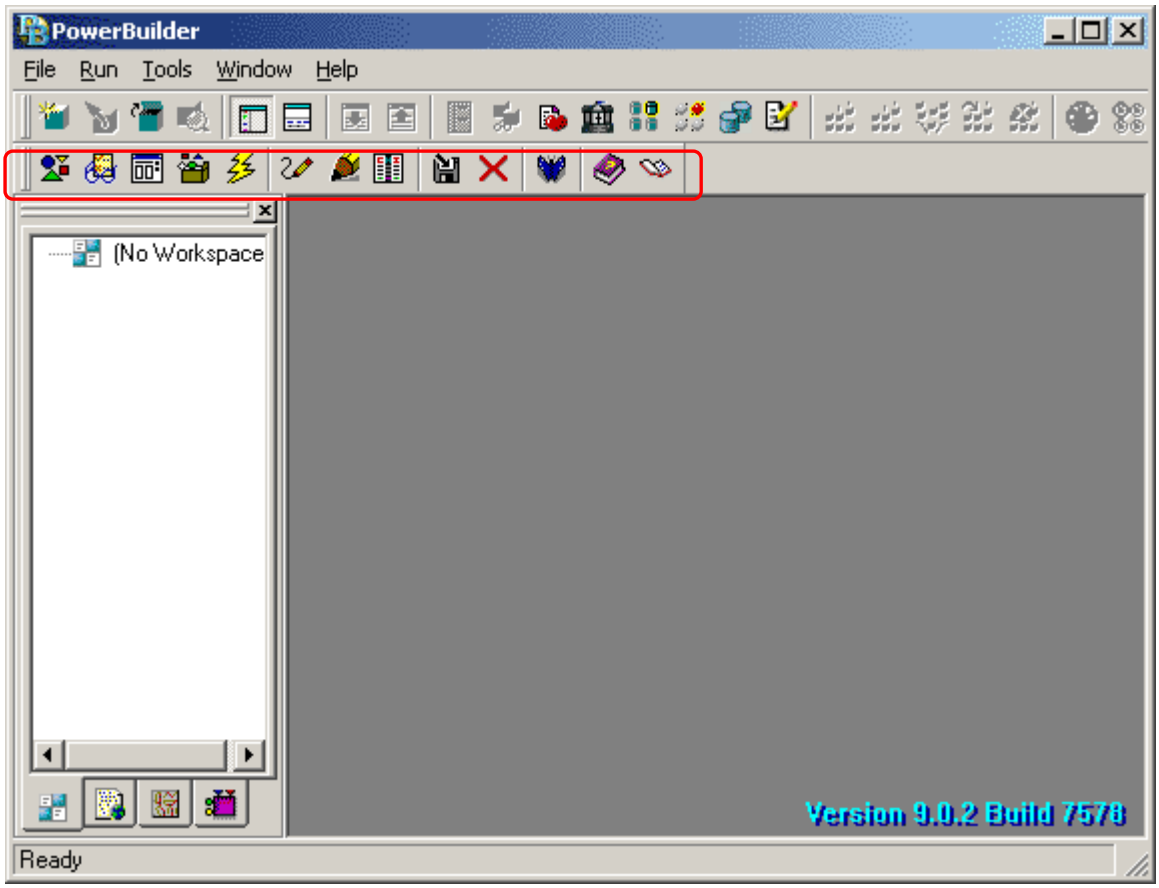
3.1 Creating a workspace

You can have only one PowerBuilder Workspace open at a time, but you can add as many targets or applications to the Workspace as you want, including opening and editing objects in multiple targets.

To create a new Workspace for the Appeon tutorial PowerBuilder application:

STEP 1 – In Windows: select *Start* | *Programs* | *Appeon 3.1 for PowerBuilder* | *PowerBuilder*. The PowerBuilder IDE starts with the Appeon Developer toolbar loaded, as shown in Figure 3-1.

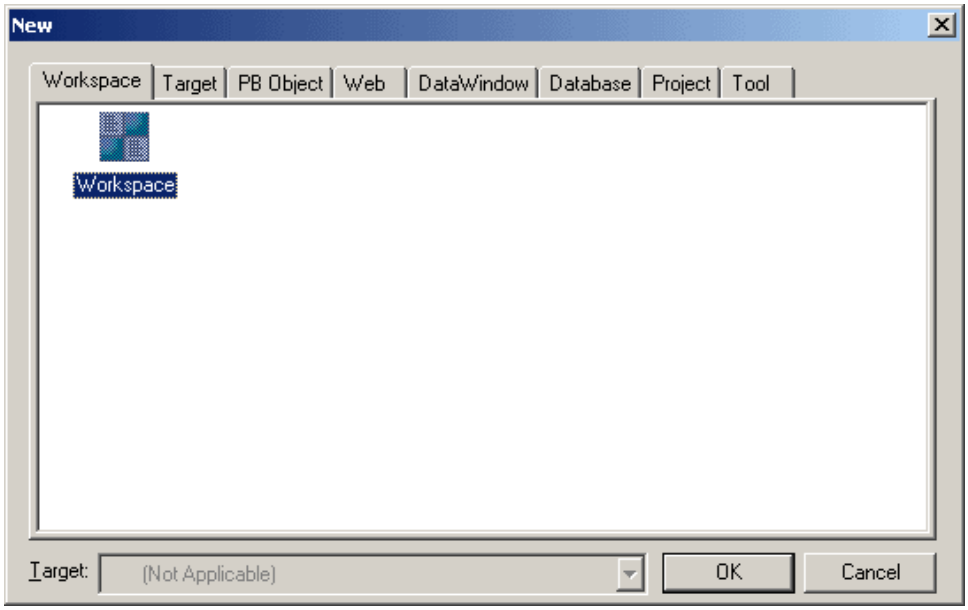
Figure 3-1: Appeon Developer tool bar loaded into PowerBuilder



STEP 2 – Select *File* | *New* from the PowerBuilder menu bar, and the *New* dialog box appears.

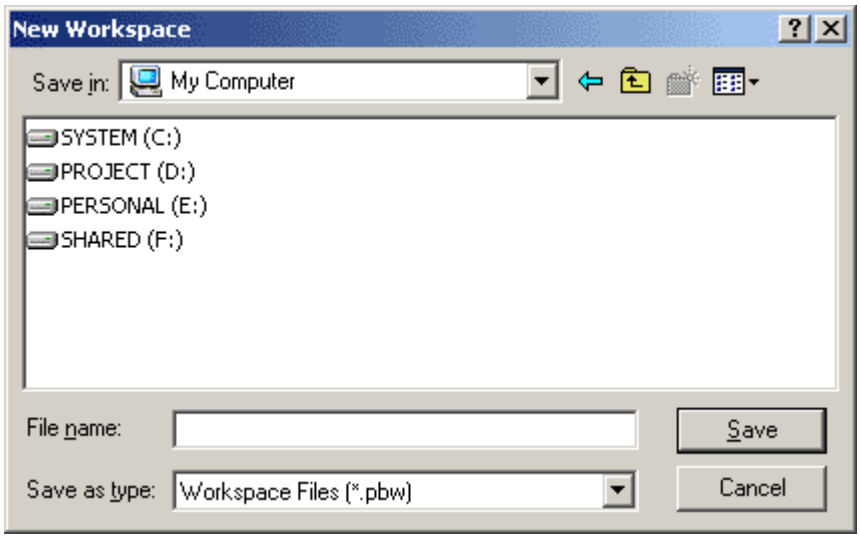
Click the Workspace tab if it is not already selected, as shown in Figure 3-2.

Figure 3-2: Adding a new Workspace



STEP 3 – Select the Workspace icon as shown in Figure 3-2 and click *OK*. The New Workspace dialog box displays, as shown in Figure 3-3.

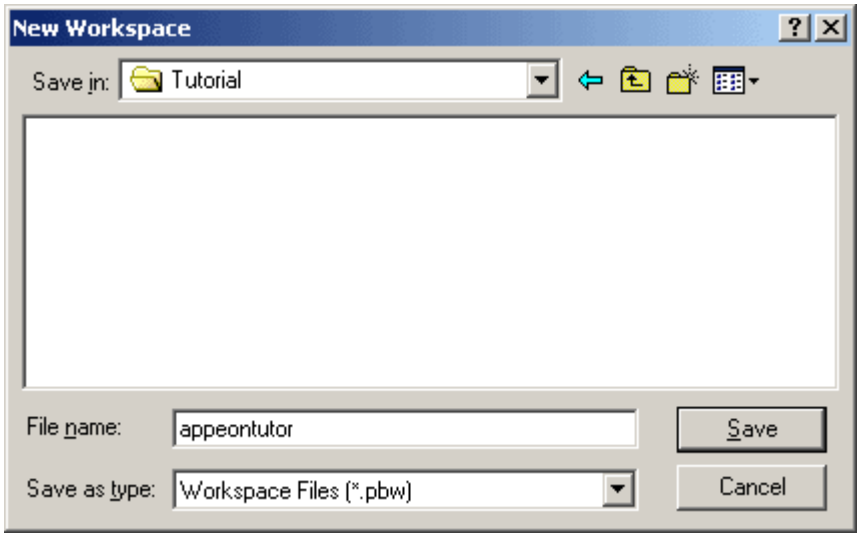
Figure 3-3: New Workspace



STEP 4 – Navigate to the *%Appeon%\Developer\Appeondemo\Tutorial* folder.

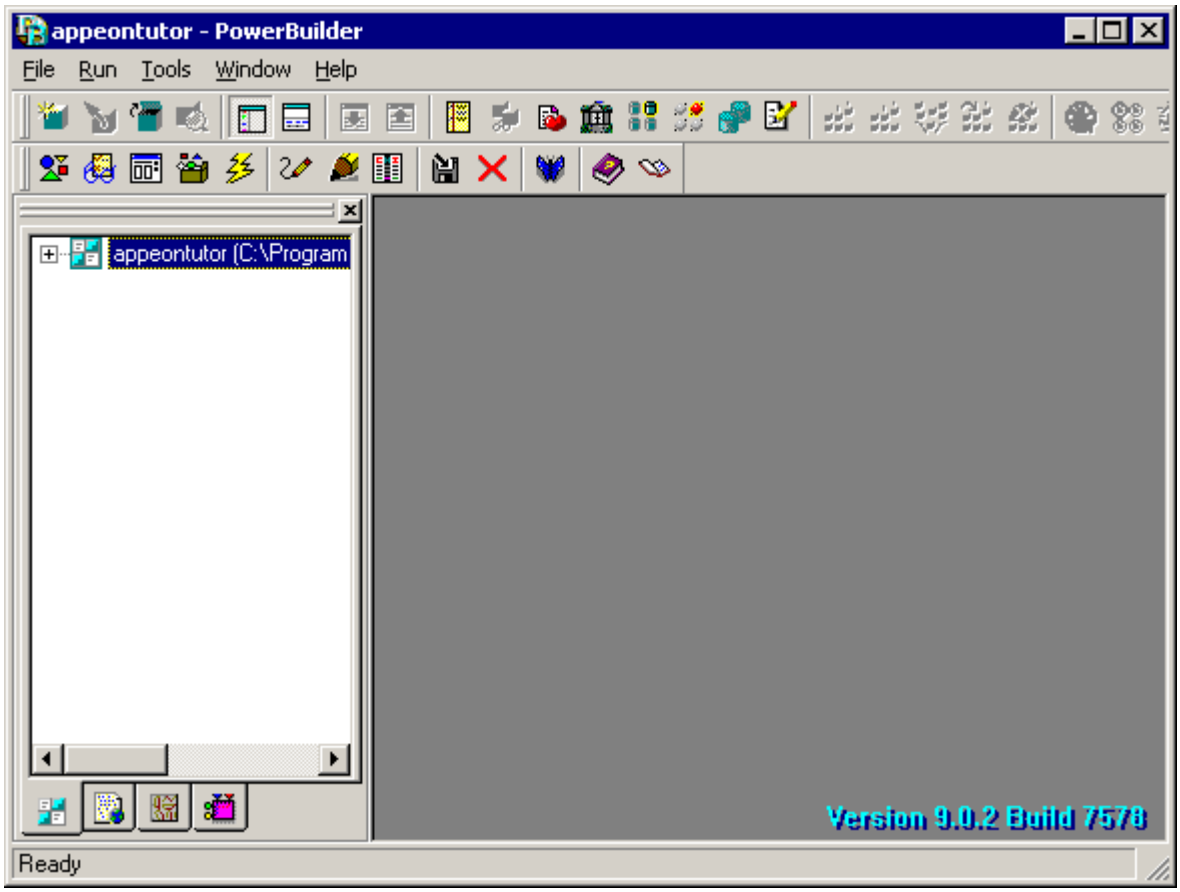
Type *appeontutor* in the File name text box and click *Save*, as shown in Figure 3-4.

Figure 3-4: Naming the new Workspace



STEP 5 – The *appeontutor* workspace has been added and appears as the top item in the PowerBuilder system tree, as shown in Figure 3-5.

Figure 3-5: Newly created Workspace



3.2 Loading the tutorial PBL file

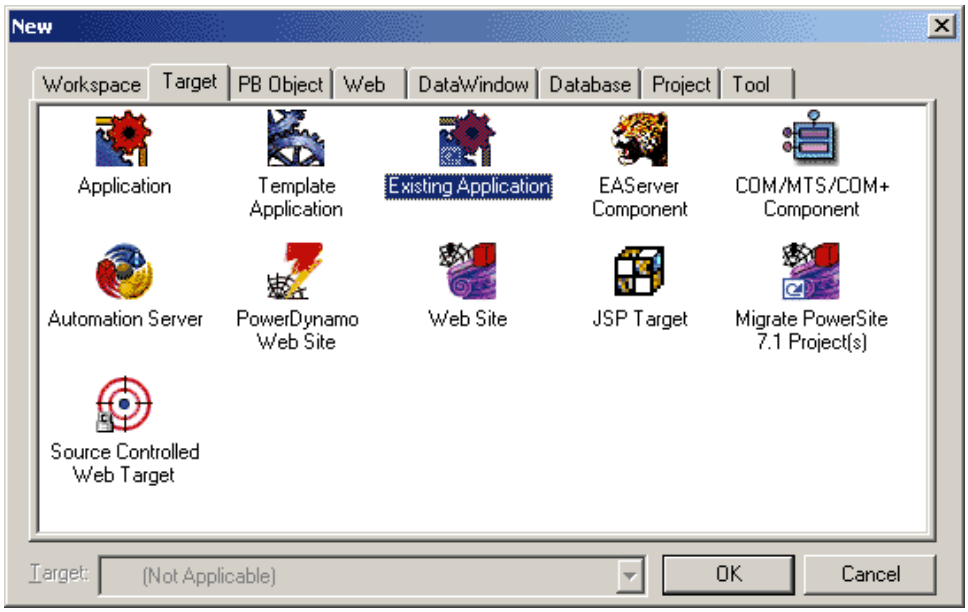
A PowerBuilder Library (*.pbl file) is a collection of compiled object definitions and source objects (including scripts) stored in the same location. The PowerBuilder painters and wizards store various objects in libraries, such as Applications, Windows, DataWindows, Menus, Functions, Structures, User Objects.

Now load the PBL file of the Appeon tutorial application into the newly created Workspace:

STEP 1 – Select *File* | *New* from PowerBuilder menu bar, and the New dialog box appears.

Click the Target tab if it is not already selected, as shown in Figure 3-6.

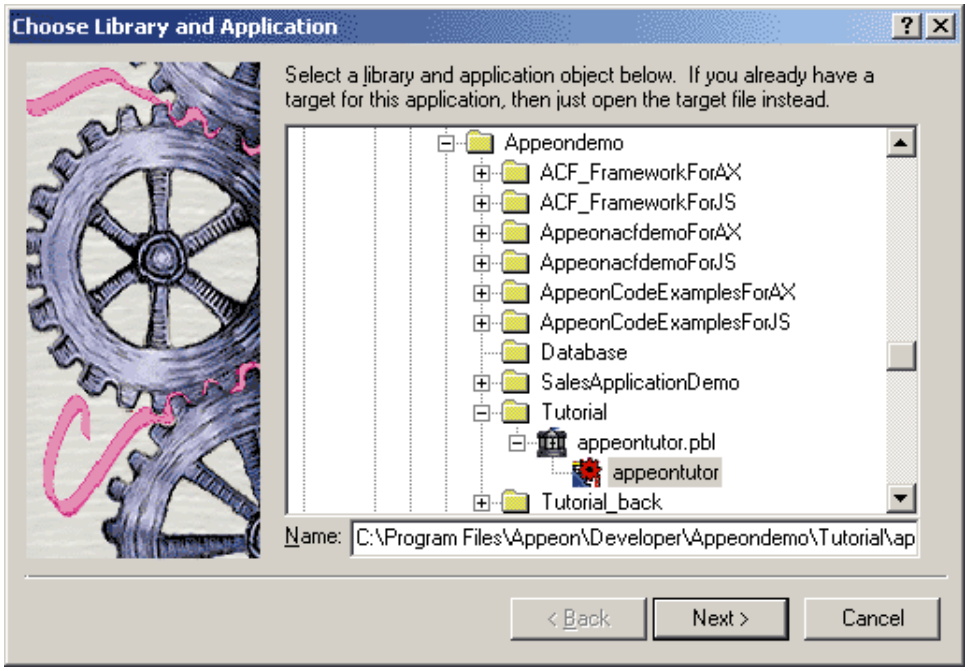
Figure 3-6: Target tab page



STEP 2 – Select the *Existing Application* icon as shown in Figure 3-6 and click *OK*. Now the Choose Library and Application dialog box is displayed, as shown in Figure 3-7.

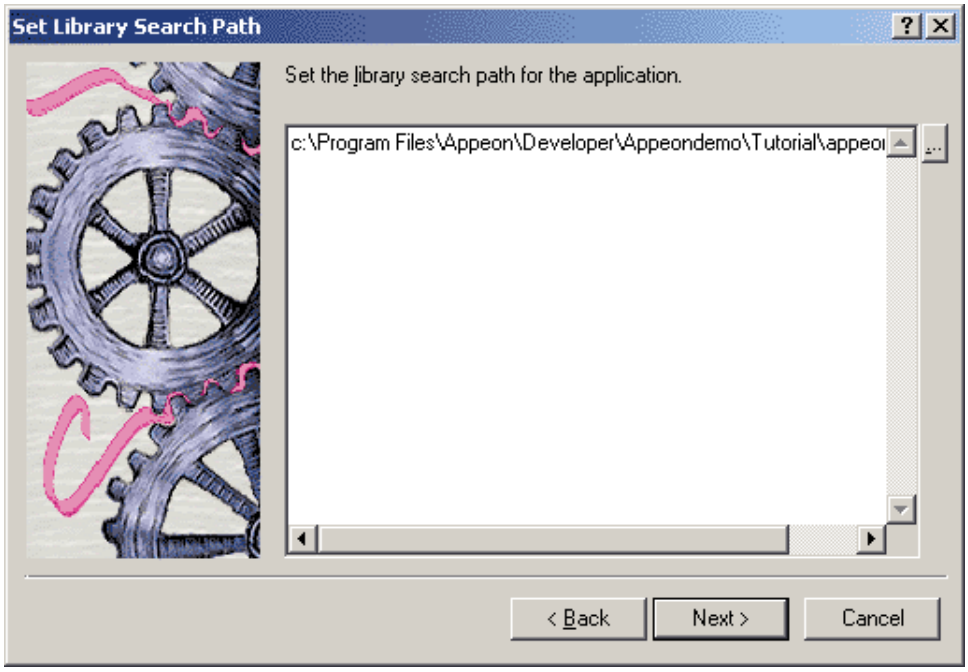
Select the *appeontutor* Application under the *appeontutor.pbl*, and click *Next*, as shown in Figure 3-7.

Figure 3-7: Choose Library and Application



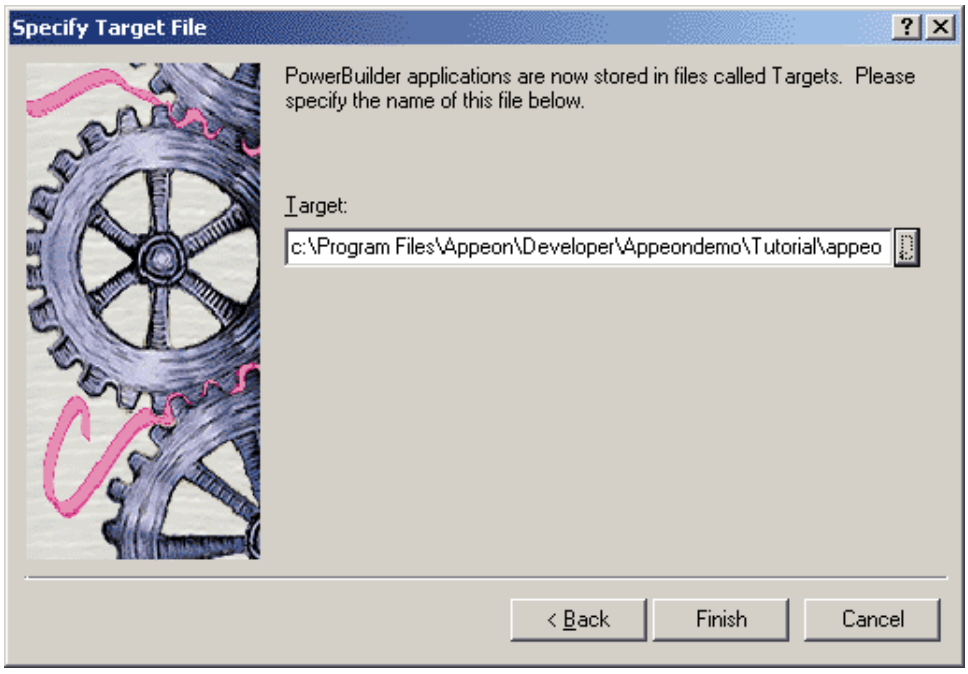
STEP 3 – Click *Next* on the Set Library Search Path dialog box, as shown in Figure 3-8.

Figure 3-8: Set Library Search Path



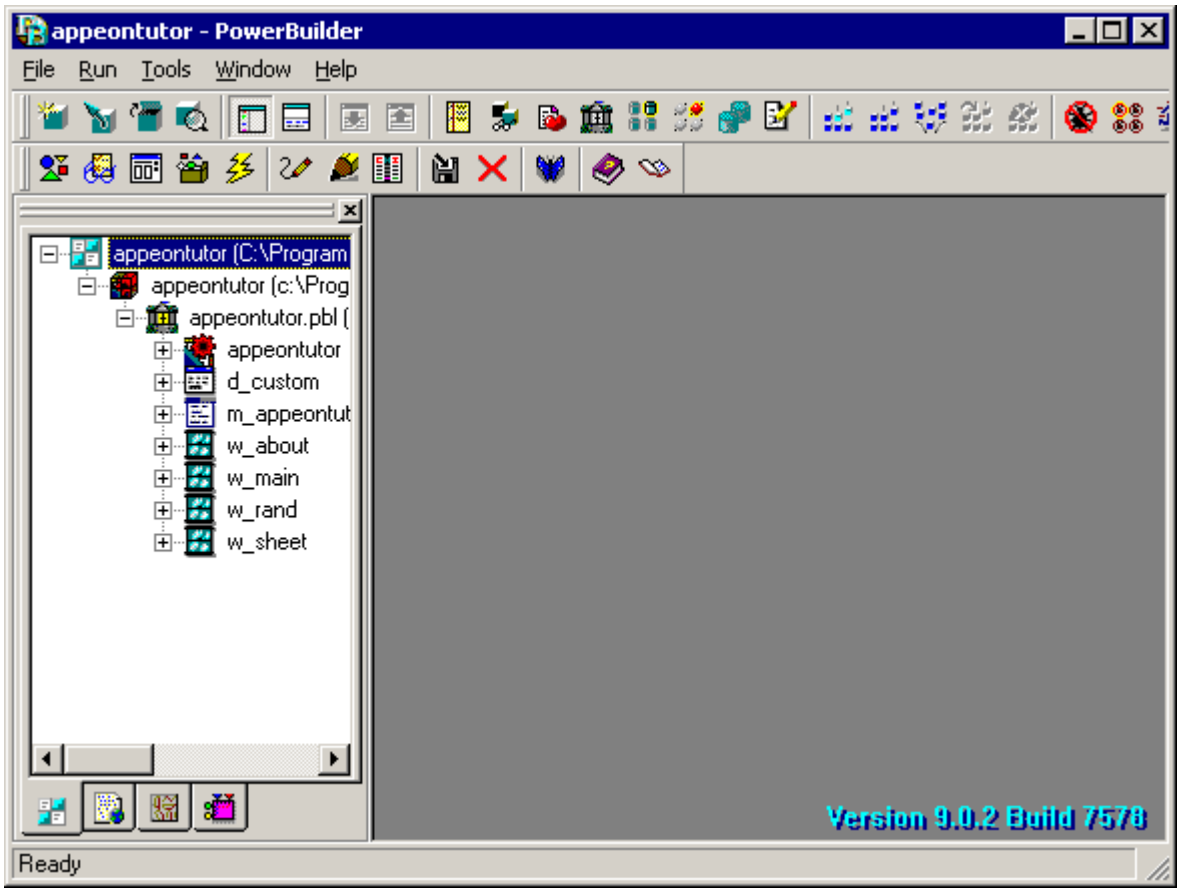
STEP 4 – Click *Finish* on the Specify Target File dialog box, as shown in Figure 3-9.

Figure 3-9: Specify Target File



STEP 5 – The Appeon tutorial PowerBuilder application has now been added to the *appeontutor* Workspace and is displayed in the system tree, as shown in Figure 3-10.

Figure 3-10: Newly added "appeontutor" Workspace



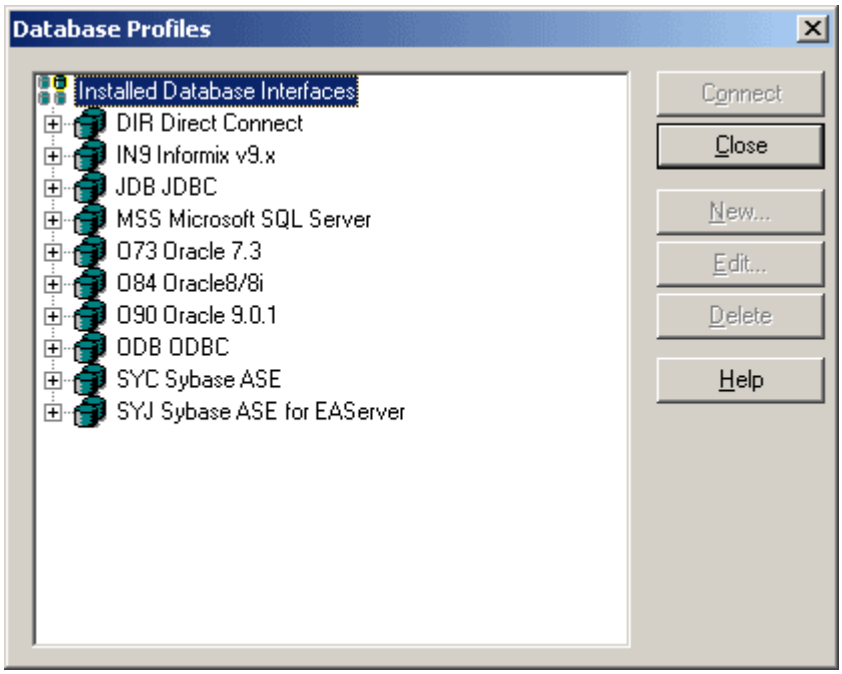
3.3 Configuring ODBC data source

An ODBC data source stores the parameters used to connect to the indicated data provider through the Open Database Connectivity interface. By setting up the ODBC data source that identifies the ASA database file for the Appeon tutorial PowerBuilder application, the tutorial application is able to establish a connection with the ASA database through a reference to the ODBC data source name.

To set up the ODBC data source for the tutorial application:

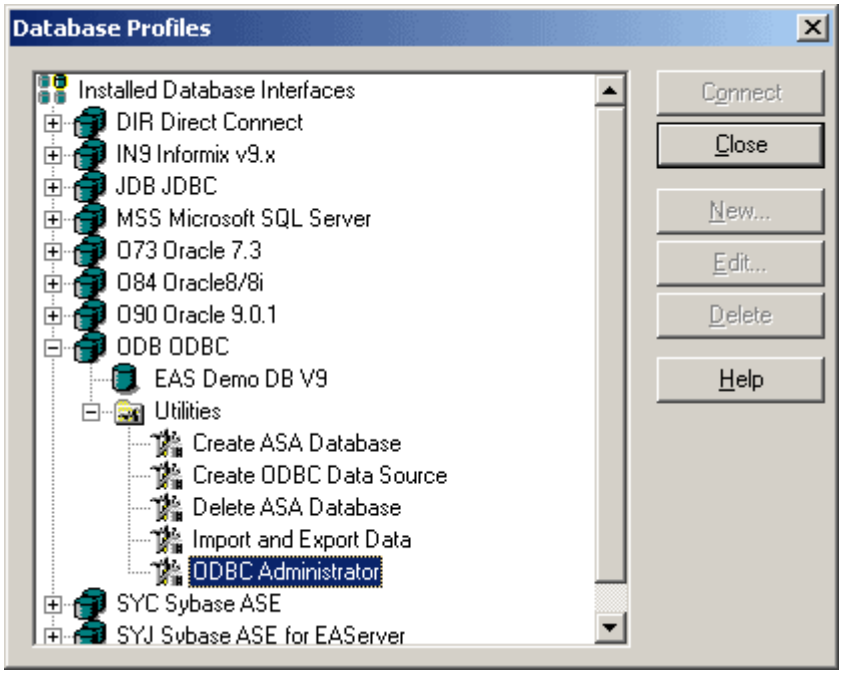
STEP 1 – Select *Tools* | *Database Profile* from the PowerBuilder menu bar, and the Database Profiles dialog box displays, as shown in Figure 3-11.

Figure 3-11: Database Profiles



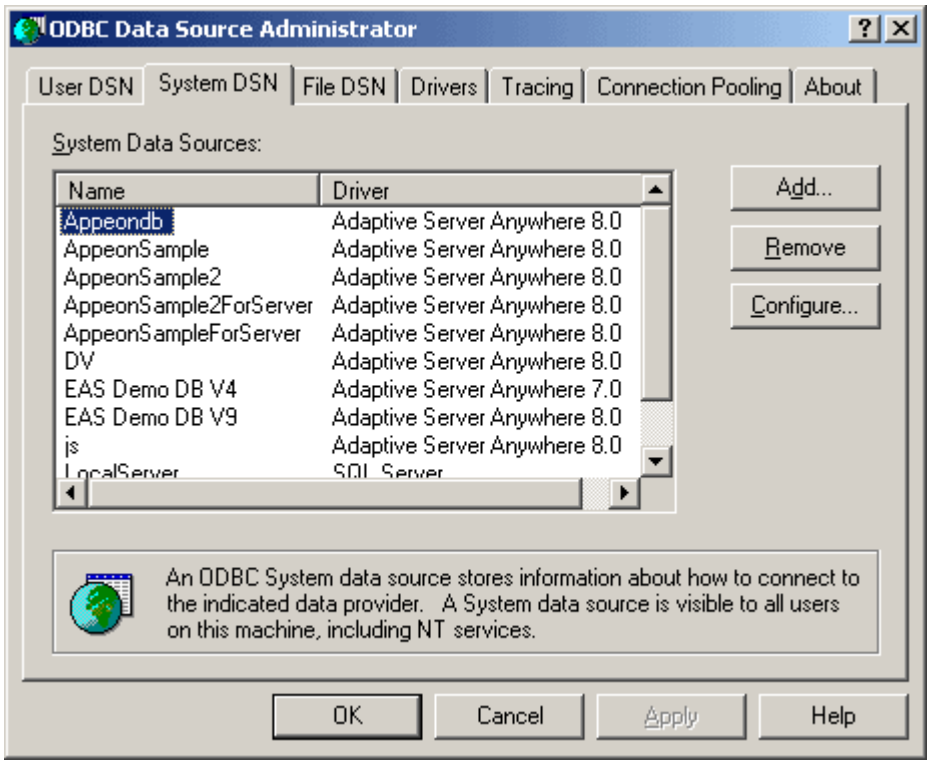
STEP 2 – Double click *ODBC Administrator* under Utilities under ODB ODBC, as shown in Figure 3-12.

Figure 3-12: Database Profiles



STEP 3 – Select the System DSN tab in the ODBC Data Source Administrator dialog box and click *Add*, as shown in Figure 3-13.

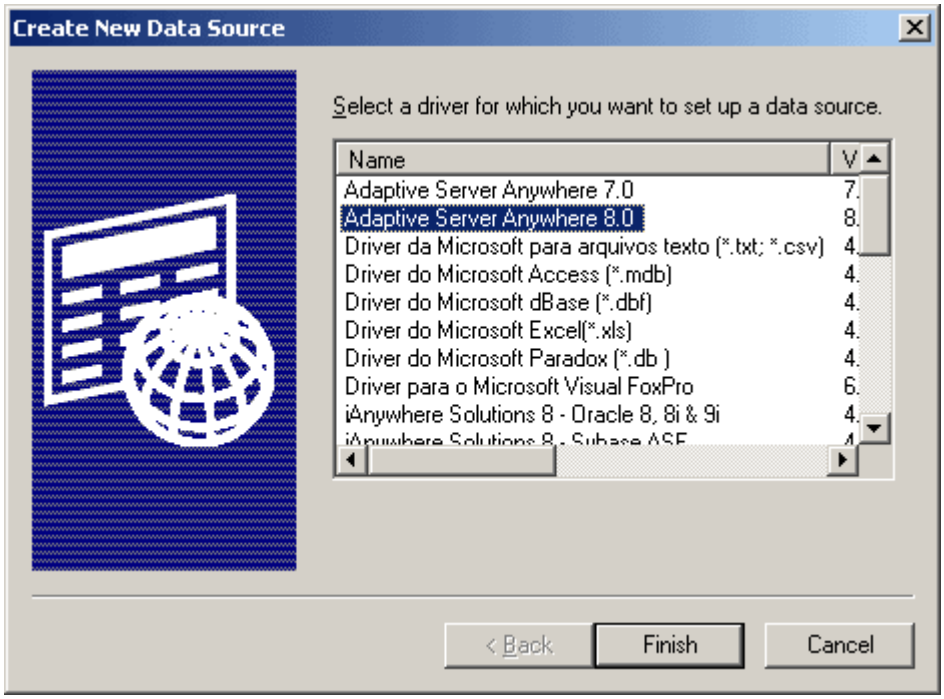
Figure 3-13: ODBC Data Source Administrator



STEP 4 – Select *Adaptive Server Anywhere 8.0* and click *Finish* in the Create New Data Source dialog box, as shown in Figure 3-14.

Select the Sybase Adaptive Server Anywhere 8.0 driver to connect to the *appeontutor.db* database file.

Figure 3-14: Create a Data Source



STEP 5 – The ODBC Configuration for Adaptive Server Anywhere dialog box is displayed. Type the necessary settings into the different tabs as specified in Table 3-1, and leave other fields at their default settings.

Table 3-1: Data source settings

In this tab…	In this field…	You should…
ODBC Figure 3-15	Data source name	Type *appeontutor*
Login Figure 3-16	User ID	Type *dba* (case sensitive)
	Password	Type *sql* (case sensitive)
Database Figure 3-17	Server name	Type *appeontutor*
	Database name	Type *appeontutor*
	Database file	Click *Browse* to select the *appeontutor.db* file. For example, *C:\Program Files\Appeon\Developer\Appeondemo\Tutorial\appeontutor.db*

Figure 3-15: ODBC tab

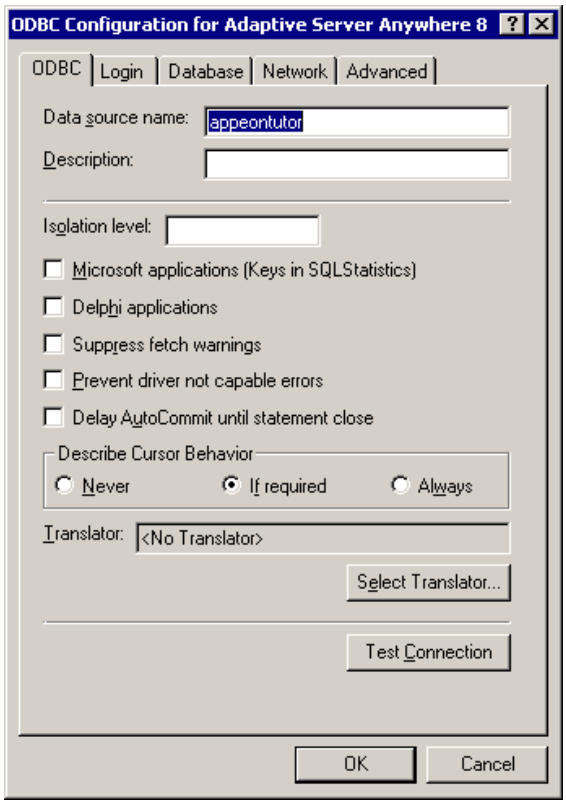


Figure 3-16: Login tab

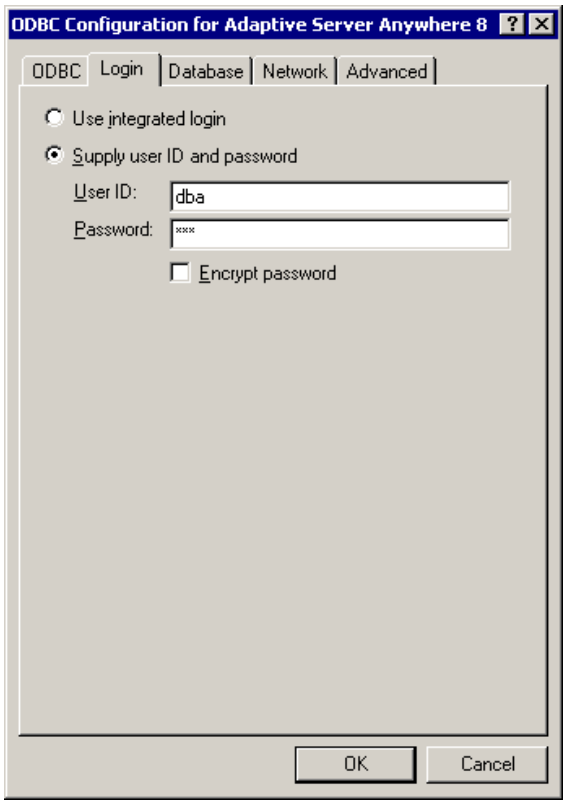
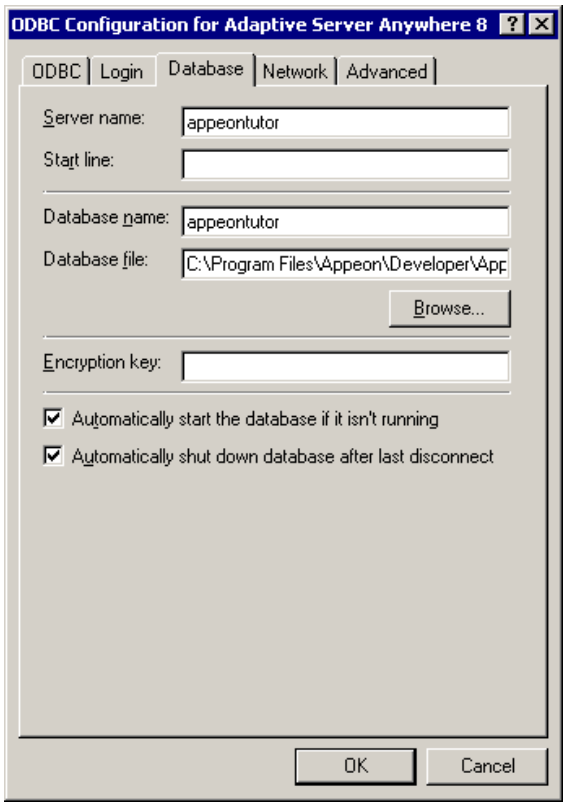


Figure 3-17: Database tab

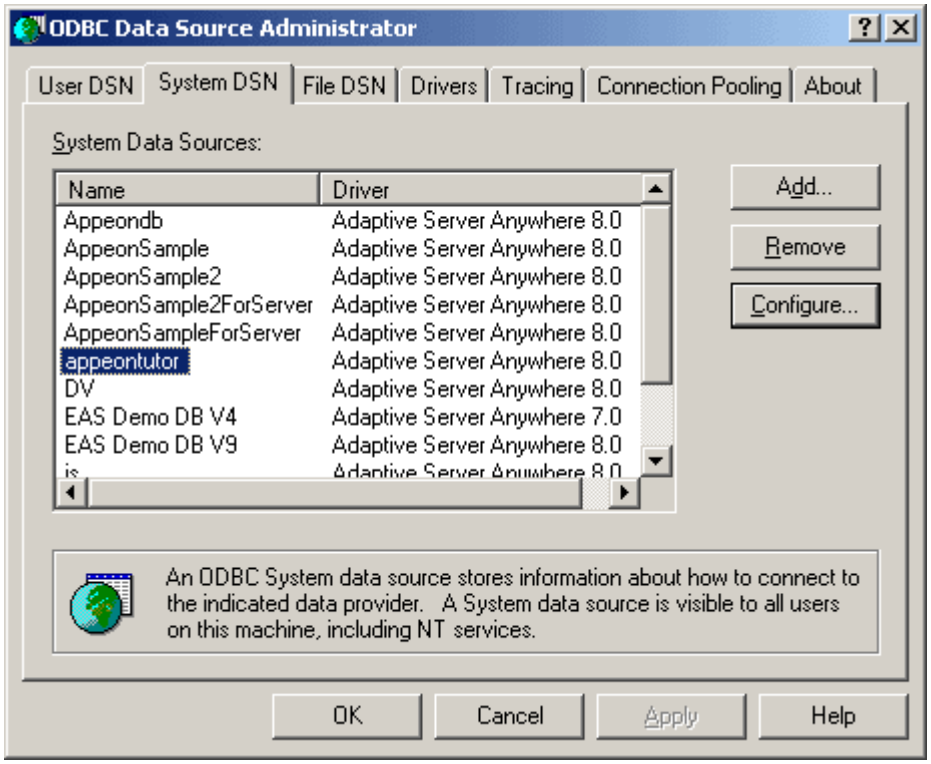


STEP 6 – Go back to the ODBC tab and click *Test Connection*. Ensure the connection test is successful before continuing.

STEP 7 – Click *OK* to close the ODBC Configuration for Adaptive Server Anywhere dialog box.

The *appeontutor* is added as a system data source under the System DSN tab, as shown in Figure 3-18.

Figure 3-18: ODBC Data Source Administrator



STEP 8 – Click *OK* to close the dialog box (shown in Figure 3-18) and click *Close* to exit the Database Profiles dialog box (shown in Figure 3-12).

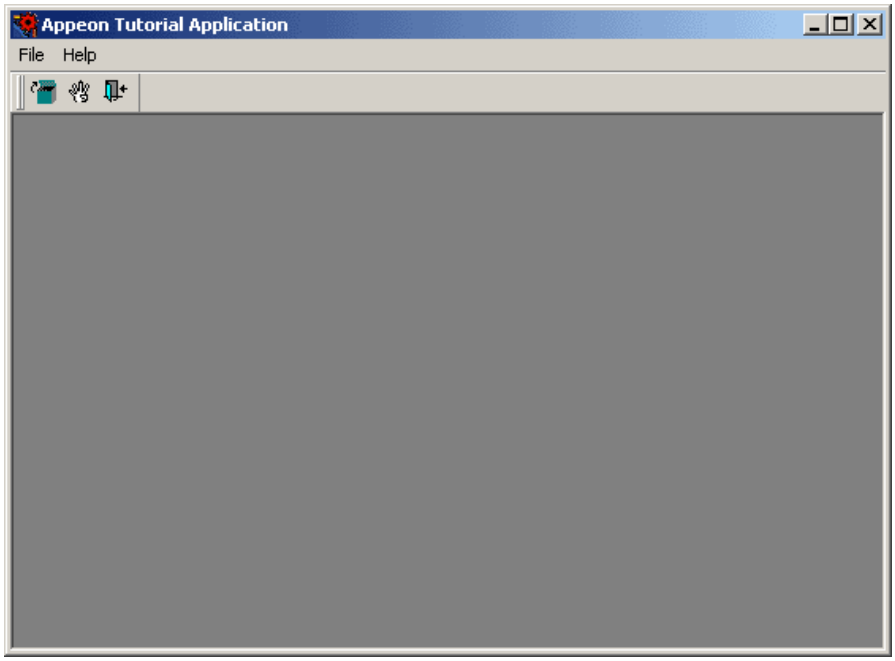
3.4 Running the tutorial application

The database connection parameters (PowerBuilder Transaction properties) have been preset in the tutorial PowerBuilder application that will connect to the data source named *appeontutor*. You can run the tutorial application in PowerBuilder now.

To run the Appeon tutorial PowerBuilder application:

STEP 1 – In the PowerBuilder IDE, choose *Run* | *Run* from the PowerBuilder menu bar, or click the *Run* button in PowerBar1. The Appeon tutorial PowerBuilder application starts, as shown in Figure 3-19.

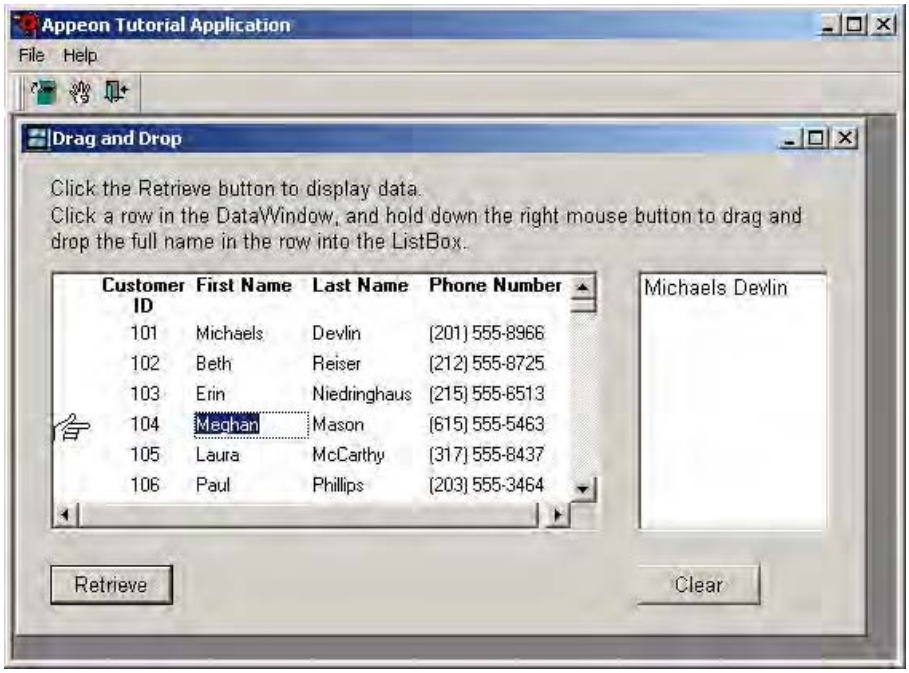
Figure 3-19: Appeon Tutorial Application



STEP 2 – In the Appeon Tutorial Application, select *File* | *Drag and Drop* to open the Drag and Drop window, as shown in Figure 3-20.

The Appeon tutorial PowerBuilder application has an MDI window, two sheet windows, and an About window. In the Drag and Drop window, click the *Retrieve* button to display data. Click a row in the DataWindow and hold down the right mouse button to drag and drop the full name in the row into the right ListBox. Click the *Clear* button to reset the ListBox.

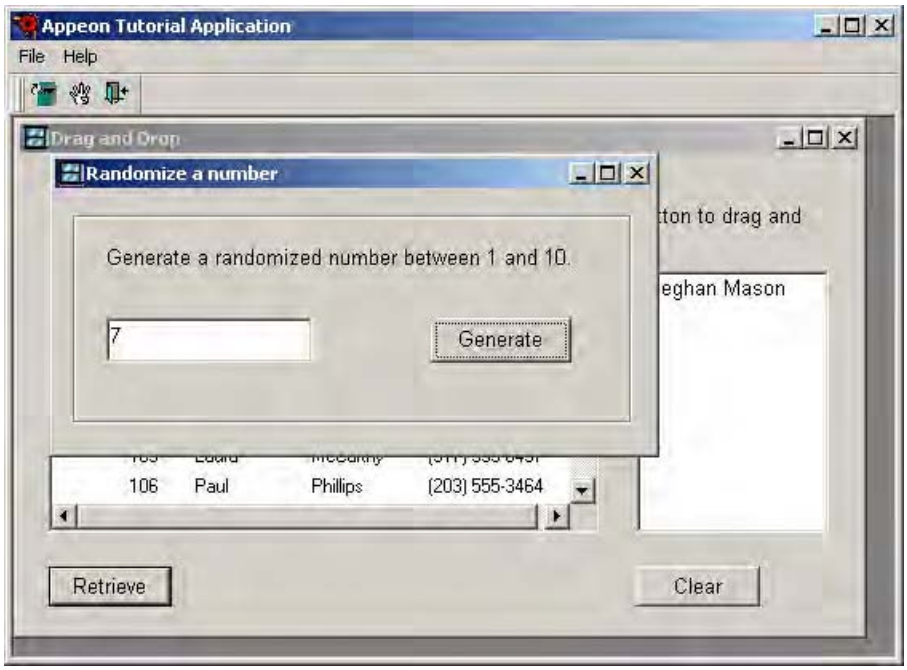
Figure 3-20: Drag and Drop



STEP 3 – Select *File* | *Randomize* to open the Randomize a number window.

This window uses the PowerBuilder *Randomize* and *Rand* functions to randomly generate an integer between 1 and 10, and display it in a window, as shown in Figure 3-21.

Figure 3-21: Randomize a number



STEP 4 – Close the tutorial application, and return to the PowerBuilder IDE.

4 Configuring Appeon Developer

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. The Appeon Developer toolbar automatically loads each time PowerBuilder is opened.

Figure 4-1: Appeon Developer toolbar

The Appeon tutorial PowerBuilder application should have the following four configurations made correctly before Appeon Developer can automate the task of Web conversion and deployment of the tutorial application.

- Add an application profile
- Add an Appeon Server profile
- Add a Web Server profile
- Add a deployment profile

4.1 Adding an application profile

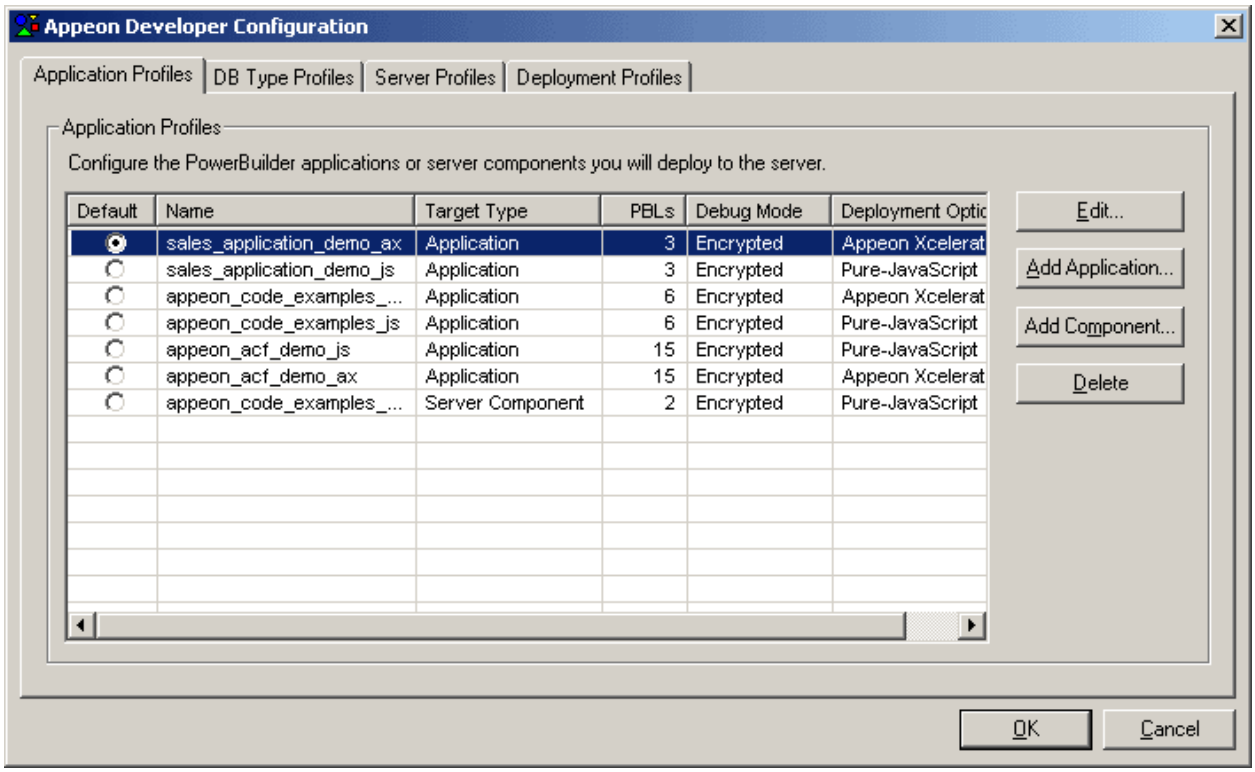
Each application profile maps to one PowerBuilder application. The application profile provides important information of an application to be used in the PowerBuilder-to-Web conversion.

To add an application profile for the tutorial application:

STEP 1 – Start PowerBuilder and Appeon Server.

STEP 2 – Click the *Configure* button () in the Appeon Developer toolbar. The Appeon Developer Configuration window is displayed, as shown in Figure 4-2.

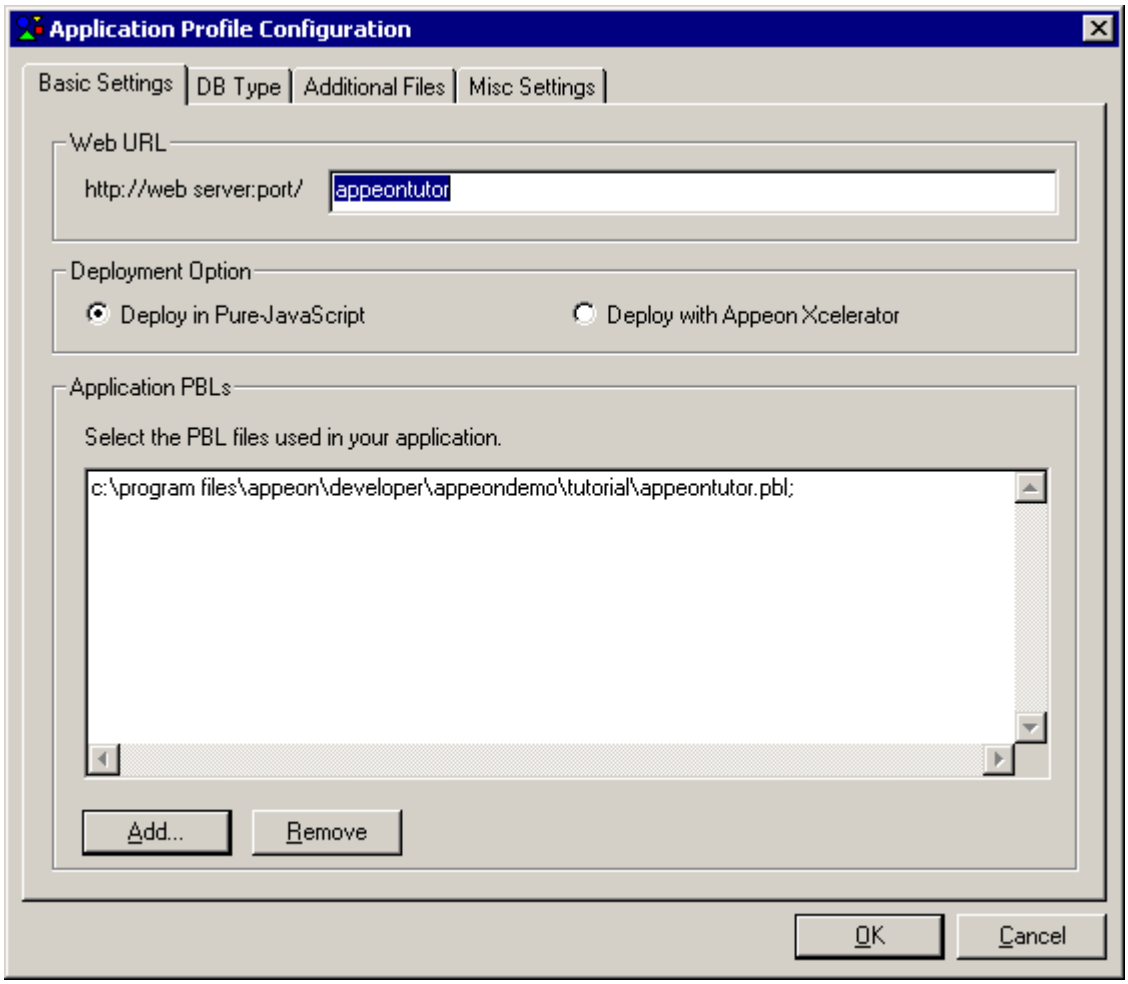
Figure 4-2: Add an application profile



STEP 3 – Click *Add Application* under the *Application Profiles* tab.

The *Application Profile Configuration* window is displayed, as shown in Figure 4-3.

Figure 4-3: Application Profile Configuration window



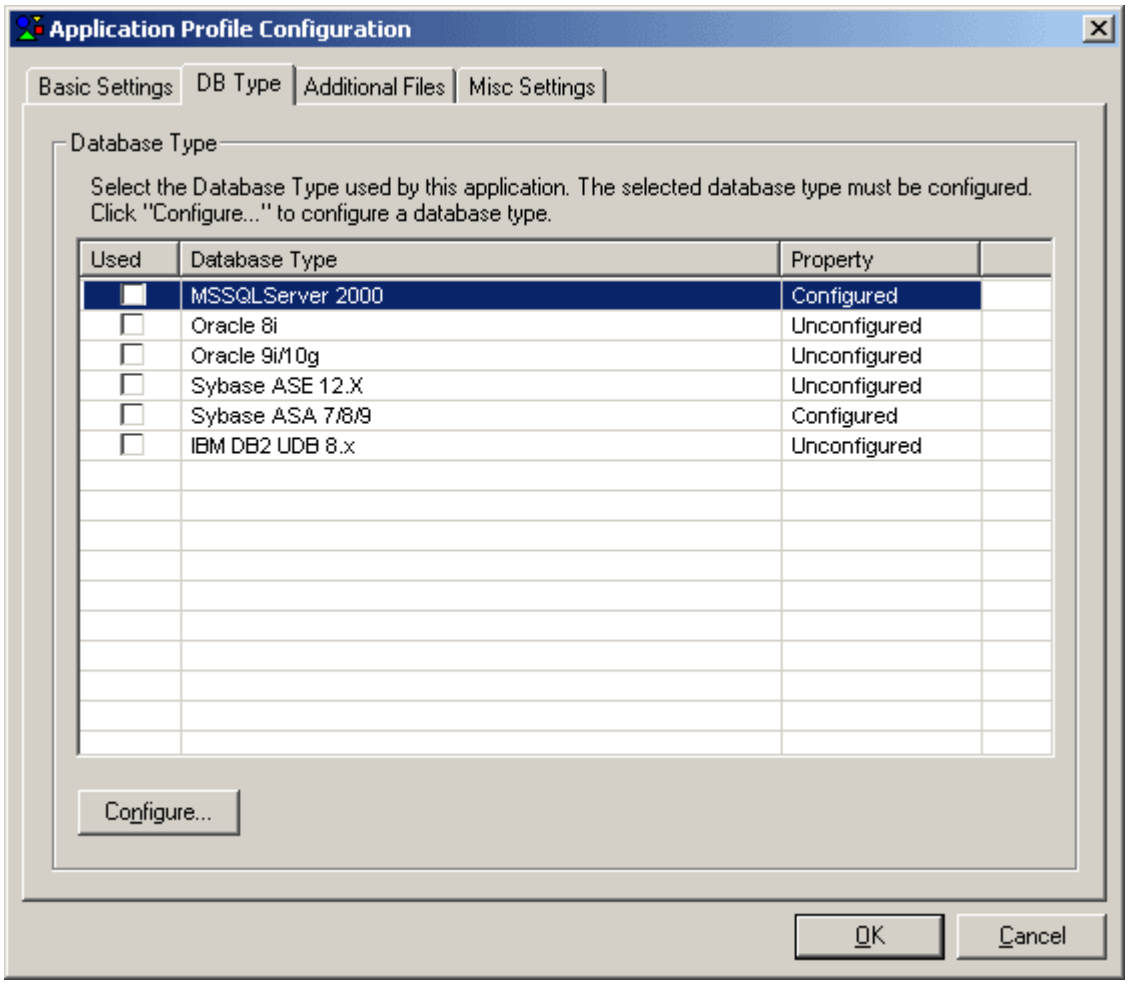
STEP 4 – On the *Basic Settings* tab, specify the following settings:

Table 4-1: Basic settings

In this group box…	You should…
Web URL	Type *appeontutor*. The input text will not only be used as the Web deployment path, but also as the name of the folder created in the home directory of the Web Server for storing the Web files generated during deployment.
Deployment Option	Select *Deploy with Pure-JavaScript* radio button.
Application PBLs	Click *Add* to add the *appeontutor.pbl* file in the *%Appeon%\Developer\Appeondemo\Tutorial* folder to the Application PBLs list.

STEP 5 – Click the *DB Type* tab to select the database type used by the application, as shown in Figure 4-4.

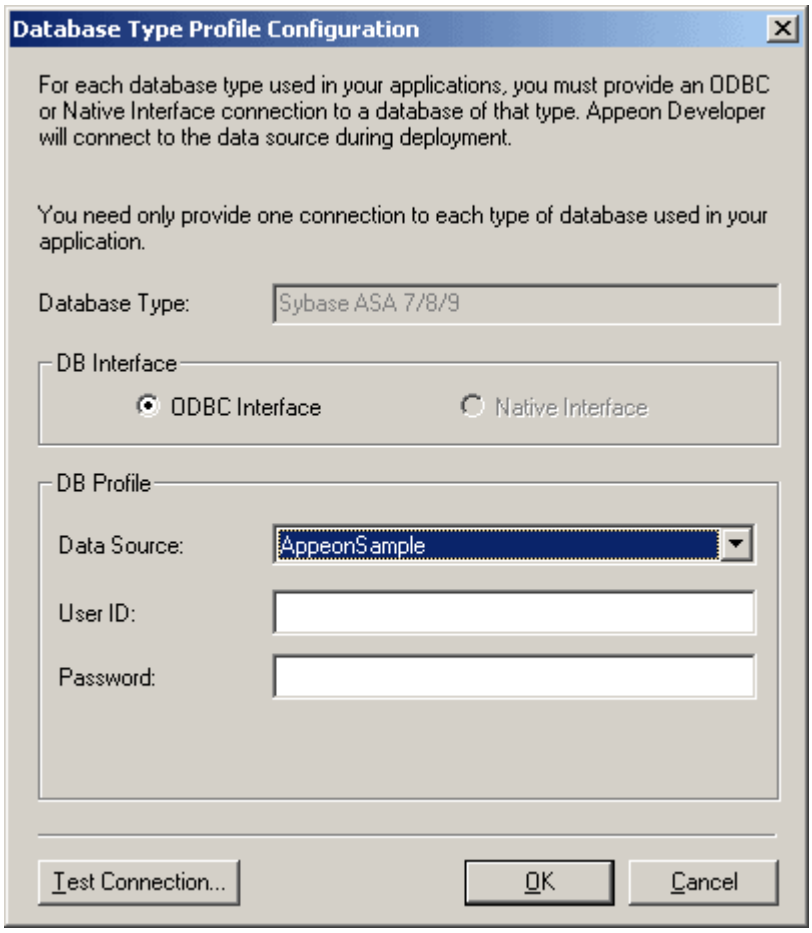
Figure 4-4: DB Type



STEP 6 – Select *Sybase ASA 7/8/9* in the *Database Type* column.

If Sybase ASA 7/8/9 is indicated as "Unconfigured", you should click the *Configure* button to create a profile for it. The *Database Type Profile Configuration* window will be displayed, as shown in Figure 4-5.

Figure 4-5: Database Type Profile Configuration window



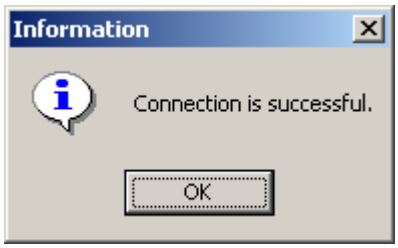
Specify the following settings in the *Database Type Profile Configuration* window, as shown in Table 4-2.

Table 4-2: Database type profile settings

In this field	You should
DB Interface	Choose the *ODBC Interface* radio button.
Data Source	Choose *appeontutor* from the dropdown listbox. If *appeontutor* data source does not exist, you can choose the Appeon demo database (*appeonsample*), or any other Sybase ASA data source.
User ID	Input *dba*. If selecting the *appeonsample* data source, leave this field blank.
Password	Input *sql*. If selecting the *appeonsample* data source, leave this field blank.

Click the *Test Connection* button to verify that the connection is successful, as shown in Figure 4-6.

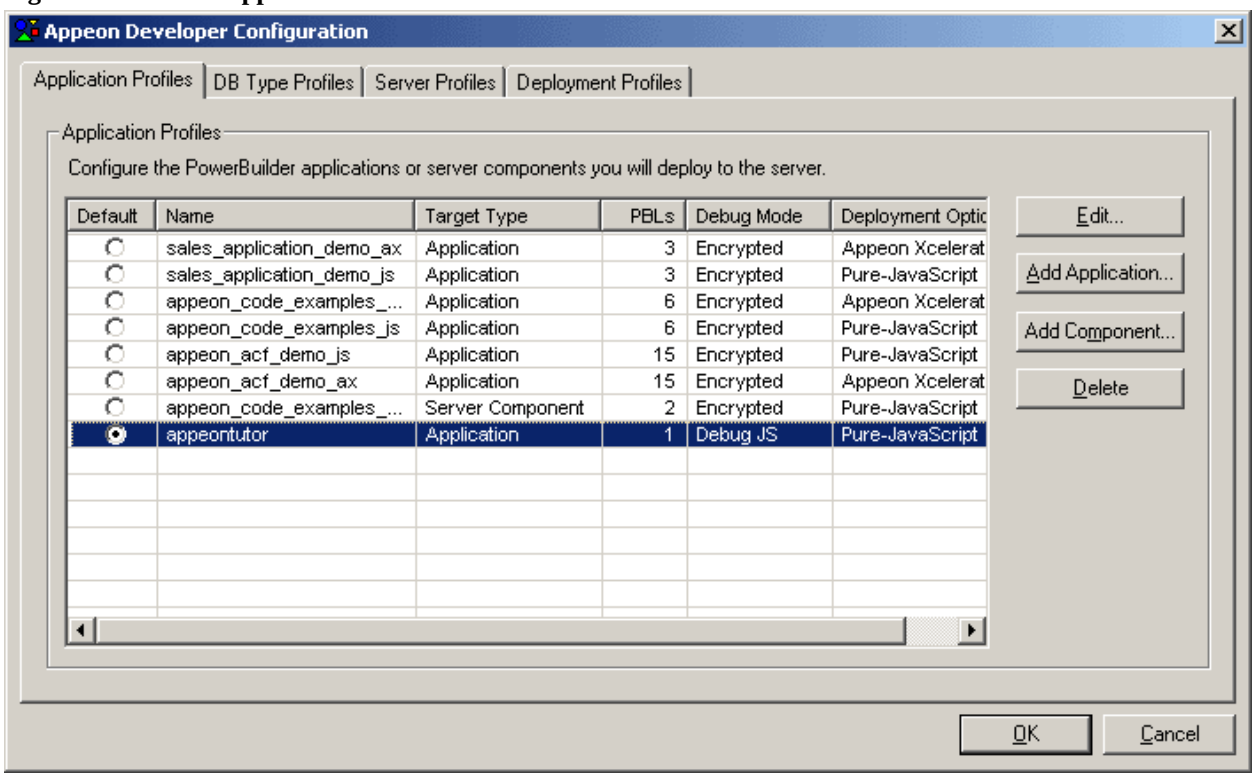
Figure 4-6: Test successful



STEP 7 – Click *OK* to return to the *Application Profiles* tab.

STEP 8 – Select *appeontutor* as the Default application, as shown in Figure 4-7.

Figure 4-7: Default application



STEP 9 – Click *OK* to close the Appeon Developer Configuration window.

4.2 Adding server profiles

A server profile is a set of parameters for Appeon Developer to connect to, and deploy applications to, a particular Application Server or Web Server. In this tutorial, you will use Appeon Server as both the Application Server and the Web Server.

Start Appeon Server before adding a profile. You will need to verify the connection to the Appeon Server during the configuration process.

4.2.1 Starting Appeon Server

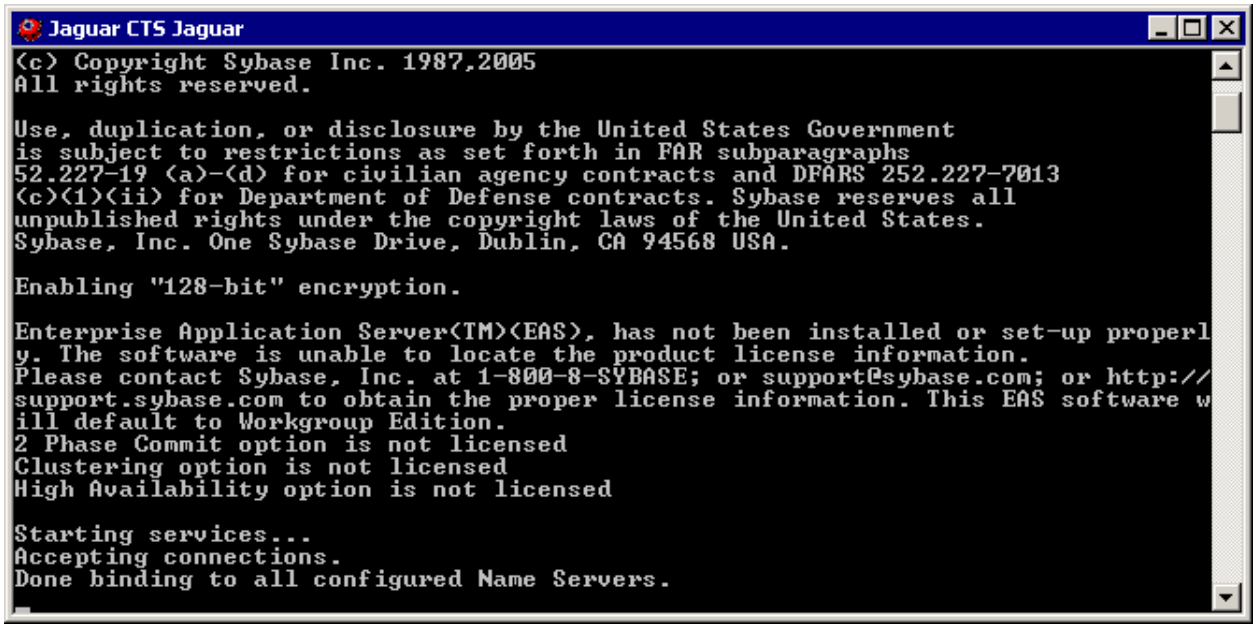
There are two ways to start Appeon Server: One is to start the EAServer that Appeon Server is installed on from a Windows shortcut, and the other is to start Appeon Server as a service. Appeon recommends that you always start Appeon Server through EAServer from the Windows shortcut. If you are using Appeon Server and starting it as a service, stop or

uninstall the service. For more information on how to stop or uninstall the EAServer service, refer to the *EAServer System Administration Guide*.

To start Appeon Server from a windows shortcut:

In Windows: select *Start* | *Programs* | *Appeon 3.1 for PowerBuilder* | *Jaguar Server*. Wait until the "Accepting connections" text line is displayed, as shown in Figure 4-8. This indicates that Appeon Server is ready for use.

Figure 4-8: Appeon Server



4.2.2 Adding an Appeon Server profile

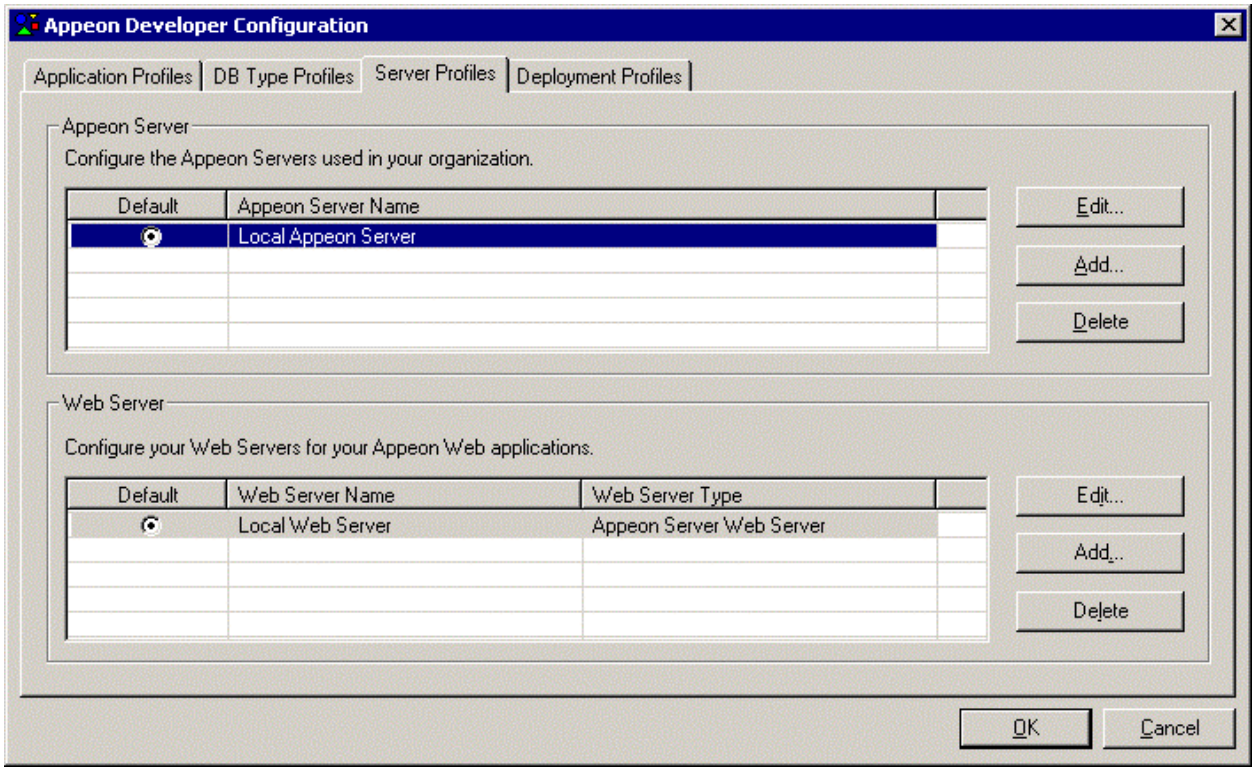
Each Appeon Server profile contains settings of an Appeon Server/application server used by the Appeon deployment.

To add the Appeon Server profile for the deployment of the tutorial application:

STEP 1 – Click the *Configure* button () in the Appeon Developer toolbar.

STEP 2 – Select the *Server Profiles* tab, as shown in Figure 4-9.

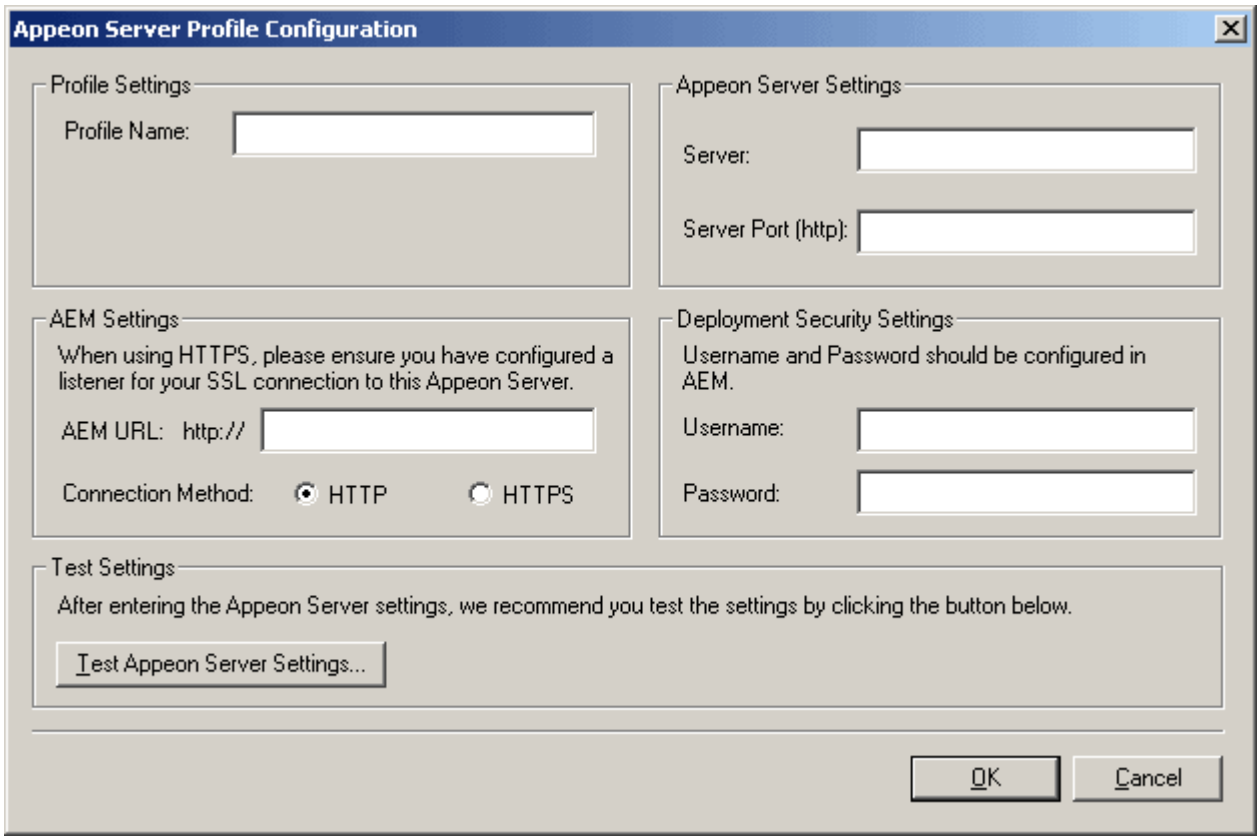
Figure 4-9: Server Profiles



STEP 3 – Click *Add* in the Appeon Server group box.

The Appeon Server Profile Configuration dialog box appears, as shown in Figure 4-10.

Figure 4-10: Appeon Server Profiles



STEP 4 – Specify the following settings for an Appeon Server profile, as shown in Table 4-3.

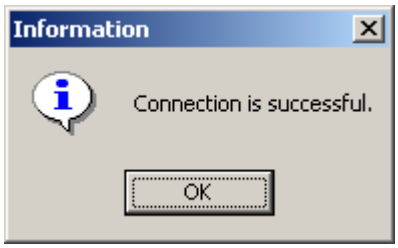
Table 4-3: Appeon Server Profile settings

In this field…	You should…
Profile Name	Type *Appeon Server Tutor*
Server	Type *localhost*
Server Port	Type *9988*
Deployment Username	Not necessary to specify this setting if the Deployment Security is off (by default) in AEM.
Deployment Password	Not necessary to specify this setting if the Deployment Security is off (by default) in AEM.
AEM URL	AEM URL will be automatically generated after you specify the Server and Server Port (for example, *localhost:9988/AEM*).

STEP 5 – Click *Test Appeon Server Settings*.

Appeon Server must be running at this time. Appeon Developer will try to connect to Appeon Server with the parameters you specified. Make sure the test is successful before you continue, as shown in Figure 4-11.

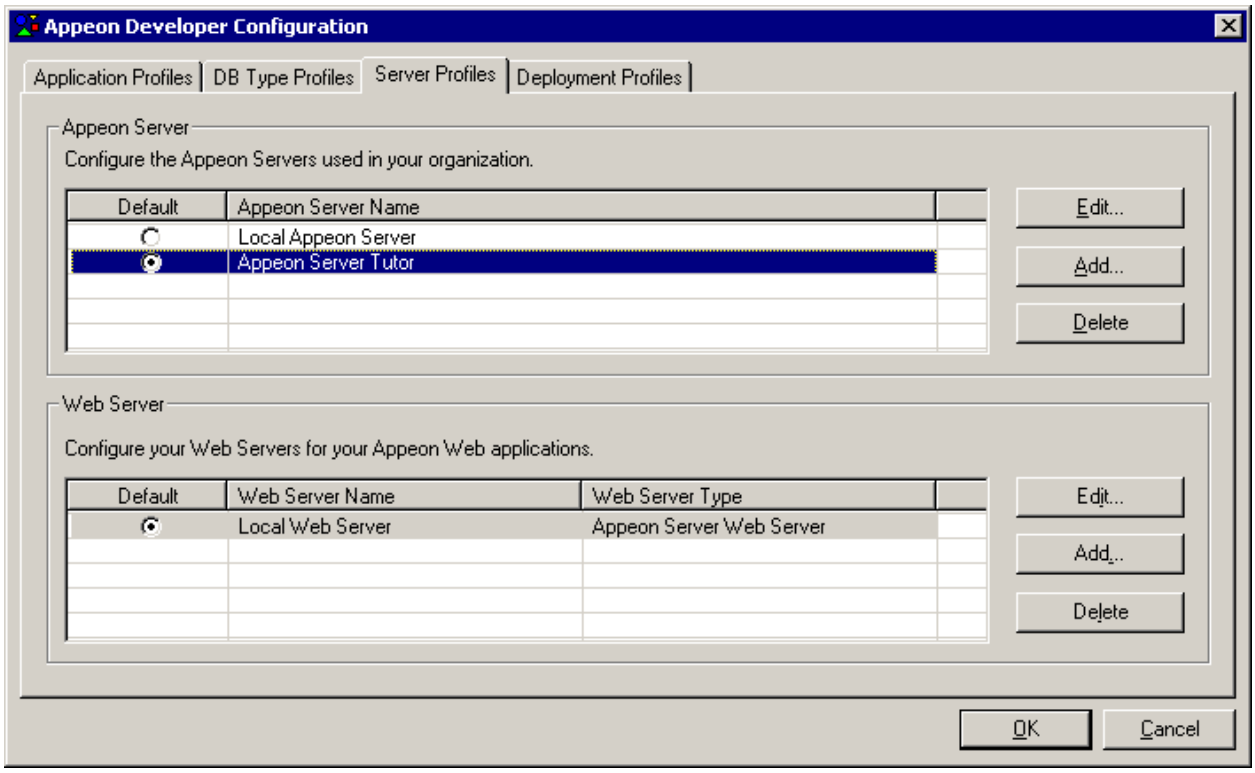
Figure 4-11: Test connection



STEP 6 – Click *OK* to return to the *Server Profiles* tab.

STEP 7 – Select the *Appeon Server Tutor* as the default server, as shown in Figure 4-12.

Figure 4-12: Default server



STEP 8 – Click *OK* to close the Appeon Developer Configuration window.

4.2.3 Adding a Web Server profile

Each Web Server profile contains settings of a Web server used by the Appeon deployment.

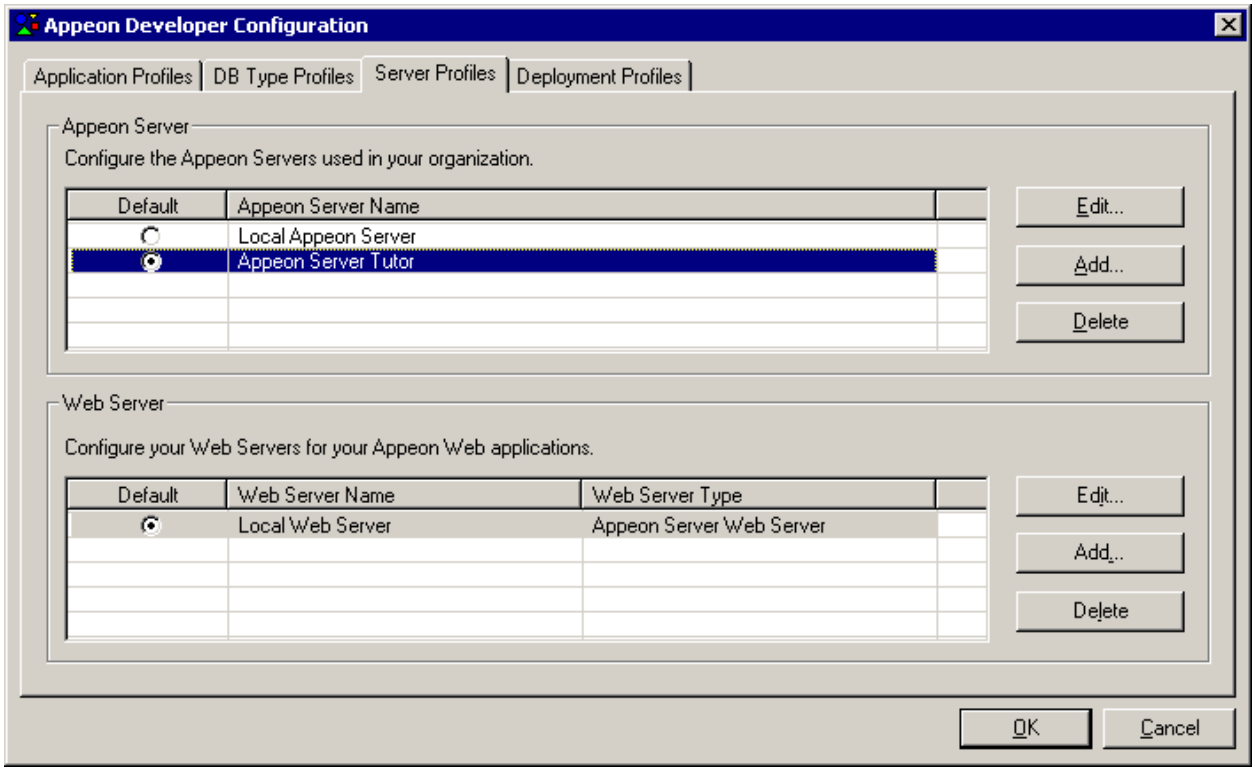
Start Web server before adding a Web Server profile. The Appeon Server Web server will be used for the deployment of the Appeon tutorial PowerBuilder application. You will need to verify the connection to the Web server during the Web Server profile configuration.

To add the Web Server profile for the deployment of the tutorial application:

STEP 1 – Click the *Configure* button ( ) in the Appeon Developer toolbar.

STEP 2 – Select the *Server Profiles* tab, as shown in Figure 4-13.

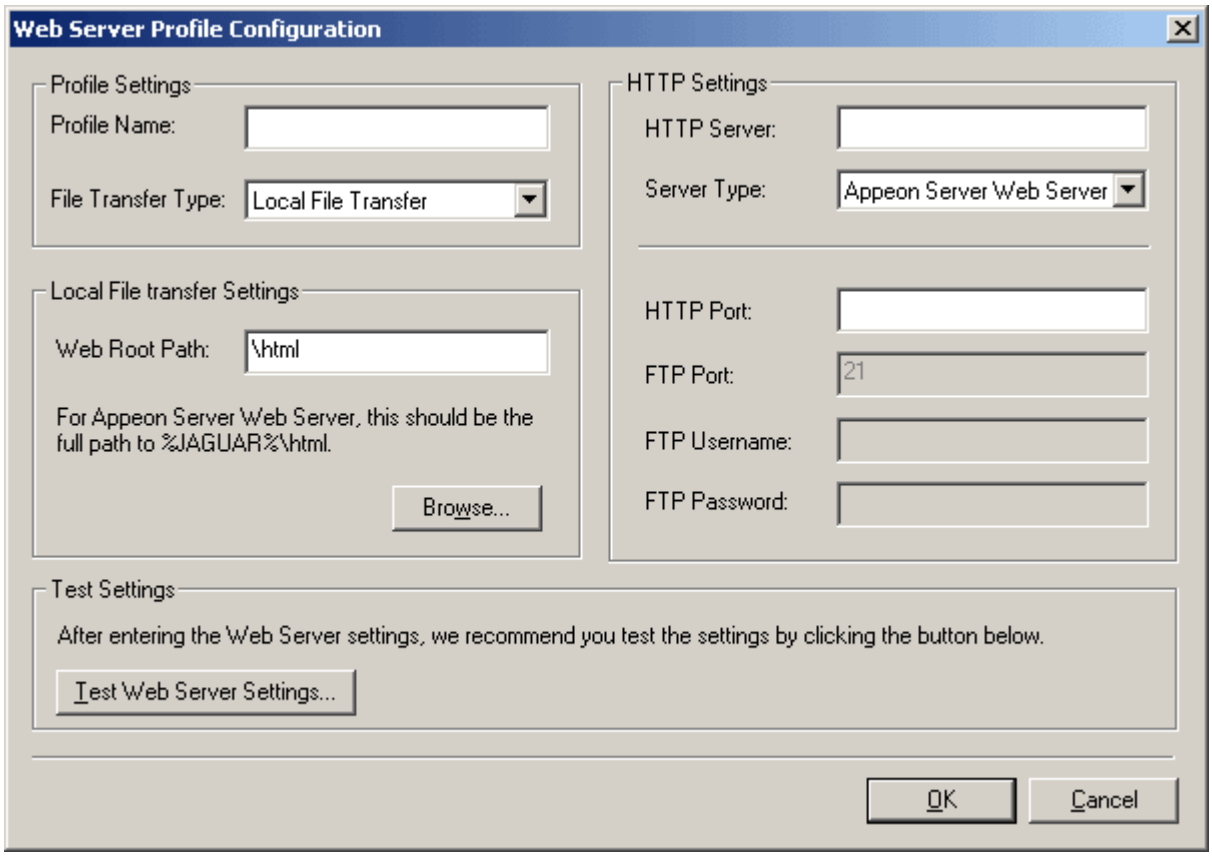
Figure 4-13: Server Profiles



STEP 3 – Click *Add* in the Web Server group box.

The Web Server Profile Configuration dialog box appears, as shown in Figure 4-14.

Figure 4-14: Web Server Profile Configuration window



STEP 4 – Specify the necessary settings for a Web Server profile, as shown in Table 4-4.

Table 4-4: Web Server Profile settings

In this field	You should
Profile Name	Type *Web Server Tutor*
File Transfer Type	Select *Local File Transfer* from the dropdown listbox.
HTTP Server	Type *localhost*
Server Type	Select *Appeon Server Web Server* from the dropdown listbox.
HTTP Port	Type *9988*
Web Root Path	Click *Browse* to navigate to *%JAGUAR%\html*, where *%JAGUAR%* indicates the EAServer installation directory (for example: C:\Program Files\Sybase\EAServer\html).

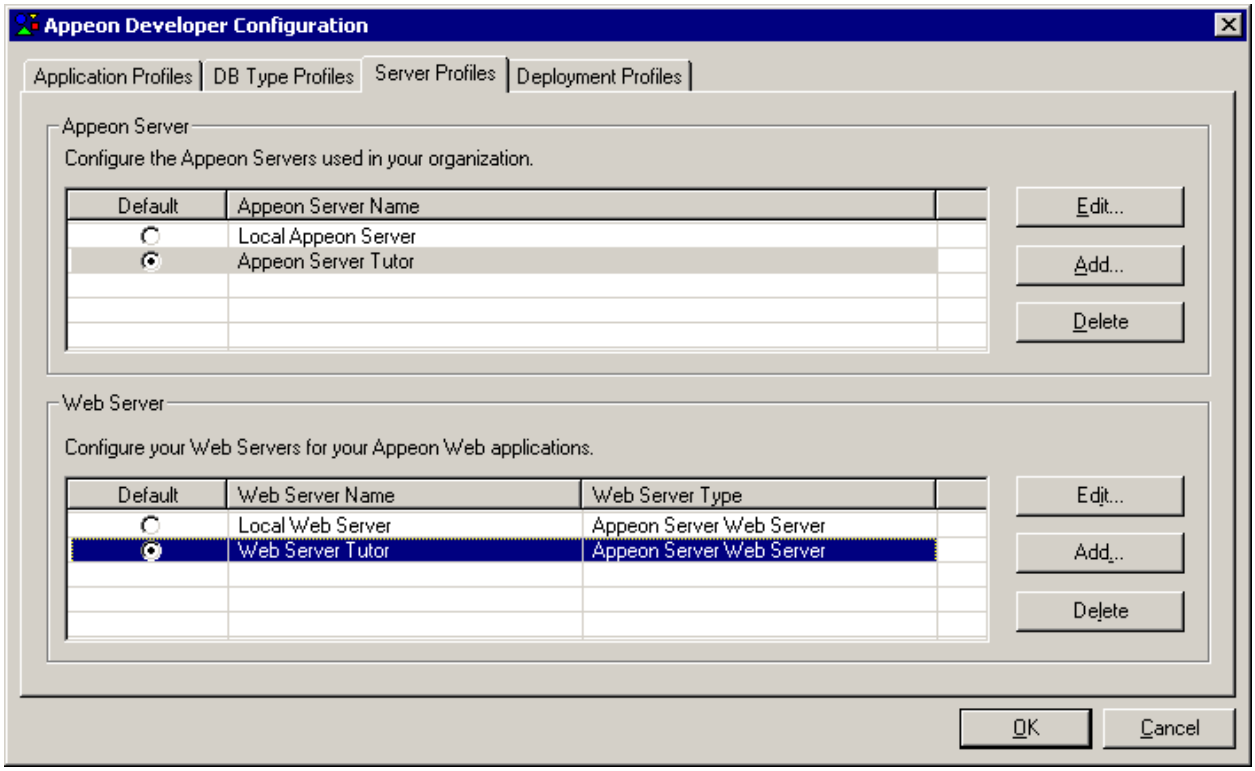
STEP 5 – Click *Test Web Server Settings*.

Web Server must be running at this time. Appeon Developer will try to connect to the Web Server with the IP (localhost) and port (9988). Make sure the test is successful before you continue.

STEP 6 – Click *OK* to return to the *Server Profiles* tab.

STEP 7 – Select *Web Server Tutor* as the default server, as shown in Figure 4-15.

Figure 4-15: Default server



STEP 8 – Click *OK* to close the Appeon Developer Configuration box.

4.3 Adding a deployment profile

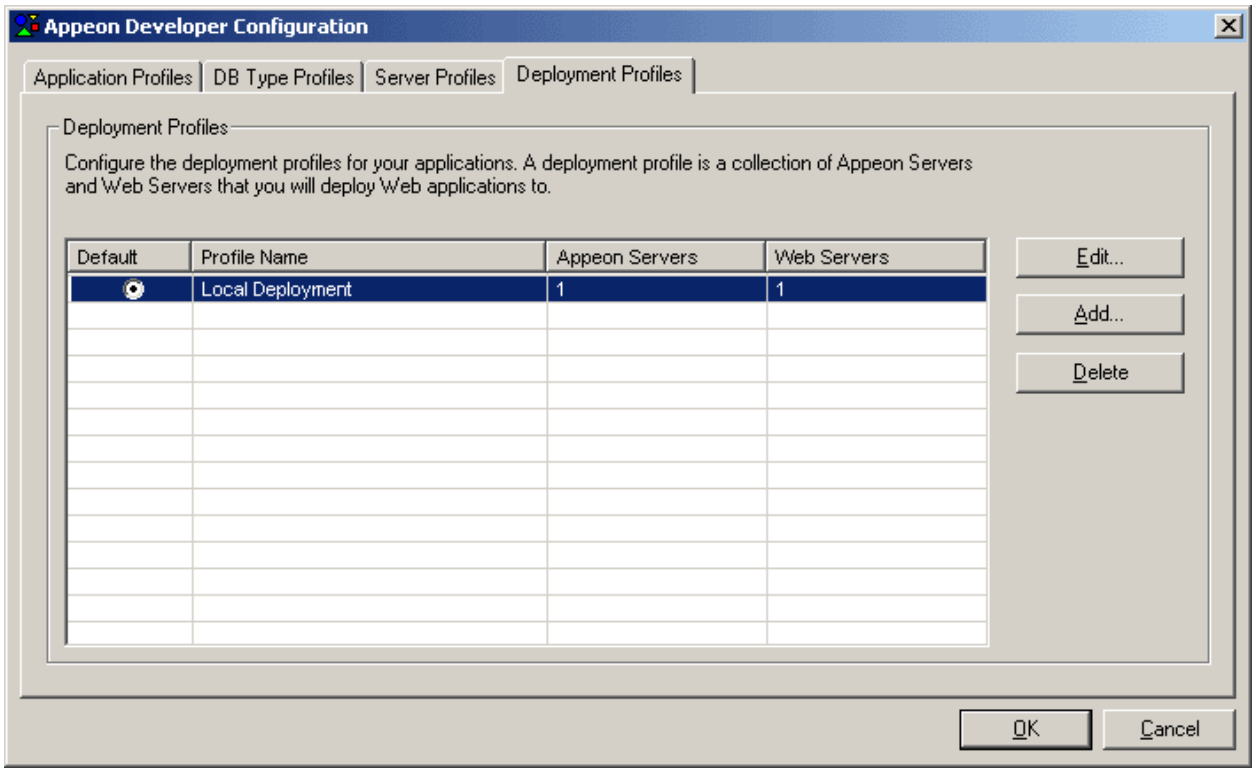
A deployment profile associates specified Appeon Server(s) and Web Server(s) as a group used for Web deployment.

To add the deployment profile used by the tutorial application:

STEP 1 – Click the *Configure* button () in the Appeon Developer toolbar.

STEP 2 – Select the *Deployment Profiles* tab, as shown in Figure 4-16.

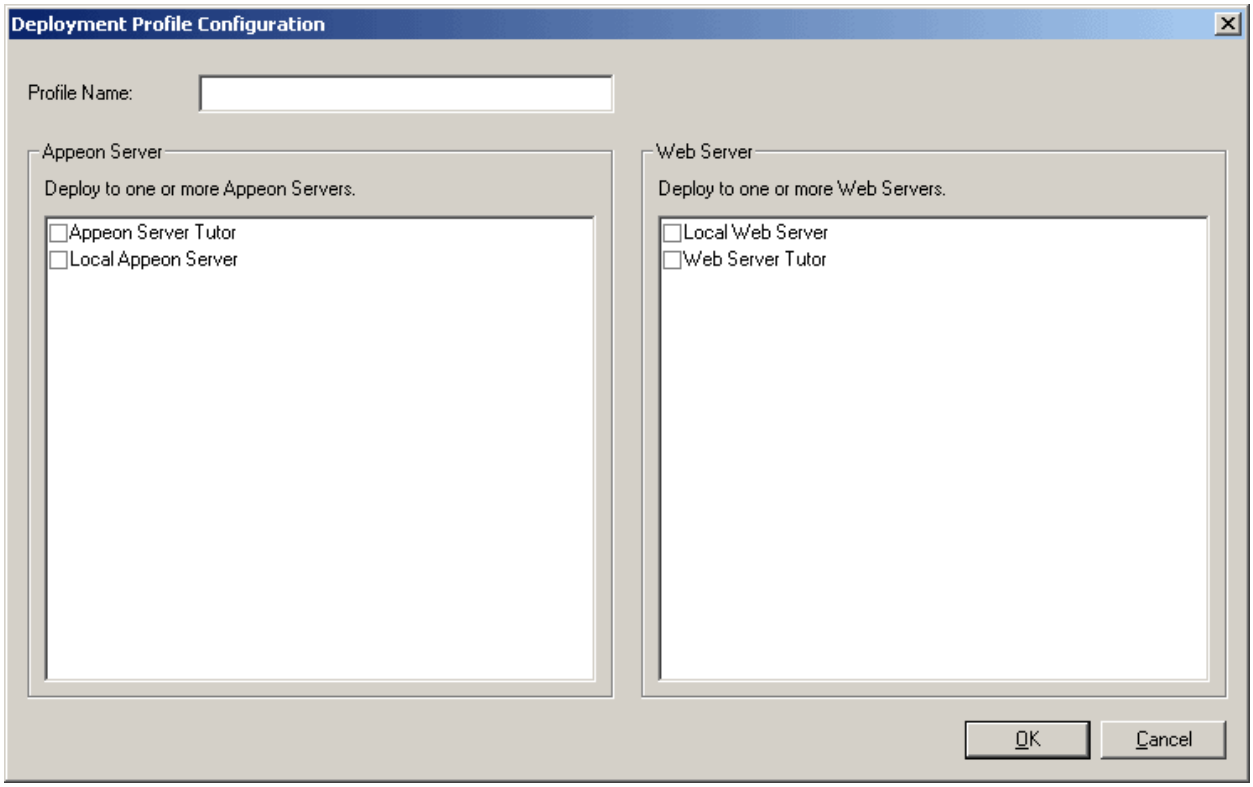
Figure 4-16: Deployment Profile



STEP 3 – Click *Add* under the *Deployment Profiles* tab.

The Deployment Profile Configuration window is displayed, as shown in Figure 4-17.

Figure 4-17: Deployment Profile Configuration window

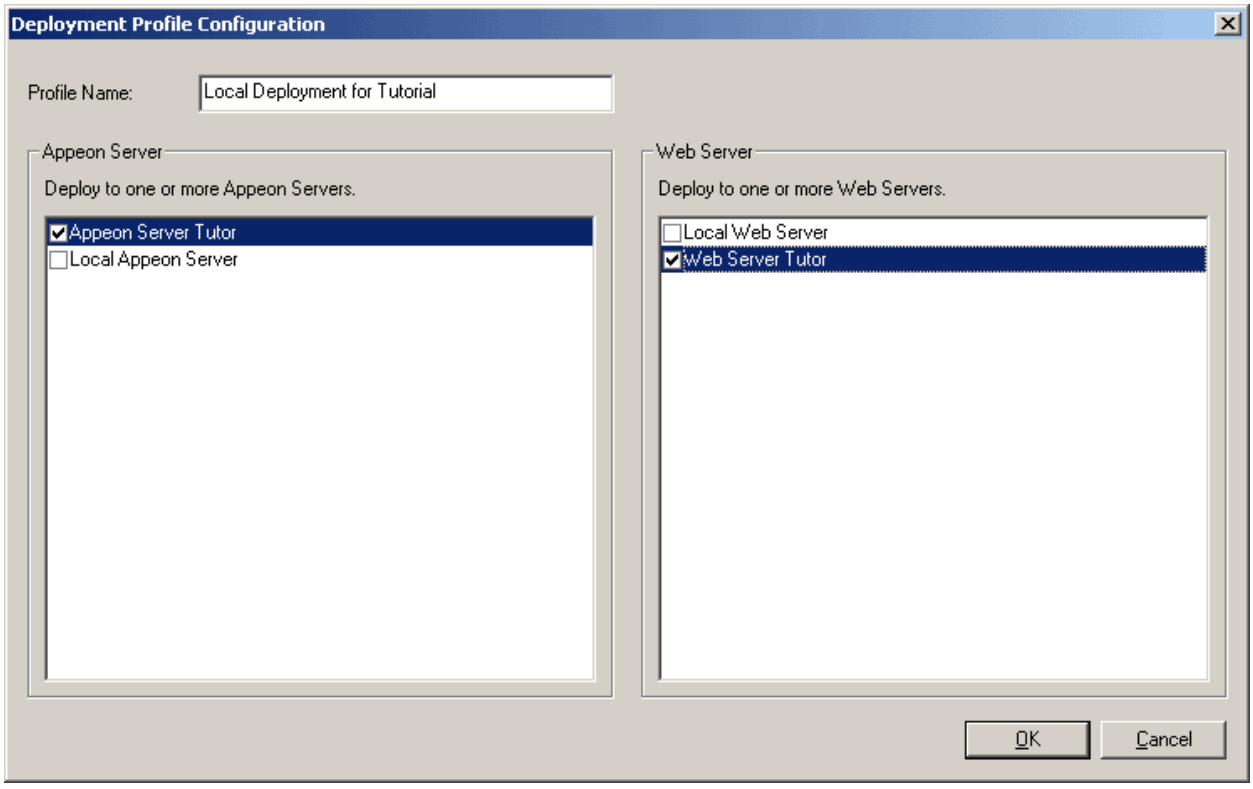


STEP 4 – Type *Local Deployment for Tutorial* in the Profile Name text box.

STEP 5 – Select *Appeon Server Tutor* in the Appeon Server list, as shown in Figure 4-18.

STEP 6 – Select *Web Server Tutor* in the Web Server list, as shown in Figure 4-18.

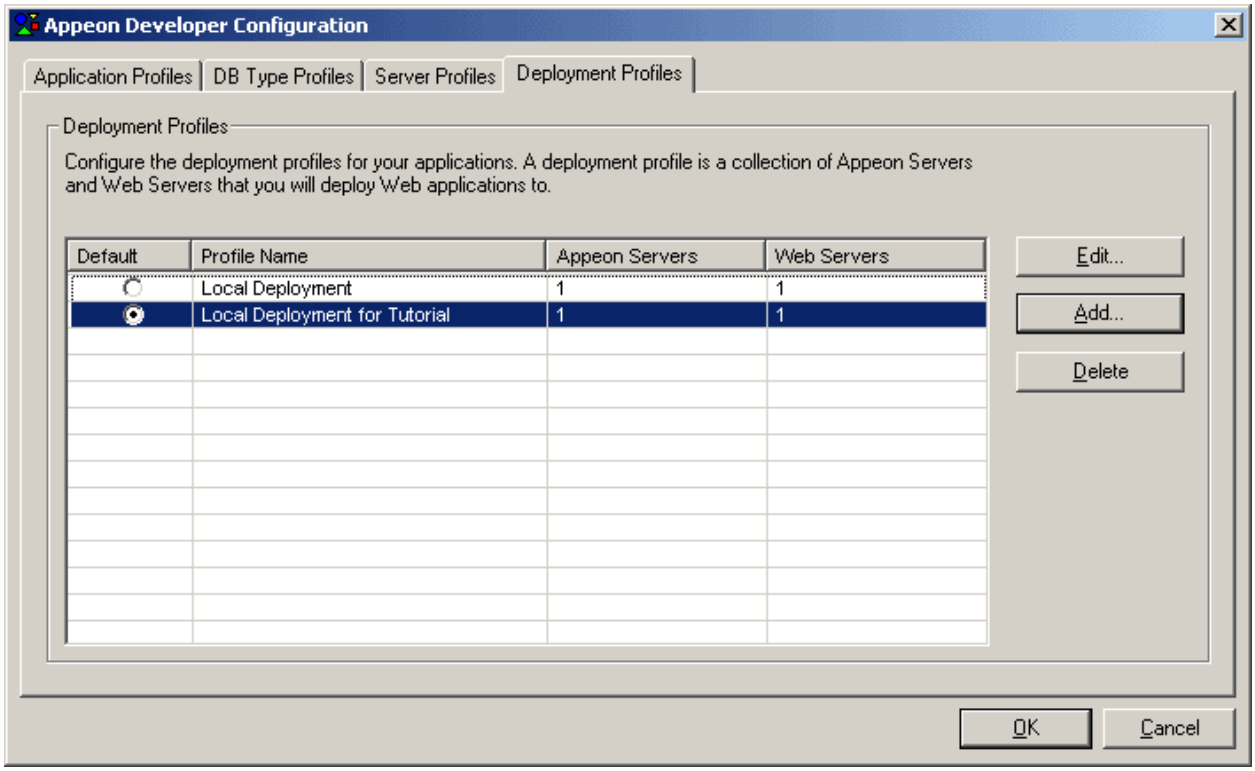
Figure 4-18: Deployment Profile settings



STEP 7 – Click *OK* to return to the *Deployment Profile* tab.

STEP 8 – Select *Local Deployment for Tutorial* as the default deployment profile, as shown in Figure 4-19.

Figure 4-19: Default deployment profile



STEP 9 – Click *OK* to close the Appeon Developer Configuration dialog box.

5 Modifying Unsupported Features

This chapter shows you how to remove unsupported features in the Appeon tutorial PowerBuilder application so that it can be completely converted without losing its functionality.

In this section, you will learn how to:

- Perform unsupported feature analysis on the tutorial application
- Place an unsupported feature in Appeon Server
- Remove an unsupported feature
- Work around an unsupported feature
- Optimize and full build the tutorial application

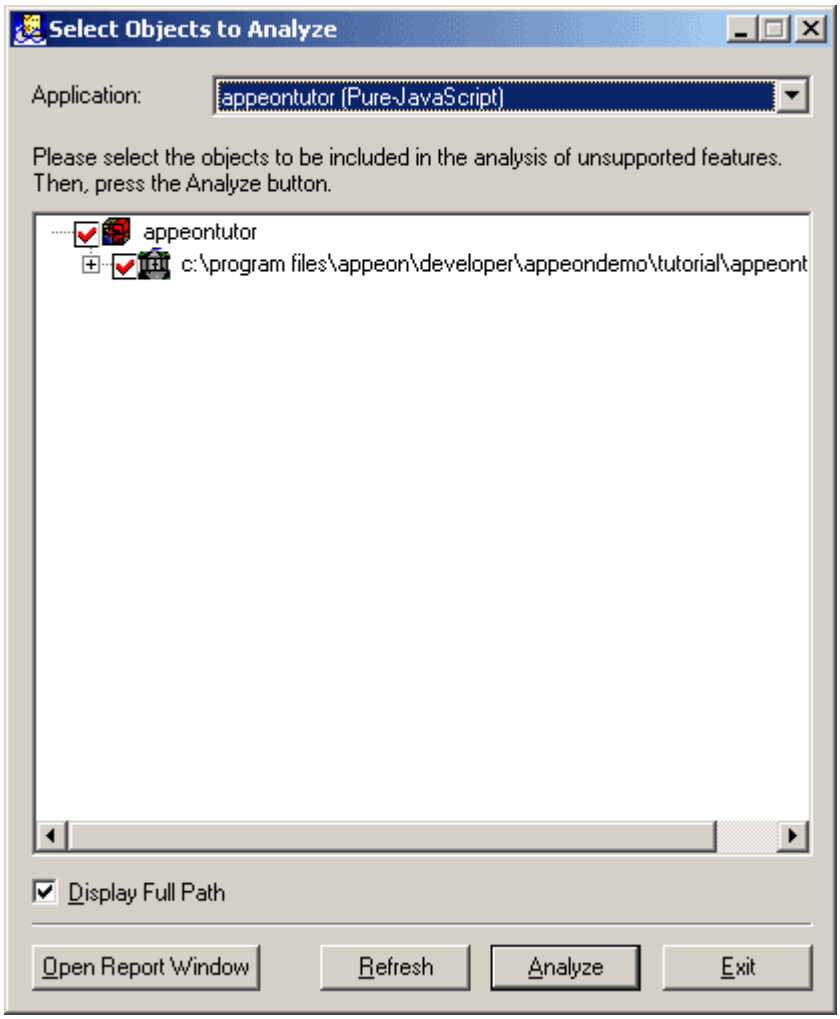
5.1 Unsupported features analysis

Appeon provides an unsupported feature analysis (UFA) tool to handle the Appeon unsupported source code in a PowerBuilder application.

To perform Unsupported Features Analysis to the tutorial application, perform the following steps:

STEP 1 – In the PowerBuilder IDE, click the *Analyze* button () on the Appeon Developer toolbar. The Select Objects to Analyze dialog box appears, as shown in Figure 5-1.

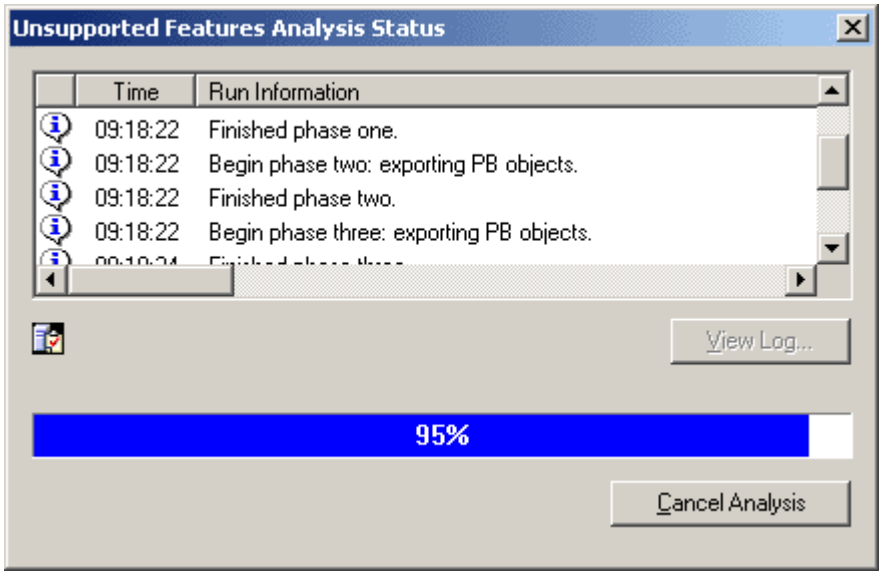
Figure 5-1: Select Objects to Analyze dialog box



STEP 2 – Click the *Analyze* button to analyze the entire application.

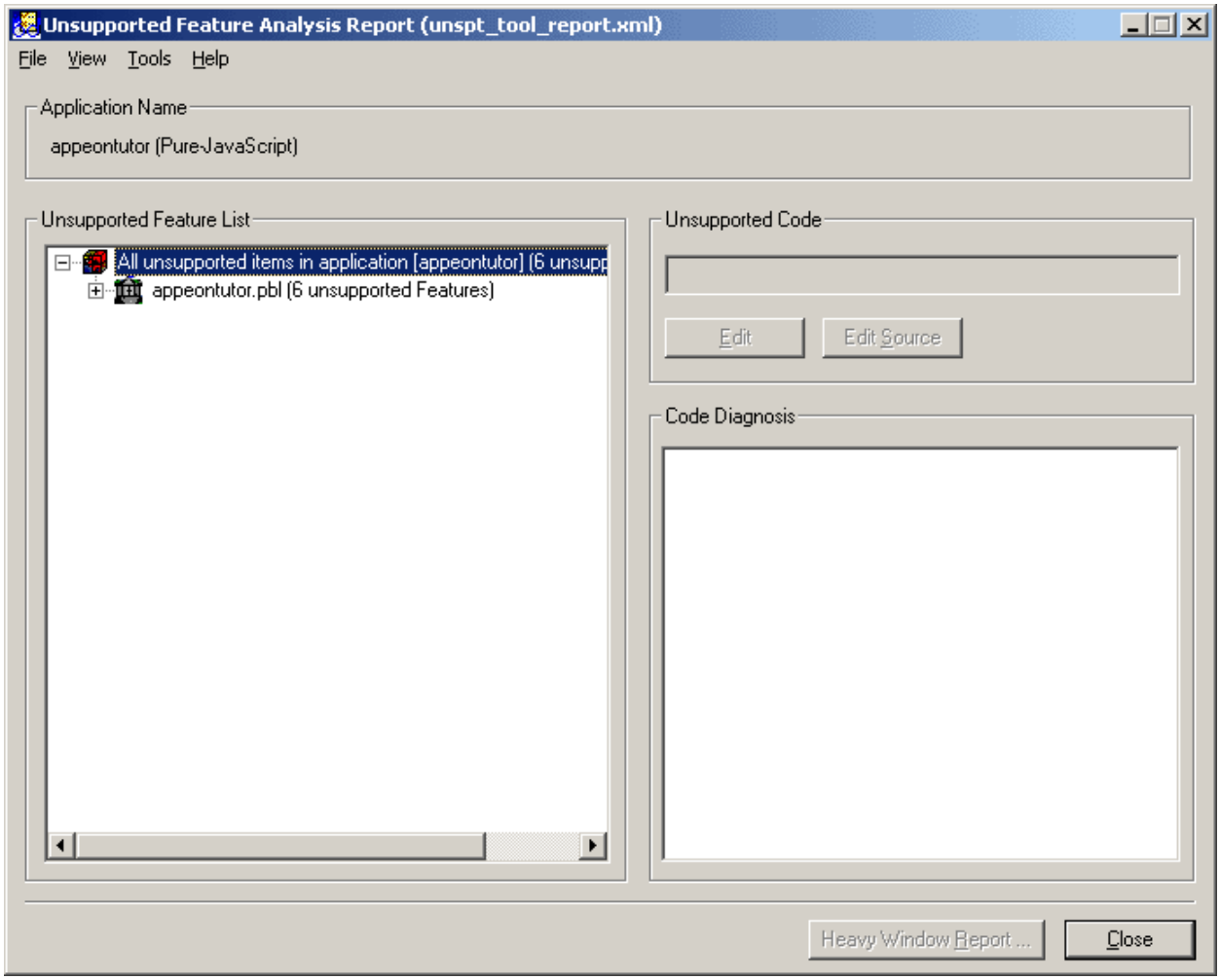
The Unsupported Features Analysis Status dialog box is displayed. This shows the progress of the Unsupported Features Analysis, as shown in Figure 5-2.

Figure 5-2: Unsupported Features Analysis Status



When the analysis is completed, the Unsupported Feature Analysis Report window (UFA Report Window) will be displayed, as shown in Figure 5-3. You can view all the unsupported features in the Unsupported Feature List treeview and modify them one by one.

Figure 5-3: UFA Report window



There are three different types of unsupported features residing in the tutorial PowerBuilder application:

- Using *Randomize* and *Rand* functions to randomly generate a number
- Setting the indicator for the current row in the DataWindow using the DataWindow *SetRowFocusIndicator* function
- Dragging and dropping between the DataWindow and the ListBox using the Drag n' Drop methods

Different modification methods are provided to modify these three different types of unsupported features. Refer to the following sections for detailed descriptions.

5.2 Modification instructions

5.2.1 Placing unsupported features in EAServer as NVOs

In this section, you will learn how to:

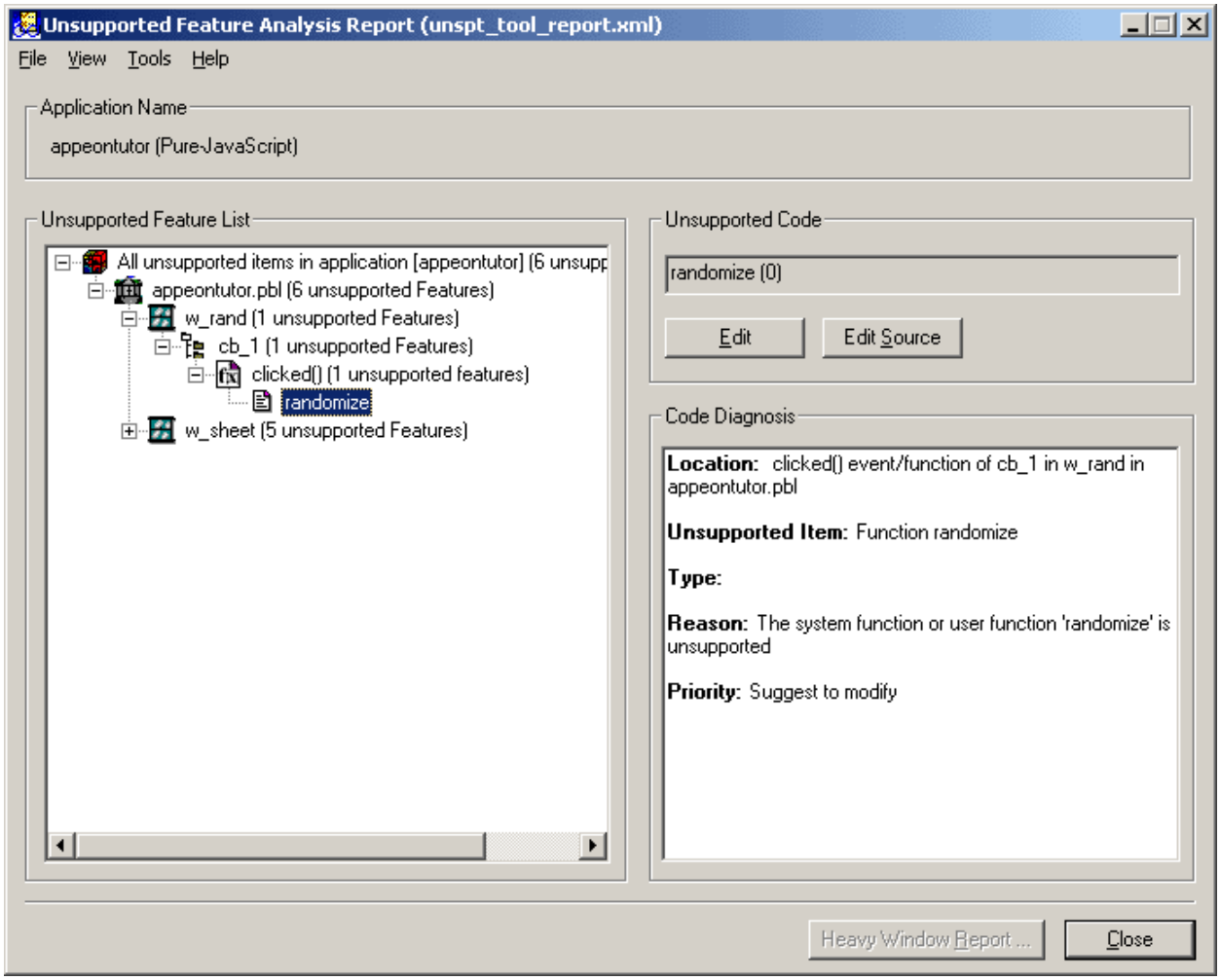
- Encapsulate the unsupported features into PowerBuilder NVO objects.

- Deploy them onto EAServer as NVOs.
- Modify the tutorial application to interact with the remote NVO components.

5.2.1.a Commenting out unsupported script

STEP 1 – In the UFA Report Window, expand *w_rand* | *cb_1* | *clicked* and select *randomize*, as shown in Figure 5-4.

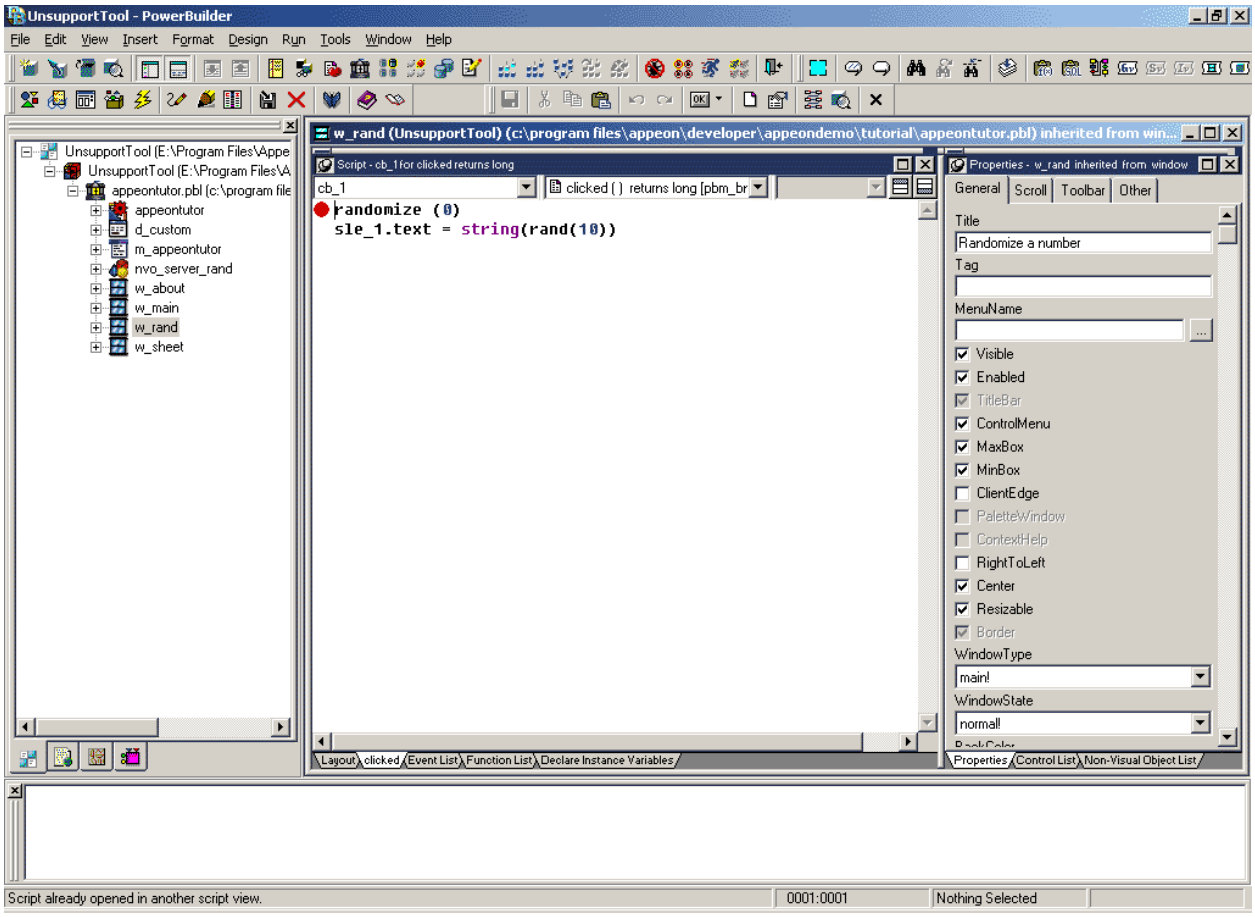
Figure 5-4: UFA Report Window



STEP 2 – Click the *Edit* button to enter the PowerBuilder Script view of the *Clicked* event of the *cd_1* CommandButton in *w_rand* Window.

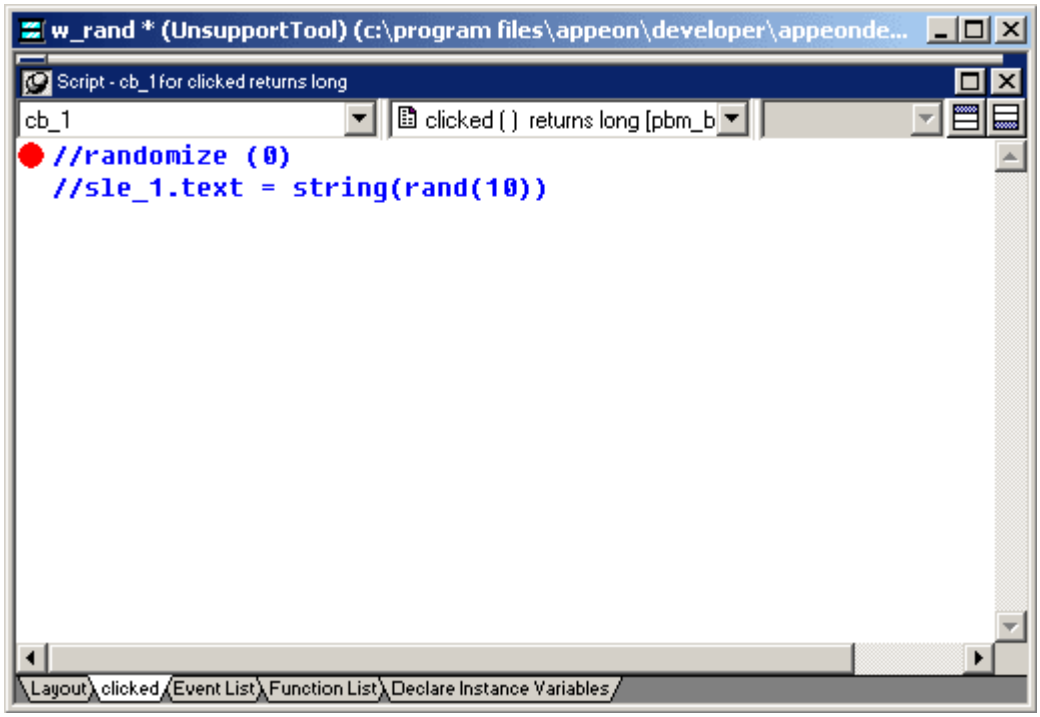
A red bullet is placed at the left side of the *randomize* function to indicate that this coding feature is unsupported by Appeon, as shown in Figure 5-5.

Figure 5-5: Unsupported code in the PowerBuilder painter



STEP 3 – Select the entire script of the *Clicked* event, and then click the *Comment* button () in PainterBar2.

Figure 5-6: Commenting unsupported script



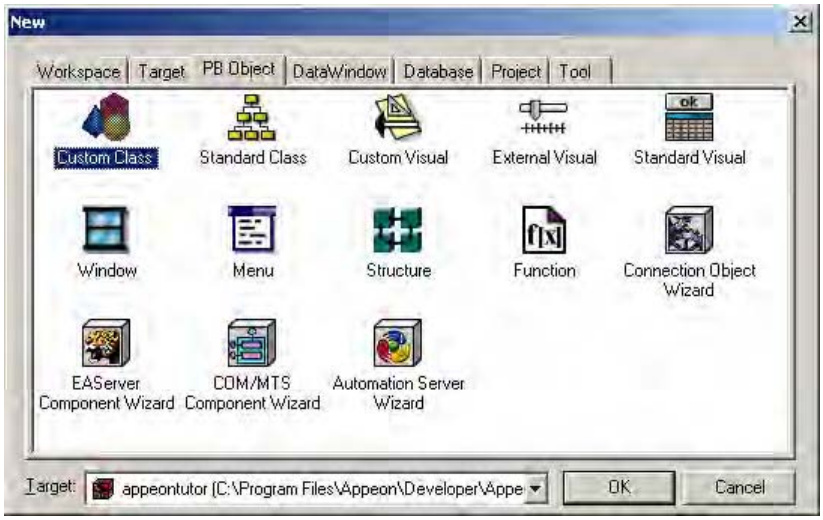
STEP 4 – Click the *Save* button, and then click the *Close* button in PainterBar1.

5.2.1.b Encapsulate unsupported features into NVOs

To encapsulate the unsupported features into NVOs, follow the steps detailed below:

STEP 1 – Choose *File | New* from the PowerBuilder menu. Then select the *Custom Class* icon under the PB Object tab in the New dialog box and click *OK*, as shown in Figure 5-7.

Figure 5-7: New

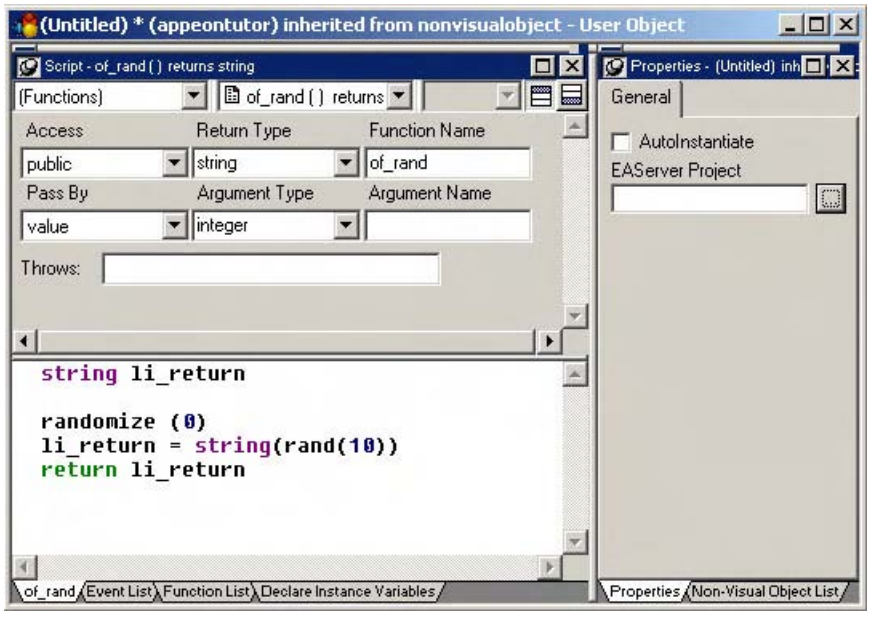


STEP 2 – The User Object painter for an untitled class user object will be displayed, as shown in Figure 5-8.

Define a new function: string *of_rand* and add the following code into the function:

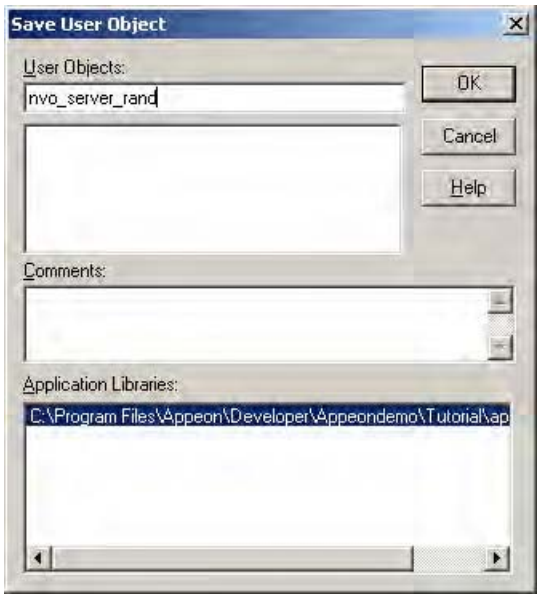
```
string li_return

randomize (0)
li_return = string(rand(10))
return li_return
```

Figure 5-8: User Object Painter



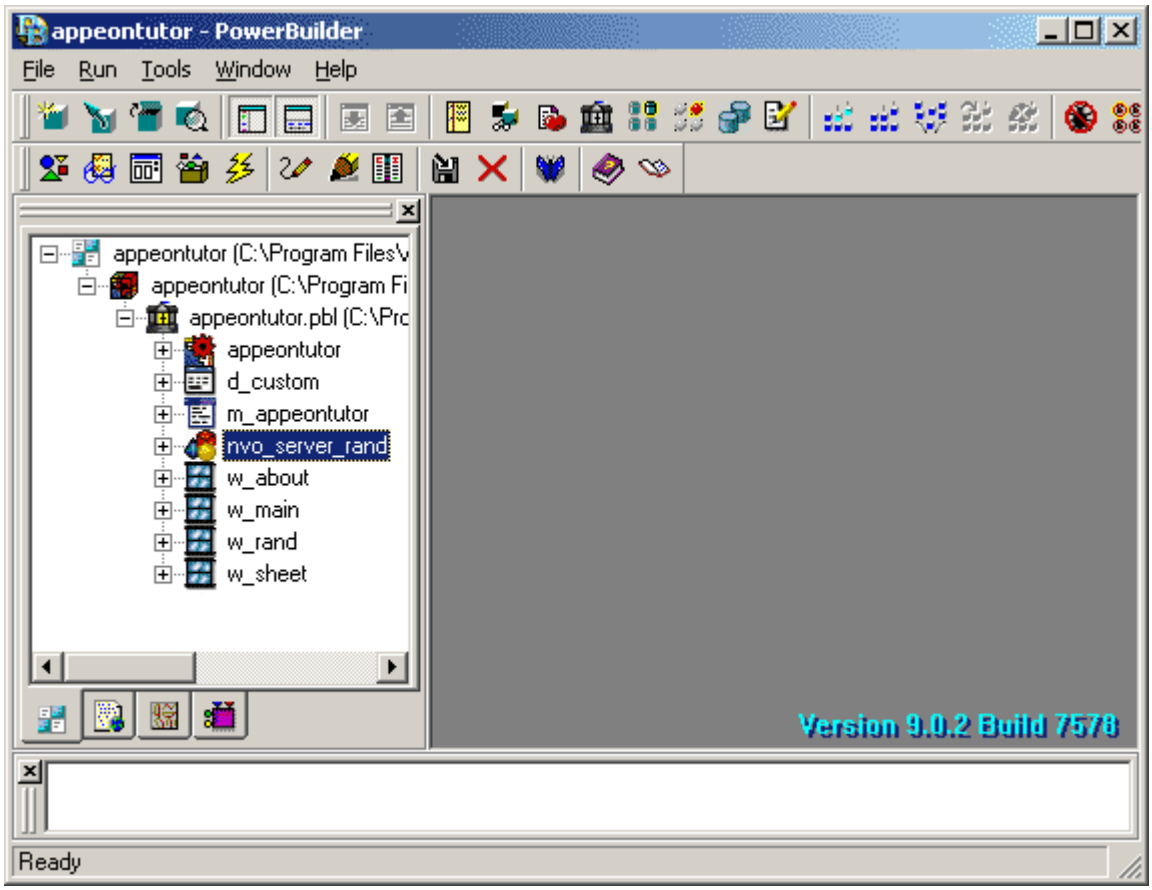
STEP 3 – Save the class user object using the name *nvo_server_rand*, as shown in Figure 5-9.

Figure 5-9: Save User Object



STEP 4 – Click *OK* and the *nvo_server_rand* user object will be added under the *appeontutor* application in the PowerBuilder system tree, as shown in Figure 5-10.

Figure 5-10: PowerBuilder system tree



5.2.1.c Adding EAServer profile in PowerBuilder

Use the PowerBuilder component wizard for deploying an NVO to Appeon Server. Before using the wizard, create an EAServer profile in PowerBuilder, which will connect to the EAServer that hosts Appeon Server.

An EAServer profile stores information on connection settings used to connect to EAServer. The profile you create is used by wizards that require a connection to EAServer.

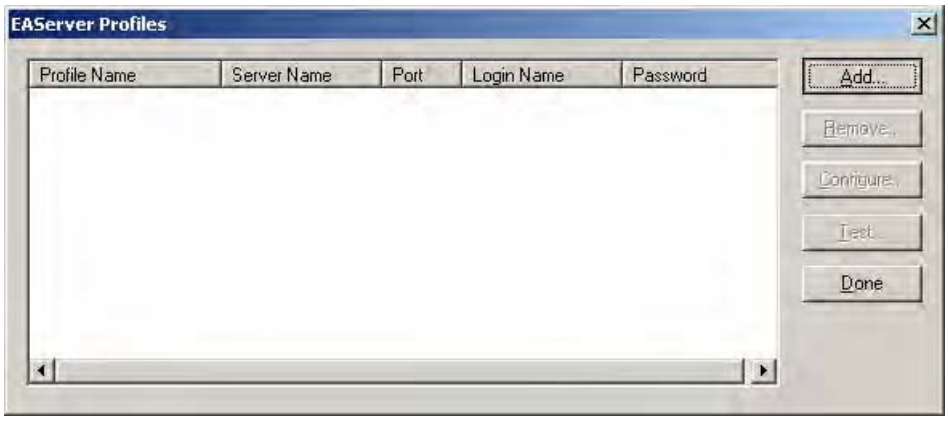
To add an EAServer profile to PowerBuilder:

STEP 1 – Start Appeon Server, if it is not already running. Make sure that Appeon Server is running during the following steps.

STEP 2 – Click the *EAServer Profile* button in PowerBar1 or choose *Tools* | *EAServer Profile* from the PowerBuilder menu.

Click *Add* on the EAServer Profiles dialog box that is displayed, as shown in Figure 5-11.

Figure 5-11: EAServer Profiles

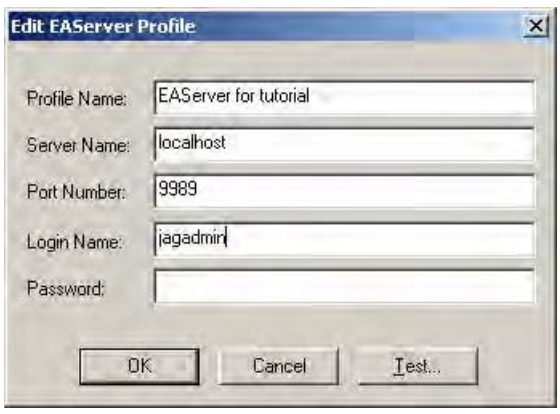


STEP 3 – The Edit EAServer Profile dialog box pops up, as shown in Figure 5-12. Type the information contained in Table 5-1 into each field.

Table 5-1: EAServer profile

In this field…	You should…
Profile Name	Type *EAServer for tutorial*
Server Name	Type *localhost*
Port Number	Type 9989
Login Name	The default login name is *jagadmin*. If you have changed the default, enter the correct login name.
Password	There is no password by default. If you have changed the default, enter the correct password.

Figure 5-12: Edit EAServer Profile



STEP 4 – Verify the EAServer profile by clicking the *Test* button on the EAServer Profile dialog box.

Note: Appeon Server must be running for the test to be successful.

Figure 5-13: EAServer Profiles

STEP 5 – Make sure that connection testing is successful.

5.2.1.d Deploy NVOs to EAServer

In the previous steps, an NVO was created and an EAServer profile configured. Now deploy the NVO to the EAServer specified in the profile. Use the EAServer Component Wizard to deploy EAServer components from PowerBuilder.

To deploy the NVO into EAServer:

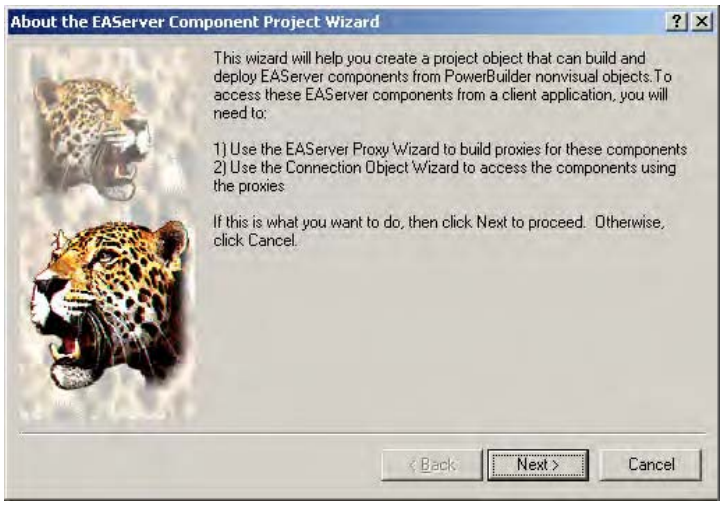
STEP 1 – Start Appeon Server, if it is not already started.

STEP 2 – Choose *File | New* from the PowerBuilder menu. Then select the *EAServer Component Wizard* icon under the Project tab in the New dialog box, and click *OK*, as shown in Figure 5-14.

Figure 5-14: New

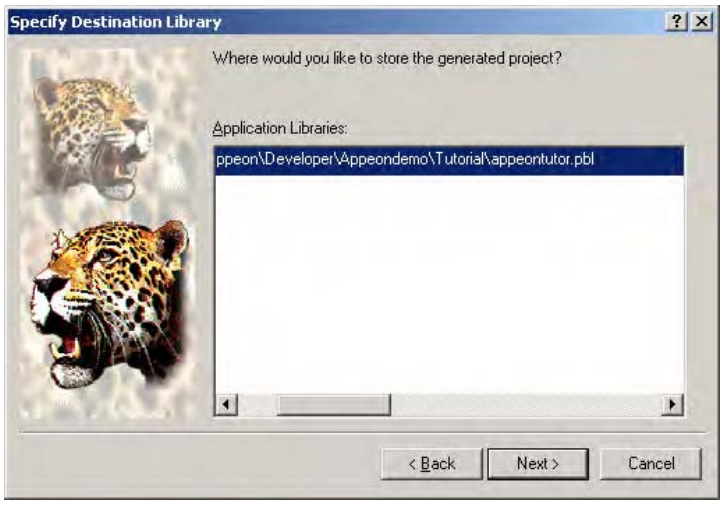
STEP 3 – The EAServer Component Project Wizard starts. Click *Next*, as shown in Figure 5-15.

Figure 5-15: About the EAServer Component Project Wizard



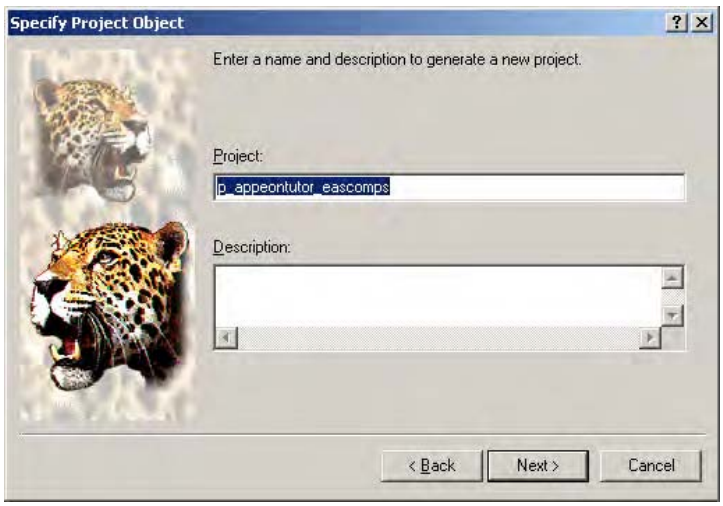
STEP 4 – The project will be stored in the *appeontutor.pbl* file, as shown in Figure 5-16. Click *Next*.

Figure 5-16: Specify Destination Library



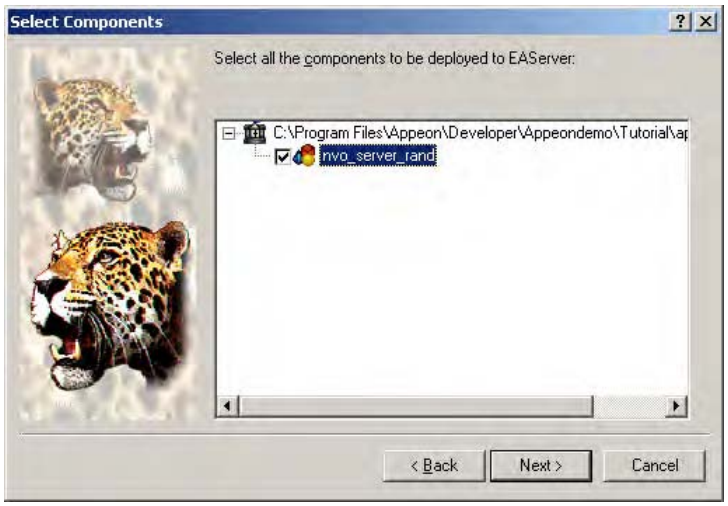
STEP 5 – Leave the project name as *p_appeontutor_eascomps*, as shown in Figure 5-17. Click *Next*.

Figure 5-17: Specify Project Object



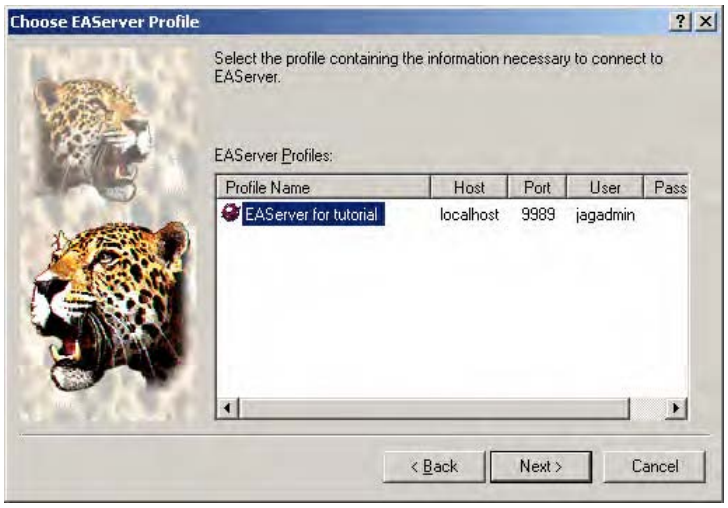
STEP 6 – Select the *nvo_server_rand* NVO that is to be deployed to EAServer, as shown in Figure 5-18. Click *Next*.

Figure 5-18: Select Components



STEP 7 – Select the *EAServer for tutorial* profile, as shown in Figure 5-19. You created this EAServer profile in a previous step. If you have set up profiles outside of the tutorial for other EAServers, there will be more profiles in the list.

Figure 5-19: Choose EAServer Profile

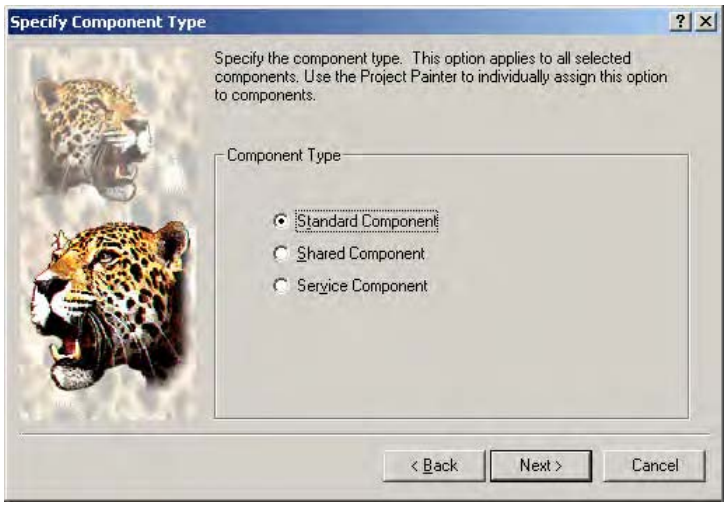


STEP 8 – Type *n_rand* into the Package Name dropdown listbox as shown in Figure 5-20. Click *Next*.

Figure 5-20: Specify Package Name

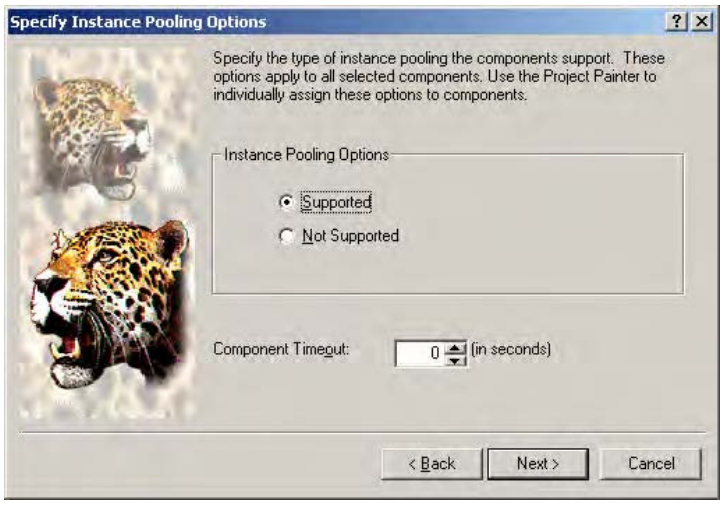
STEP 9 – Select *Standard Component* on the Specify Component Type dialog box, as shown in Figure 5-21. Click *Next*.

Figure 5-21: Specify Component Type



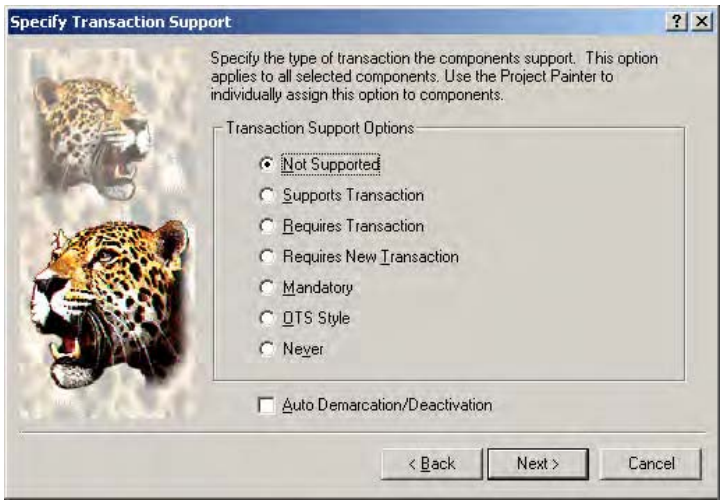
STEP 10 – Leave the *Supported* option checked for the *Instance Pooling Options* field, as shown in Figure 5-22. Click *Next*.

Figure 5-22: Specify Instance Pooling Options



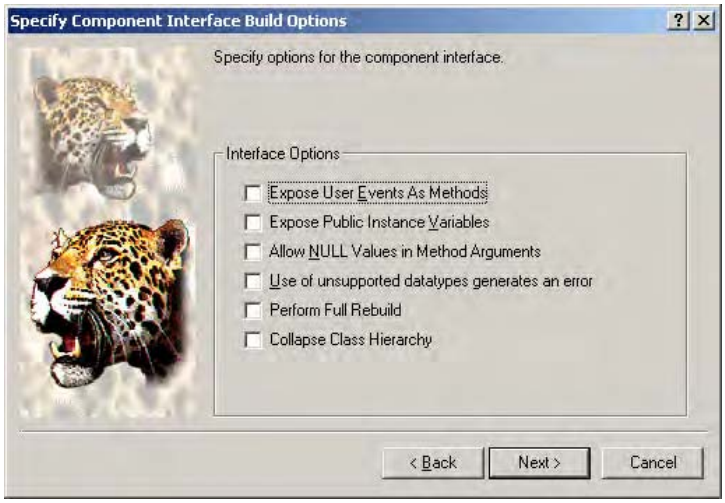
STEP 11 – In the *Transaction Support Options* field leave the default option as *Not Supported*, as shown in Figure 5-23. Click *Next*.

Figure 5-23: Specify Transaction Support



STEP 12 – Leave all options at default, as shown in Figure 5-24. Click *Next*.

Figure 5-24: Specify Component Interface Build Options

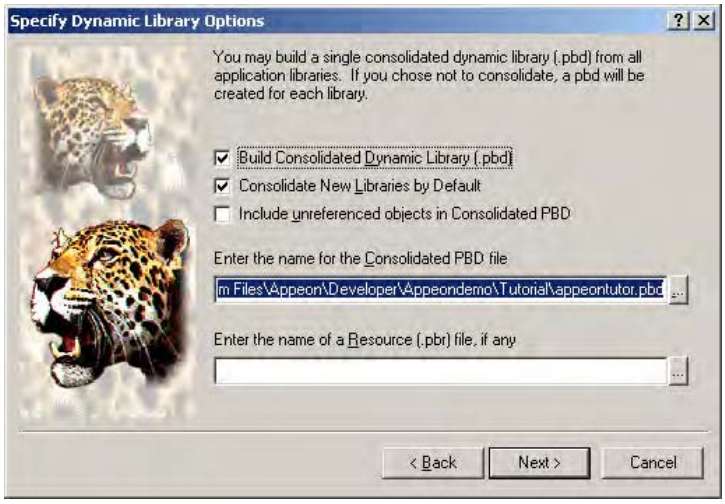


STEP 13 – Leave all options at default, as shown in Figure 5-25. Click *Next*.

Figure 5-25: Specify Other Component Options

STEP 14 – Leave all options at default, as shown in Figure 5-26. Click *Next*.

Figure 5-26: Specify Dynamic Library Options



STEP 15 – The wizard now comes to its final stage. Leave all options at default, as shown in Figure 5-27. Click *Finish*.

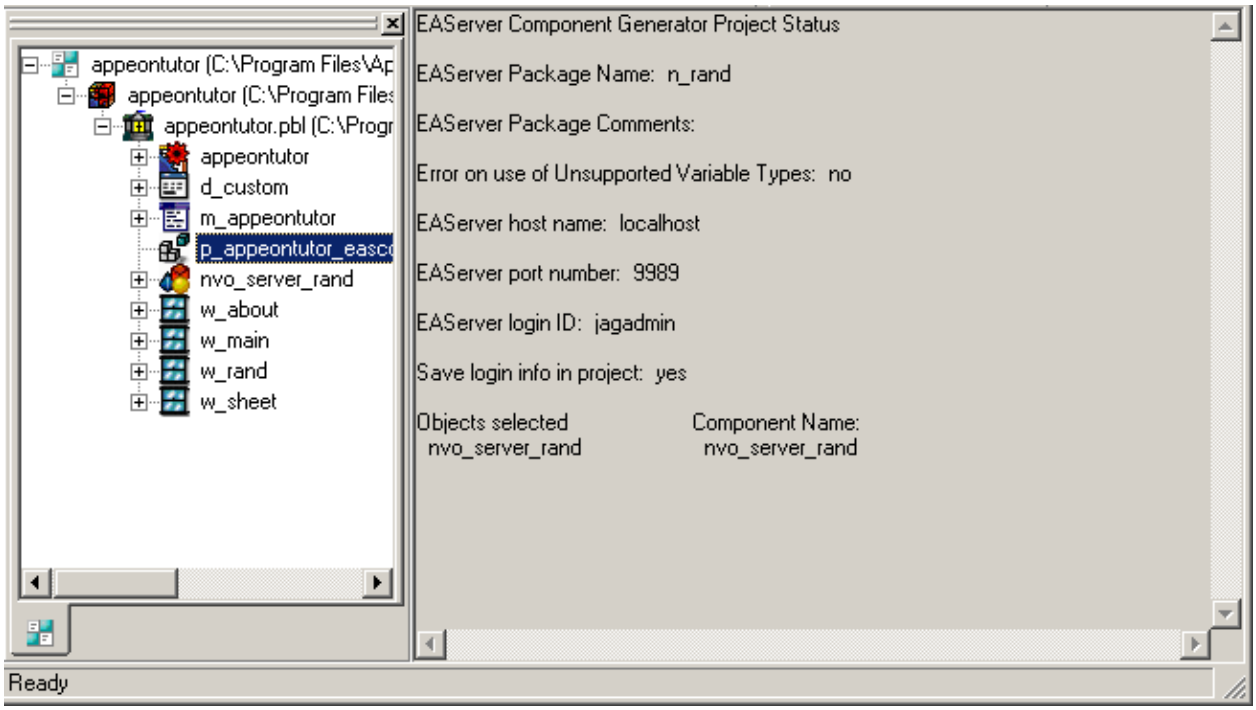
Figure 5-27: Ready to Create EAServer Component

STEP 16 – The wizard now creates a component project. The Project Painter for the component opens automatically in PowerBuilder, and the project name (p_appeontutor_eascomps) appears in the left system tree list, as shown in Figure 5-28.

Click the *Deploy* button in PowerBuilder PainterBar1, or choose *Design* | *Deploy Project* from the PowerBuilder menu to start deployment of the component.

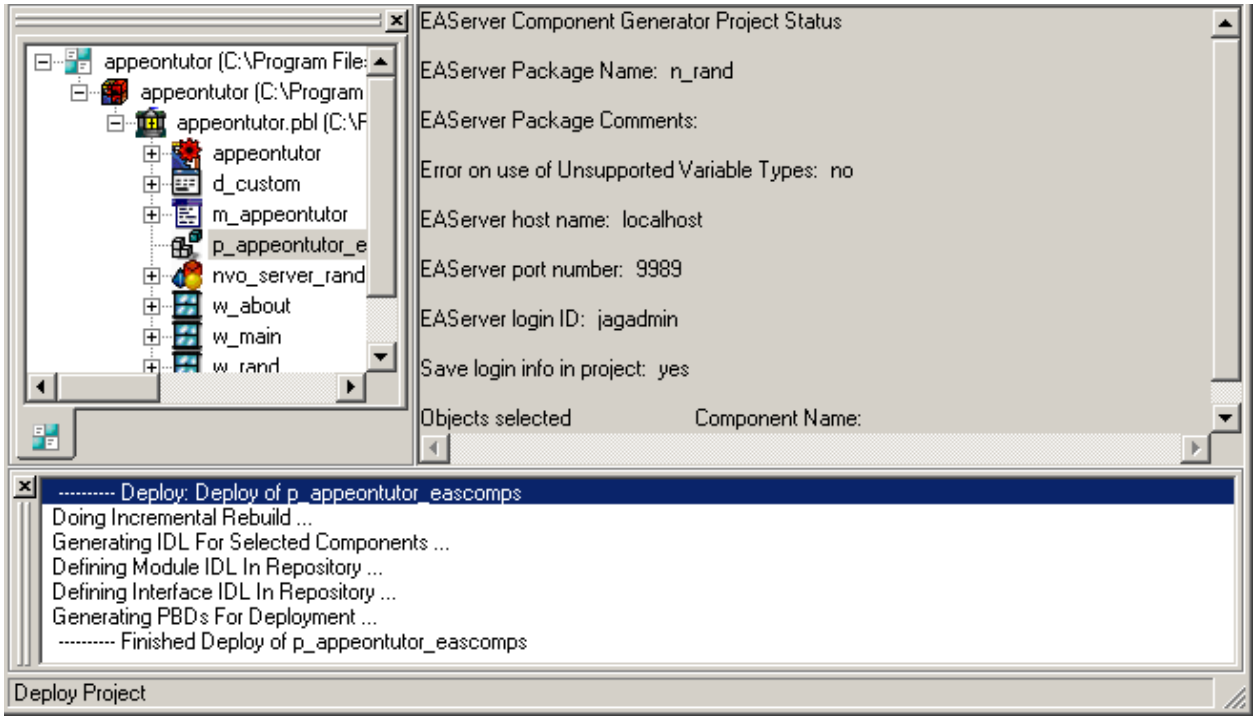
Note: Ensure that Appeon Server is running.

Figure 5-28: New Component project created



STEP 17 – The deployment starts and relevant information is displayed in the Output window. When the process is complete, the Output window will display "Finished Deploy of p_appeontutor_eascomps", as shown in Figure 5-29.

Figure 5-29: New Component deployed



STEP 18 – Close the *p_appeontutor_eascomps* Project Painter.

5.2.1.e Generating EAServer Stubs and Skeletons

EAServer Stub and Skeleton files allow communication between a Client and a component in EAServer. They act as the agents used by Appeon Web applications to access the PowerBuilder NVO components deployed in Appeon Server.

To generate Stub and Skeleton files:

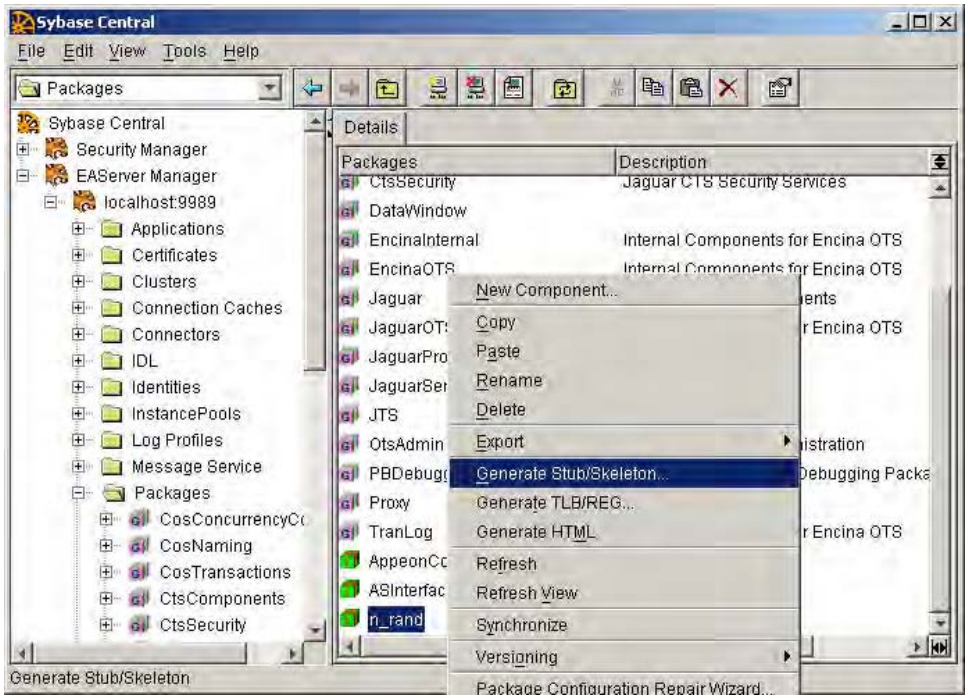
STEP 1 – Start EAServer if it is not already started.

STEP 2 – Login to EAServer using EAServer Manager. If you do not know how to use EAServer Manager to login, refer to the *Logging in EAServer* section in this tutorial.

STEP 3 – Right click on the *n_rand* package under the Packages node and choose *Generate Stub/Skeleton* from the popup menu, as shown in Figure 5-30.

This package is created during the deployment of the EAServer component project in the previous section.

Figure 5-30: Generate Stub/Skeleton

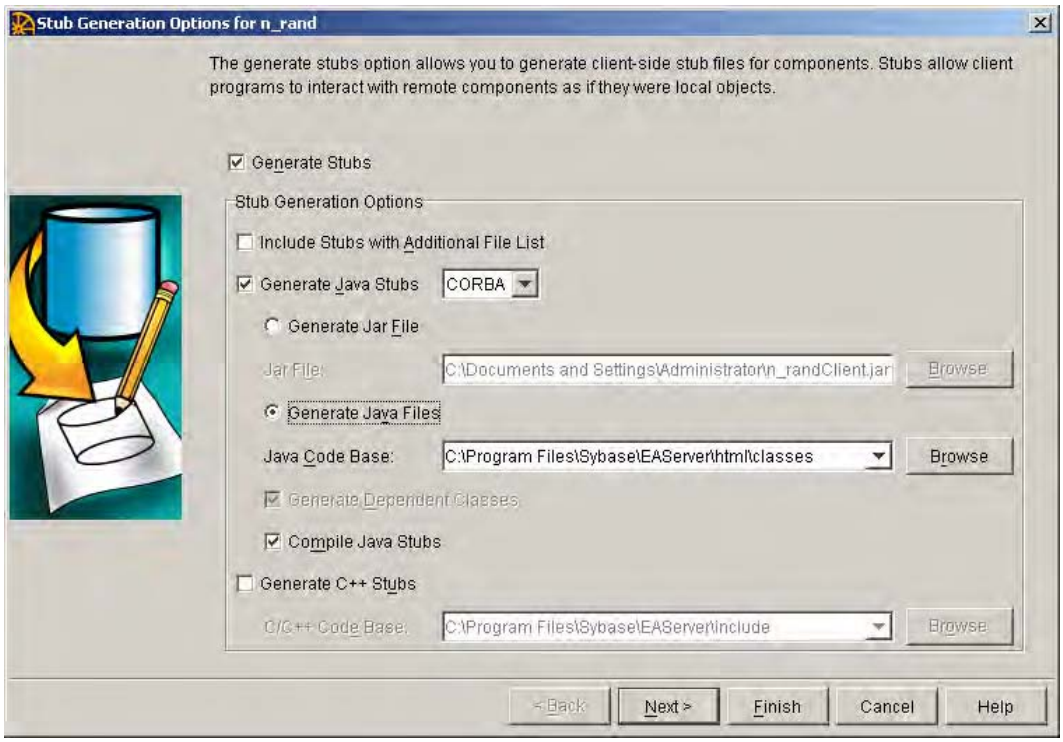


STEP 4 – Check the following options and leave other options at their default values, as shown in Figure 5-31.

- Generate Stubs
- Generate Java Stubs
- Generate Java Files
- Compile Java Stubs

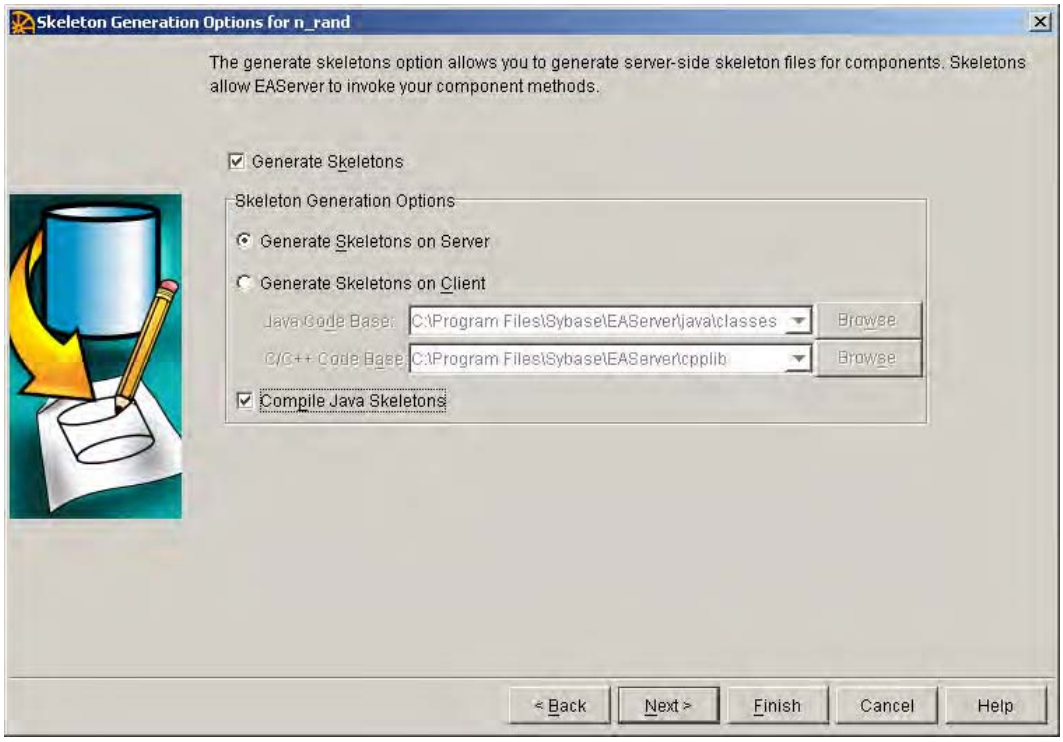
Click *Next*.

Figure 5-31: Stub Generation Options



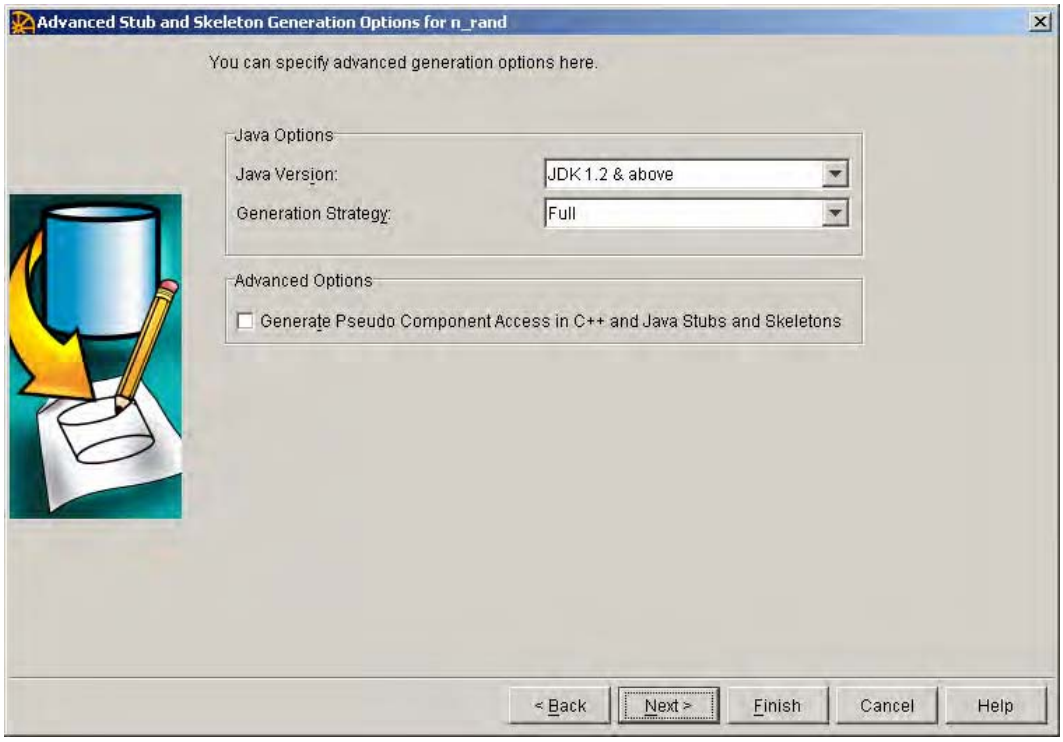
STEP 5 – Select the *Generate Skeletons* and *Compile Java Skeletons* options and leave the others at their default values, as shown in Figure 5-32. Click *Next*.

Figure 5-32: Skeleton Generation Options



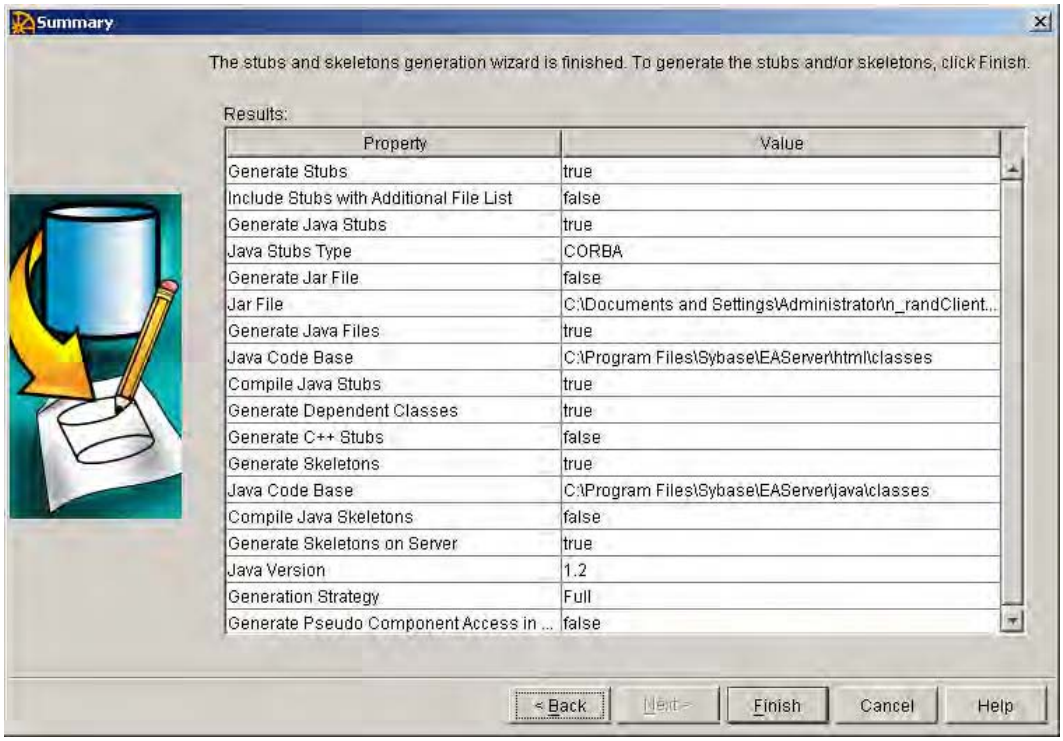
STEP 6 – Leave all options at default and click *Next*, as shown in Figure 5-33.

Figure 5-33: Advanced Stub and Skeleton Generation Options



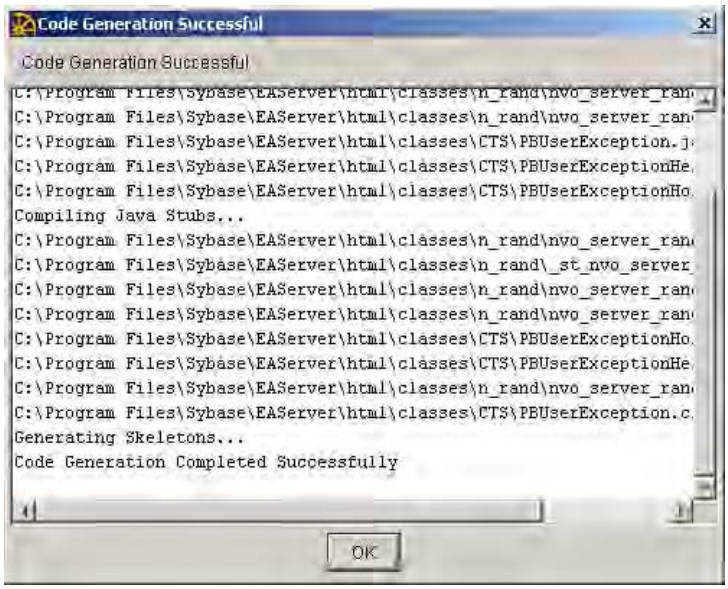
STEP 7 – The wizard now comes to the last screen, as shown in Figure 5-34. Click *Finish*.

Figure 5-34: Summary



STEP 8 – EAServer automatically generates the Stub and Skeleton files, as shown in Figure 5-35. Once code generation has completed successfully, click *OK*. Now exit EAServer Manager.

Figure 5-35: Code Generation Successful



5.2.1.f Generating the EAServer Proxy object

A PowerBuilder Proxy object is a local representation of a remote object in a middle-tier server. To communicate with an EAServer component, a PowerBuilder Client needs to have a proxy available locally.

To generate the Proxy object in the tutorial application:

STEP 1 – Start EAServer, if it is not already started.

STEP 2 – Choose *File | New* from the PowerBuilder menu. Then select the *EAServer Proxy Wizard* icon under the Project tab in the New dialog box, as shown in Figure 5-36. Click *OK*.

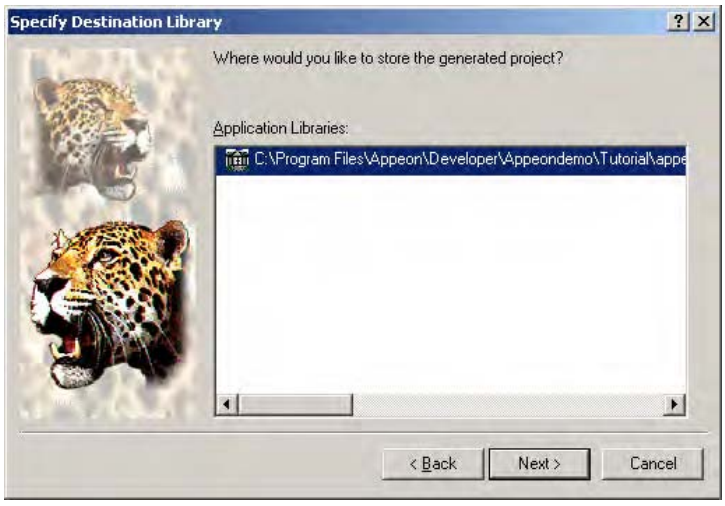
Figure 5-36: New

STEP 3 – The EAServer Proxy Project wizard starts, as shown in Figure 5-37. Click *Next*.

Figure 5-37: EAServer Proxy Wizard

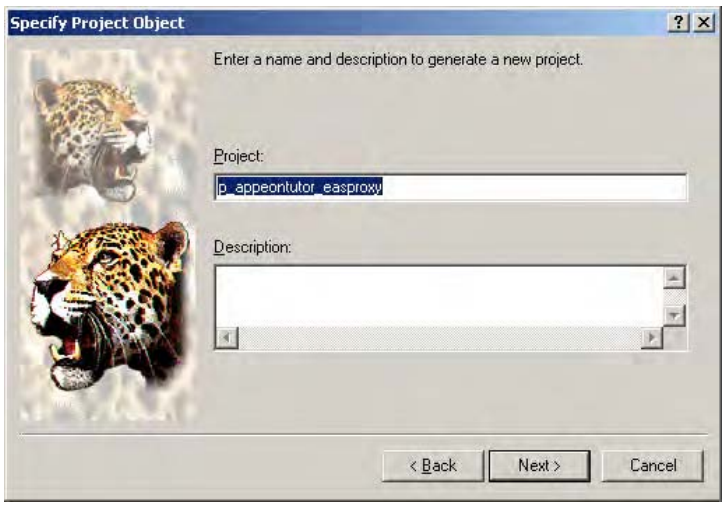
STEP 4 – Store the Proxy project in the default *appeontutor.pbl* file, as shown in Figure 5-38. Click *Next*.

Figure 5-38: Specify Destination Library



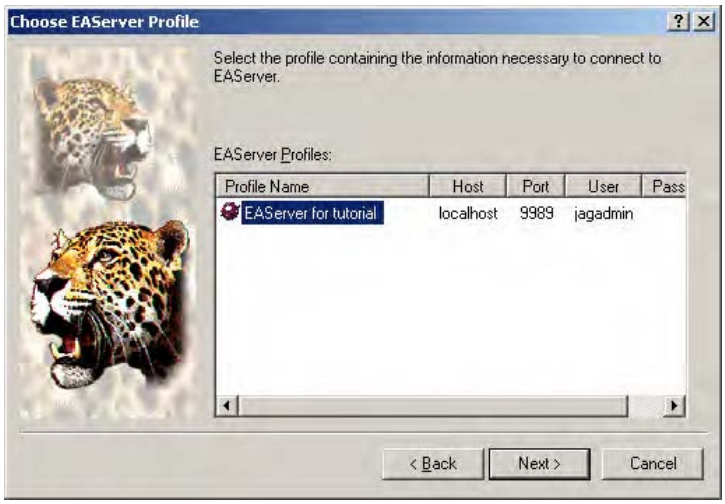
STEP 5 – Do not change the default name for the proxy project (p_appeontutor_easproxy), as shown in Figure 5-39. Click *Next*.

Figure 5-39: Specify Project Object



STEP 6 – Choose the *EAServer for tutorial* profile. (You created this profile previously) Make sure that EAServer is running at this time. Click *Next*.

Figure 5-40: Choose EAServer Profile



STEP 7 – The Wizard connects to the EAServer specified in the previous step and lists all the packages/components in that EAServer. Make sure to check *n_rand*, as shown in Figure 5-41. Click *Next*.

Figure 5-41: Select Components

STEP 8 – Leave the default library (*appeontutor.pbl*) to store the Proxy object, as shown in Figure 5-42. Click *Next*.

Figure 5-42: Specify New Proxy Library Name

STEP 9 – Leave all the options unchecked (default), as shown in Figure 5-43, and click *Next*.

Figure 5-43: Specify Build Options

STEP 10 – The Wizard comes to the last screen, as shown in Figure 5-44. Click *Finish*.

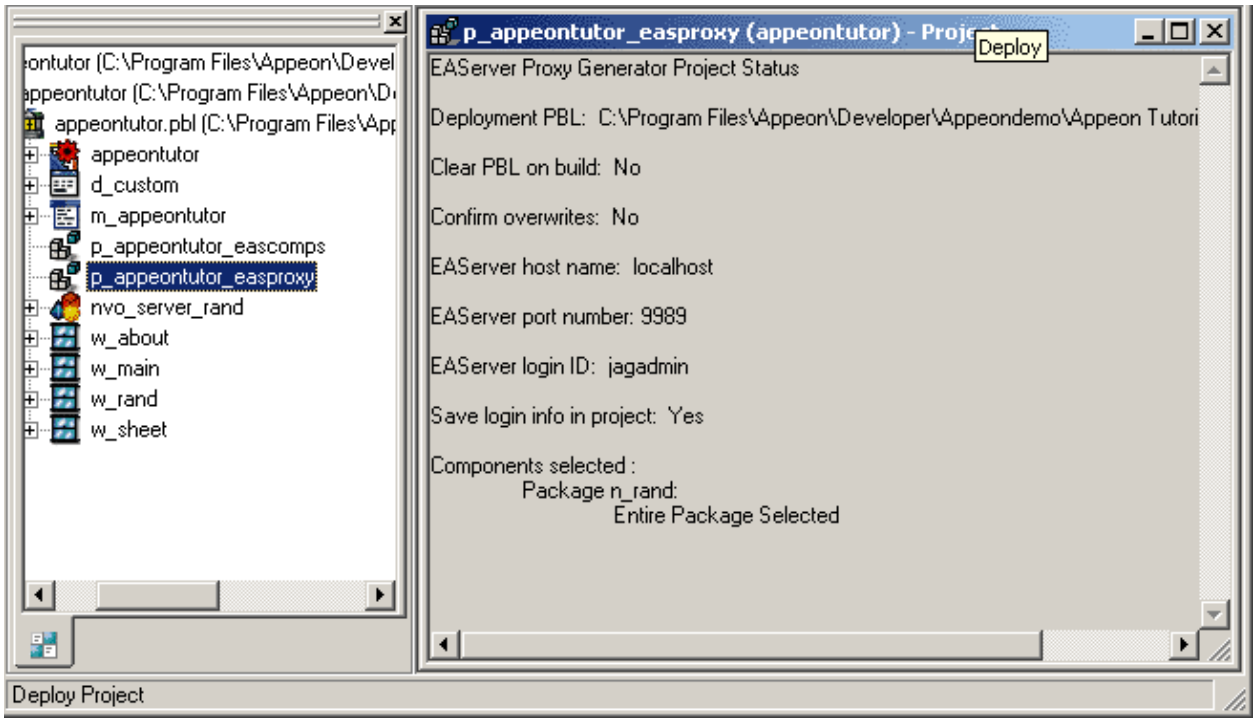
Figure 5-44: Ready to Create EAServer Proxy

STEP 11 – The Proxy project is automatically created and opened in the Project Painter in PowerBuilder. The Proxy project name (p_appeontutor_easproxy) is listed in the system tree under the appeontutor application Target.

Make sure that EAServer is running at this time.

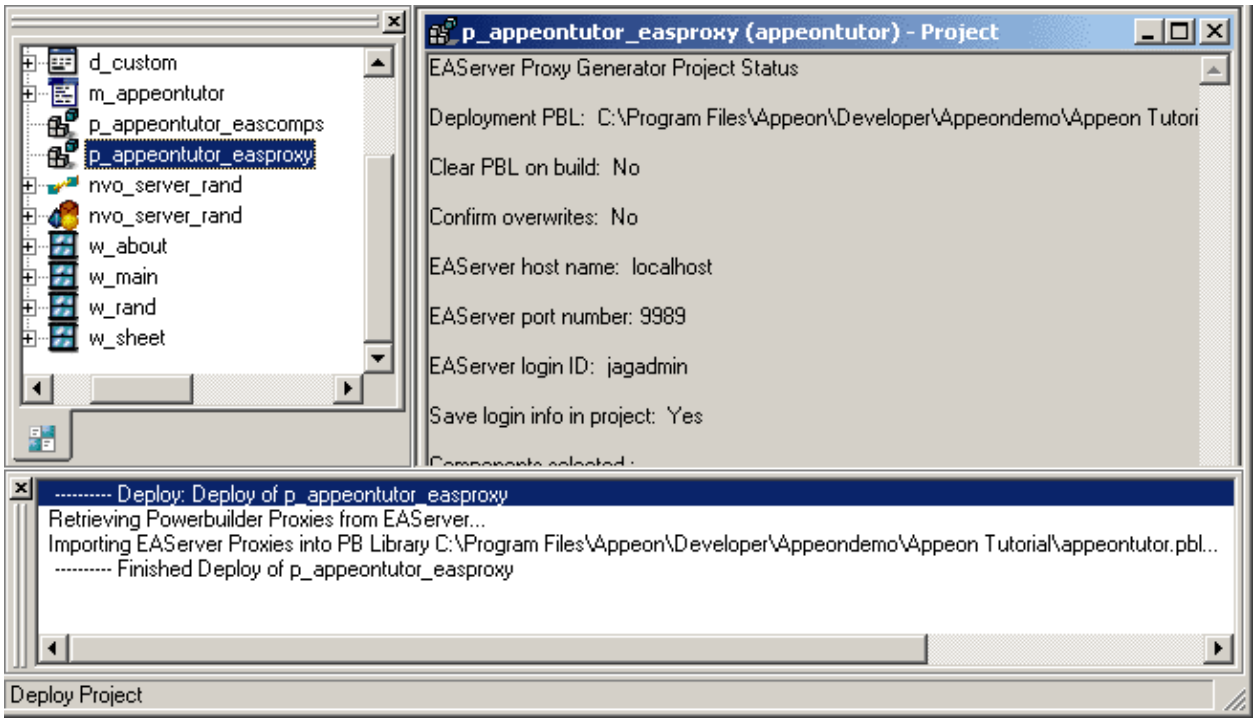
STEP 12 – Click the *Deploy* button in PainterBar1, as shown in Figure 5-45, or choose *Design | Deploy Project* from the PowerBuilder menu. PowerBuilder begins to deploy the Proxy project.

Figure 5-45: Deploy Proxy project



STEP 13 – The deployment process of the Proxy project is displayed in the PowerBuilder Output window, as shown in Figure 5-46. After it deploys successfully, the Proxy object *nvo_server_rand* appears in the application objects list.

Figure 5-46: Finished Deploy of p_appeontutor_easproxy



5.2.1.g Modifying the tutorial application to invoke the NVO components

The NVO holding the unsupported features has been deployed into EAServer, and the Proxy object has been prepared in the tutorial application Client. Now you will modify the tutorial application to interact with the remote NVO component in EAServer.

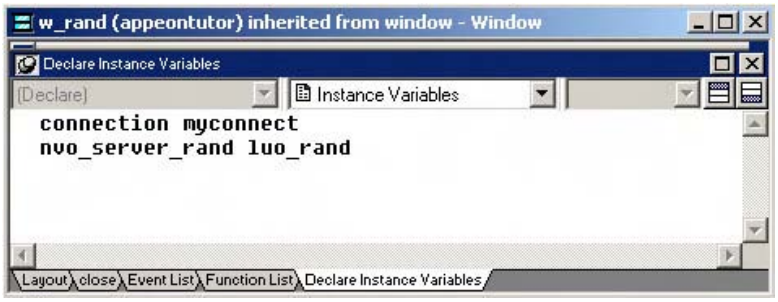
To modify the tutorial application to work with the NVO:

STEP 1 – Add the following two instance variables to the *w_rand* Window.

```
connection myconnect
nvo_server_rand luo_rand
```

The *myconnect* Connection variable is used to establish the connection to EAServer. The *luo_rand* variable of *nvo_server_rand* type will hold the object instance of the NVO component in EAServer.

Figure 5-47: Declare instance variables



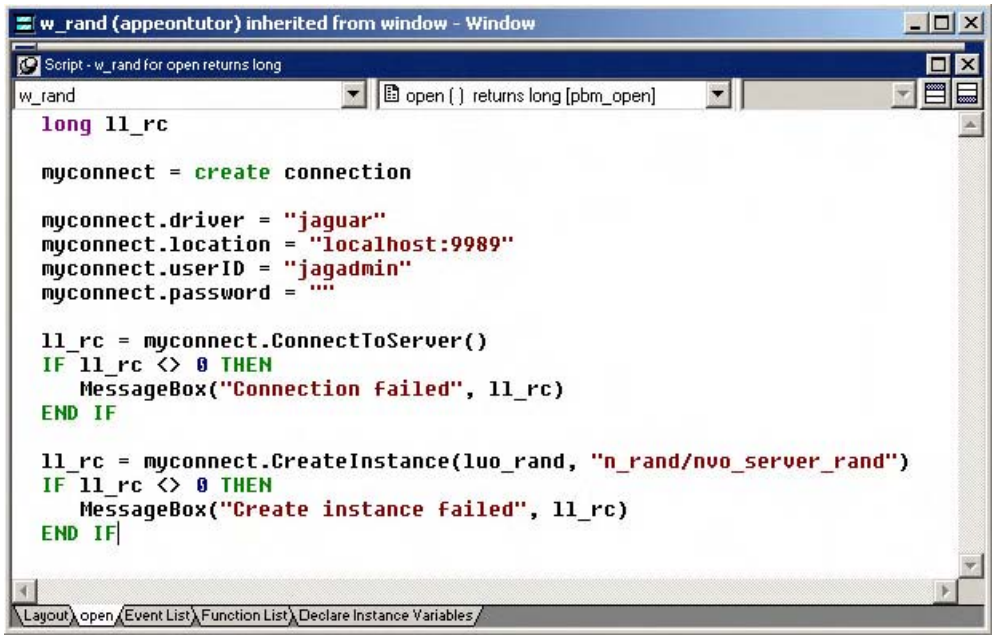
STEP 2 – Connect to EAServer and get the server component instance when the *w_rand* Window opens.

In the *Open* event of *w_rand*, add the following code:

```
long ll_rc
```

```
// Create connection object
myconnect = create connection

// Set properties to connect
myconnect.driver = "jaguar"
myconnect.location = "localhost:9989"
myconnect.userID = "jagadmin"
myconnect.password = ""

// Connect to EAServer
ll_rc = myconnect.ConnectToServer()
IF ll_rc <> 0 THEN
   MessageBox("Connection failed", ll_rc)
END IF

// Get object instance
ll_rc = myconnect.CreateInstance(luo_rand,
"n_rand/nvo_server_rand")
IF ll_rc <> 0 THEN
   MessageBox("Create instance failed", ll_rc)
END IF
```

Figure 5-48: Script in the w_rand Open event

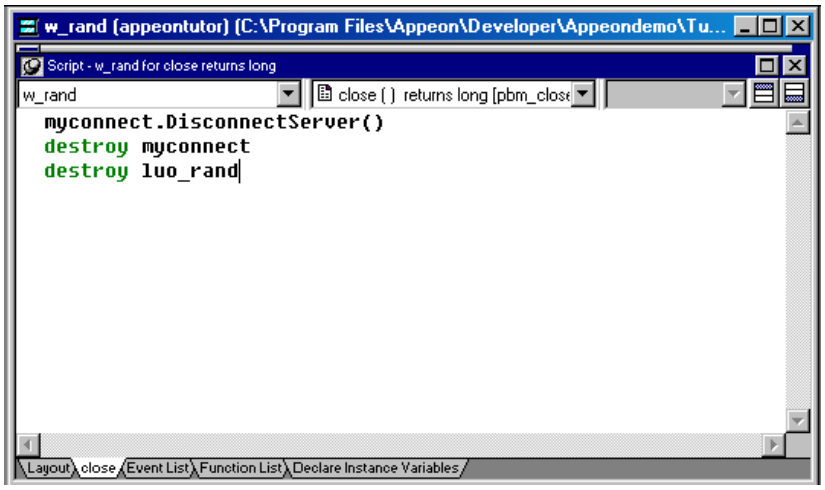


STEP 3 – Disconnect from EAServer and delete the Connection object after the *w_rand* Window closes.

In the *Close* event of *w_rand*, destroy what was created: the connection object and the NVO instance:

```
myconnect.DisconnectServer()
destroy myconnect
destroy luo_rand
```

Figure 5-49: Script in the w_rand Close event

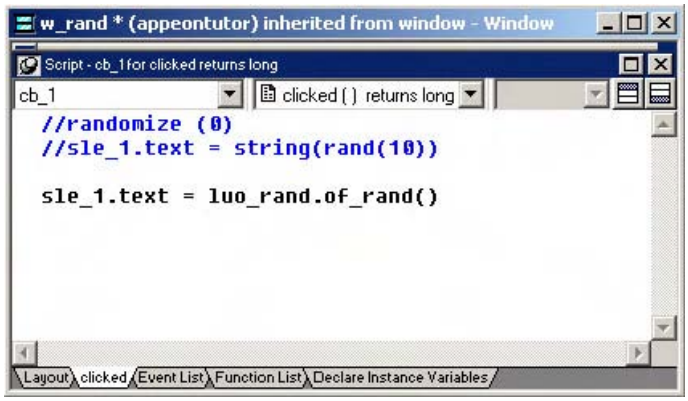


STEP 4 – Now call the functions in the NVO to retrieve and display the randomized number when you click the button in the *w_rand* Window.

In the *Clicked* event of *cb_1* CommandButton in the *w_rand* Window, add the following code:

```
//randomize (0)
//sle_1.text = string(rand(10))

sle_1.text = luo_rand.of_rand()
```

Figure 5-50: Script in the cb_1 Clicked event



5.2.1.h Backing up the PowerBuilder NVO

In previous steps, the *nvo_server_rand* NVO was deployed into EAServer. You have also modified the tutorial application Client to interact with the server NVO component. This NVO and its component project are no longer needed in the Client. You can now

- move them from the *appeontutor* application; or
- use the PowerBuilder NVO in the Client/Server application.

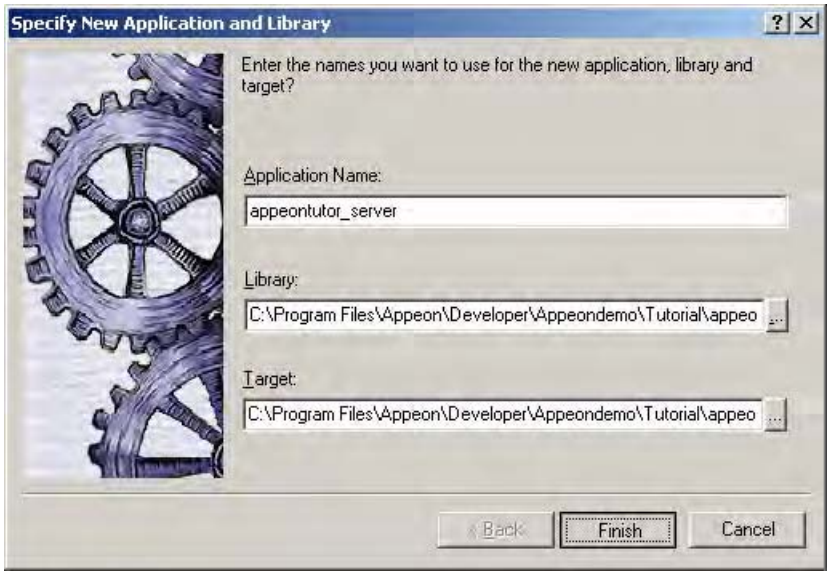
To move the PowerBuilder NVO into another application target:

STEP 1 – Choose *File* | *New* from the PowerBuilder menu. Then select the *Application* icon under the Target tab in the New dialog box, as shown in Figure 5-51. Click *OK*.

Figure 5-51: New

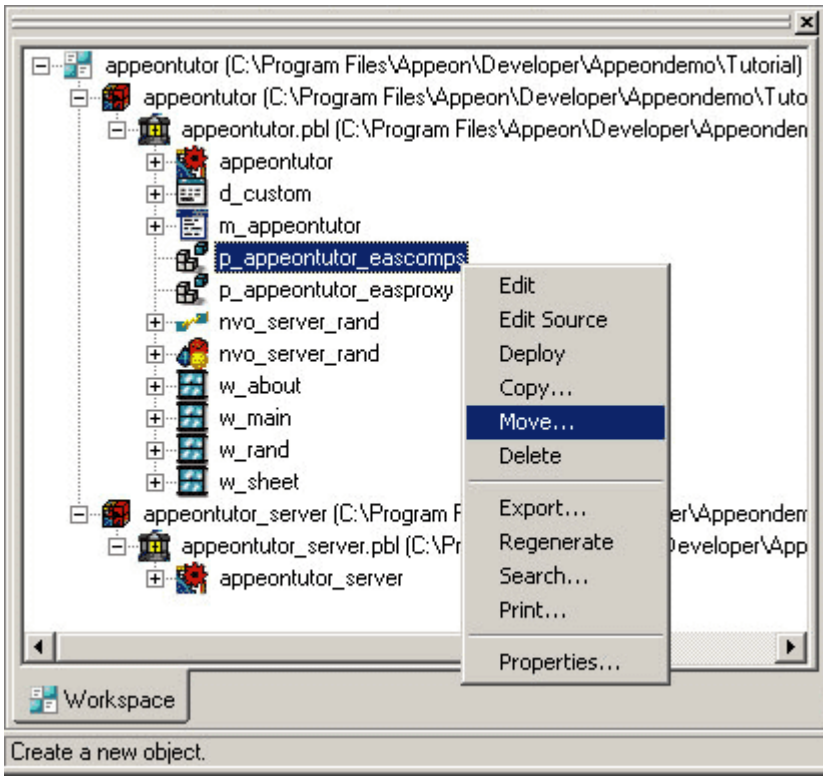
STEP 2 – Type *appeontutor_server* as the Application Name, and click *Finish*, as shown in Figure 5-52.

Figure 5-52: Specify New Application and Library



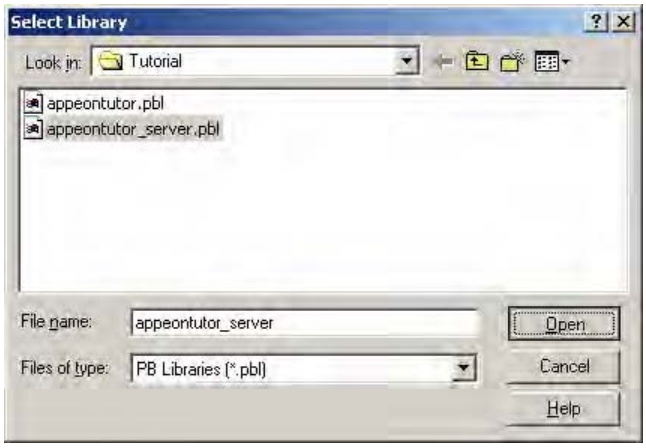
STEP 3 – The *appeontutor_server* Application Target has now been added into the appeontutor Workspace. Right click on the *p_appeontutor_eascomps* Project under *appeontutor.pbl*, and select *Move* from the popup menu.

Figure 5-53: Move



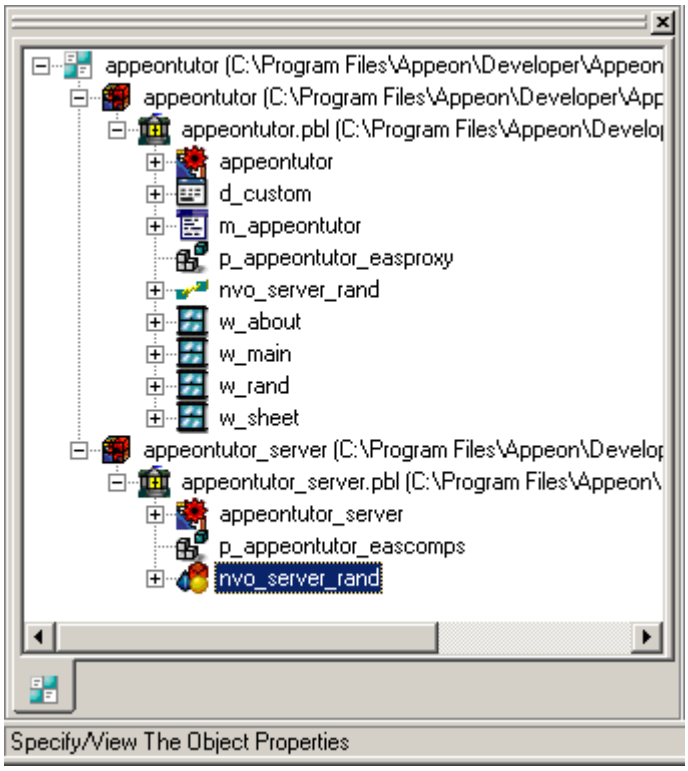
STEP 4 – Select the *appeontutor_server.pbl* file, and click *Open*, as shown in Figure 5-54.

Figure 5-54: Select Library



STEP 5 – The *p_appeontutor_eascomps* Project has been moved into *appeontutor_server.pbl*. Repeat the same process to move the *nvo_server_rand* NVO into the *appeontutor_server* Application Target, as shown in Figure 5-55.

Figure 5-55: Already moved

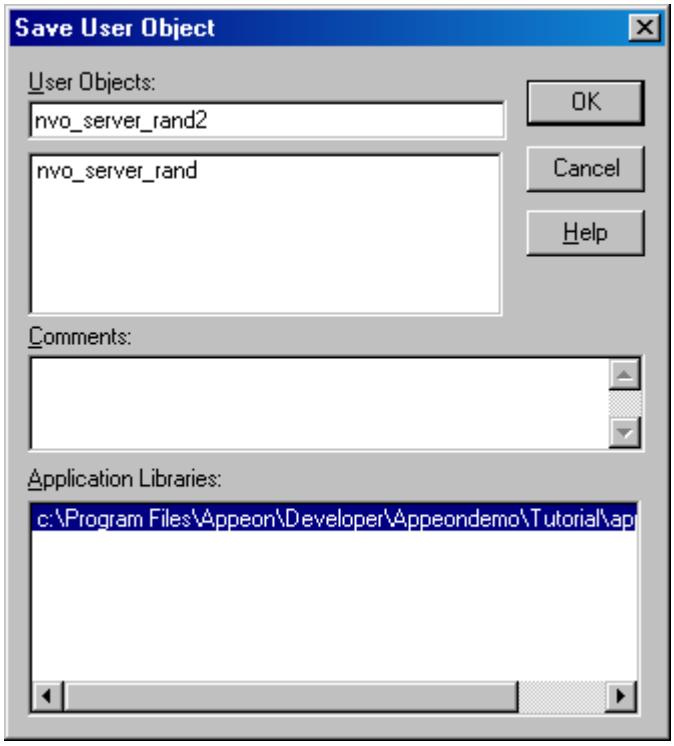


To use the PowerBuilder NVO in the Client/Server application:

STEP 1 – Save the PowerBuilder NVO with a new name.

With the NVO object *nvo_server_rand* selected, Choose *File* | *Save As…* from the PowerBuilder menu, and save the object as *nvo_server_rand2*.

Figure 5-56: Save the object as nvo_server_rand2



STEP 2 – Define a function in the *w_rand* Window for checking what architecture the application is run on (Client/Server or Browser/Server). For example, you can define a local function *is_Web* and write the following script in the function:

```
long ll_a,ll_b
SetNull(ll_a)
SetNull(ll_b)
If ll_a = ll_b Then
   Return True
Else
   Return False
End if
```

The function returns FALSE if the architecture is Client Server and returns TRUE if the architecture is Browser Server. This is because PowerScript and JavaScript handle Null values in different ways.

STEP 3 – Define two sets of Open and Close functions for the *w_rand* Window.

1. open_web and close_web

The *open_web* and *close_web* functions will be executed in Browser/Server architecture. Move the entire script in the *Open* event of the *w_rand* Window to the *open_web* function. Move the entire script of the *Close* event of the *w_rand* Window to the *close_web* function.

2. open_pb and close_pb

The *open_pb* and *close_pb* functions will be executed in Client/Server architecture.

Add the following instance variable to the *w_rand* Window.

```
nvo_server_rand2 luo_rand2
```

In the *open_pb* function, add the following code:

```
long ll_rc
ll_rc = 1
luo_rand2 = Create nvo_server_rand2
IF not IsValid(luo_rand2) THEN
   ll_rc = -1
   MessageBox("Create object failed", ll_rc)
end if
return ll_rc
```

In the *close_pb* function, add the following code:

```
destroy luo_rand2
```

STEP 4 – Add the following code to the *Open* and *Close* events of the *w_rand* Window.

The code in the *Open* event of the *w_rand* Window will be:

```
If is_Web() Then
   open_web()
Else
   open_pb()
End if
```

The code in the *Close* event of the *w_rand* Window will be:

```
If is_Web() Then
   close_web()
Else
```

```
    close_pb()
End if
```

STEP 5 – Modify the *Clicked* event of the *cb_1* CommandButton.

The code in the *Clicked* event of the *cb_1* CommandButton will be:

```
if is_web() then
sle_1.text = luo_rand.of_rand()
else
sle_1.text = luo_rand2.of_rand()
end if
```

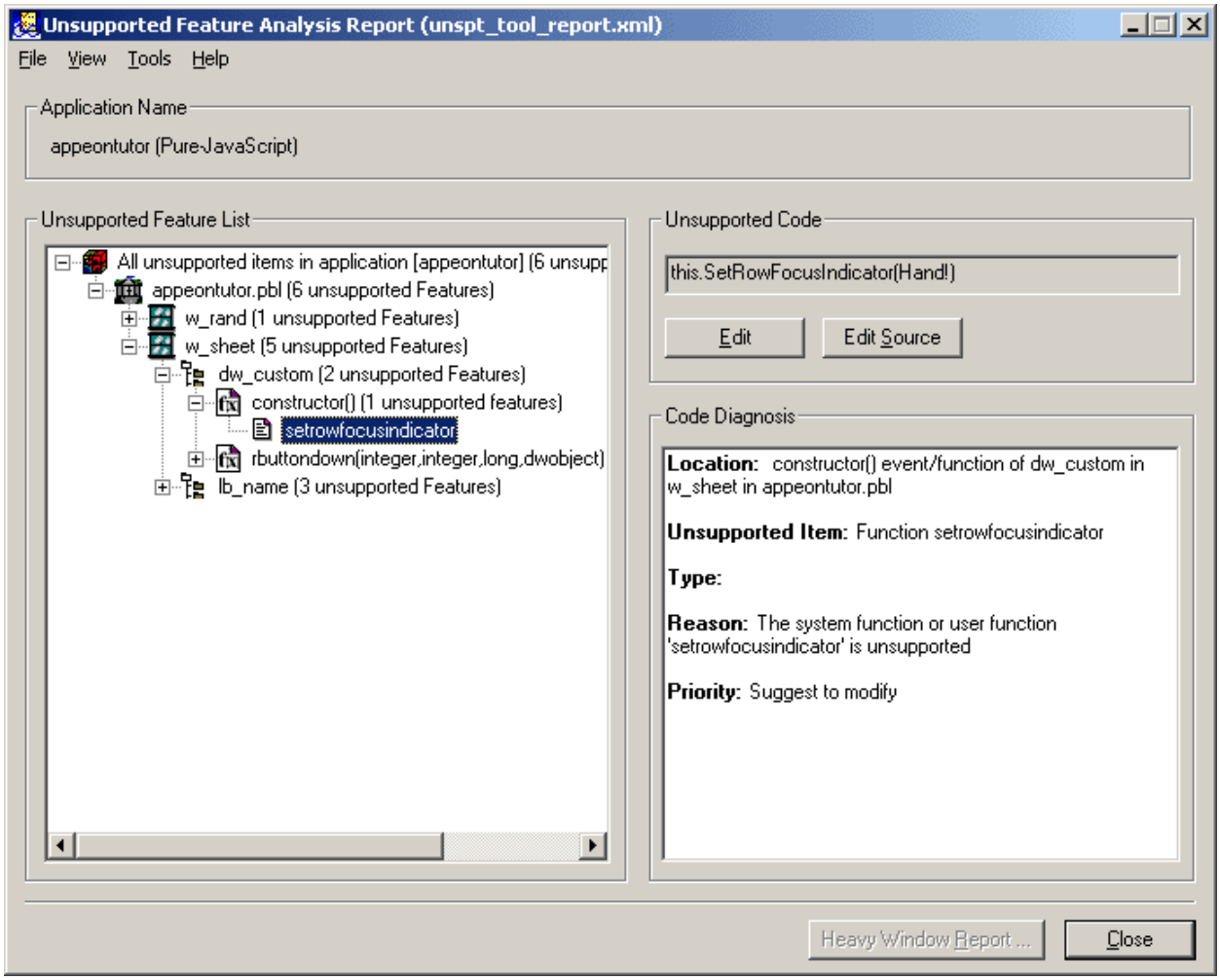
5.2.2 Removing unsupported features

The *SetRowFocusIndicator* function is reported as an unsupported feature in the Appeon tutorial application. This function is used to specify the visual indicator that identifies the current row in the DataWindow control. It is not an essential component of the application's functionality; we recommend removing it.

To remove the unsupported *SetRowFocusIndicator* function:

STEP 1 – In the UFA Report Window, expand *w_sheet* | *dw_custom* | *constructor* and select *setrowfocusindicator*, as shown in Figure 5-57.

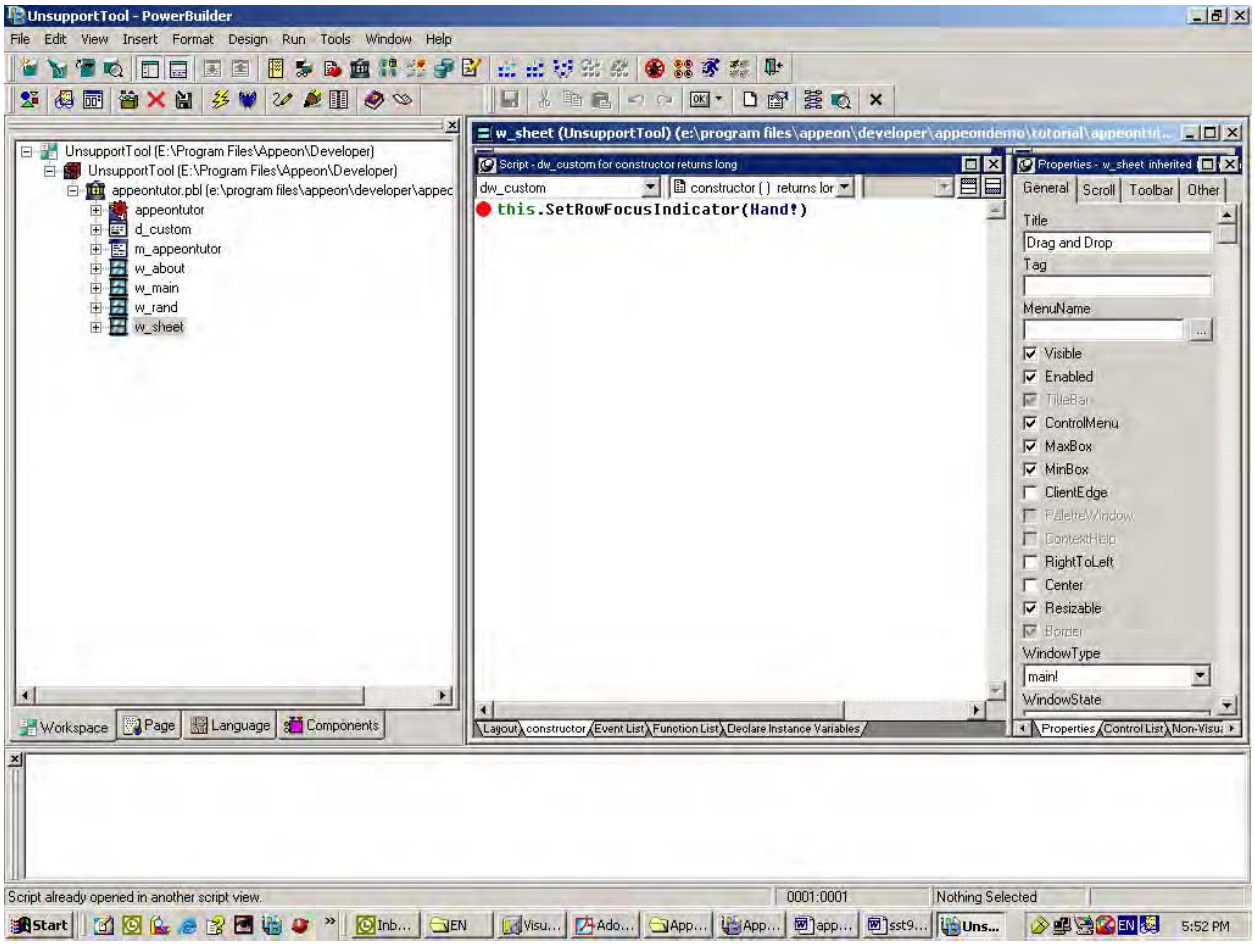
Figure 5-57: UFA Report Window



STEP 2 – Click the *Edit* button to enter the PowerBuilder Script view of the *Constructor* event of the *dw_customer* DataWindow in *w_sheet* Window.

A red bullet is placed at the left side of the *SetRowFocusIndicator* function to indicate that this coding feature is unsupported by Appeon, as shown in Figure 5-58.

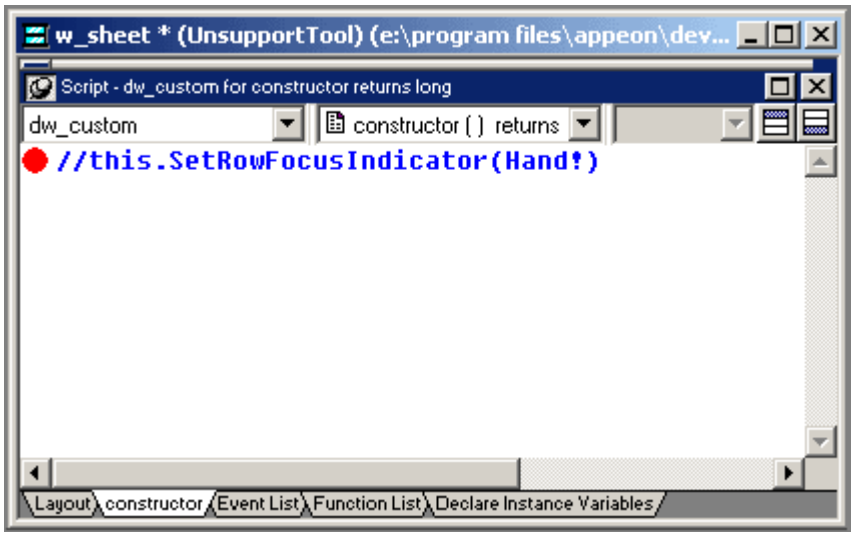
Figure 5-58: Unsupported code in the PowerBuilder painter



STEP 3 – Click in the first line in the Script view, and then click the *Comment* button () in PainterBar2.

By commenting out this line, the *SetRowFocusIndicator* function will not be executed when the *dw_custom* DataWindow is constructed.

Figure 5-59: Commenting unsupported script



STEP 4 – Click the *Save* button, and then click the *Close* button in PainterBar1.

5.2.3 Working around unsupported features

The Appeon tutorial PowerBuilder application uses the drag and drop technique. In the tutorial application, hold down the right mouse button on the *dw_custom* DataWindow control and drag the current row in the DataWindow. Drop the first and last names in the row into the *lb_name* ListBox.

PowerBuilder applications support drag and drop, but Appeon deployed Web applications do not. To work around this, comment the relevant code that implements drag and drop. Add additional buttons and script to perform the task of copying content from the DataWindow and pasting it into the ListBox.

To comment the relevant code that implements drag and drop:

STEP 1 – In the UFA Report Window, expand *w_sheet* | *dw_custom* | *rbuttondown* and select drag, as shown in Figure 5-61.

If the UFA Report Window is not open, click on the UFA tool icon on the status bar and select *Show report window* from the popup menu to display the UFA Report Window, as shown in Figure 5-60.

Figure 5-60: Popup menus

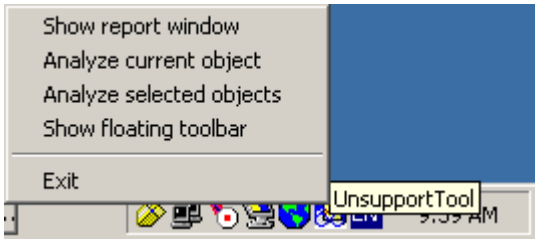
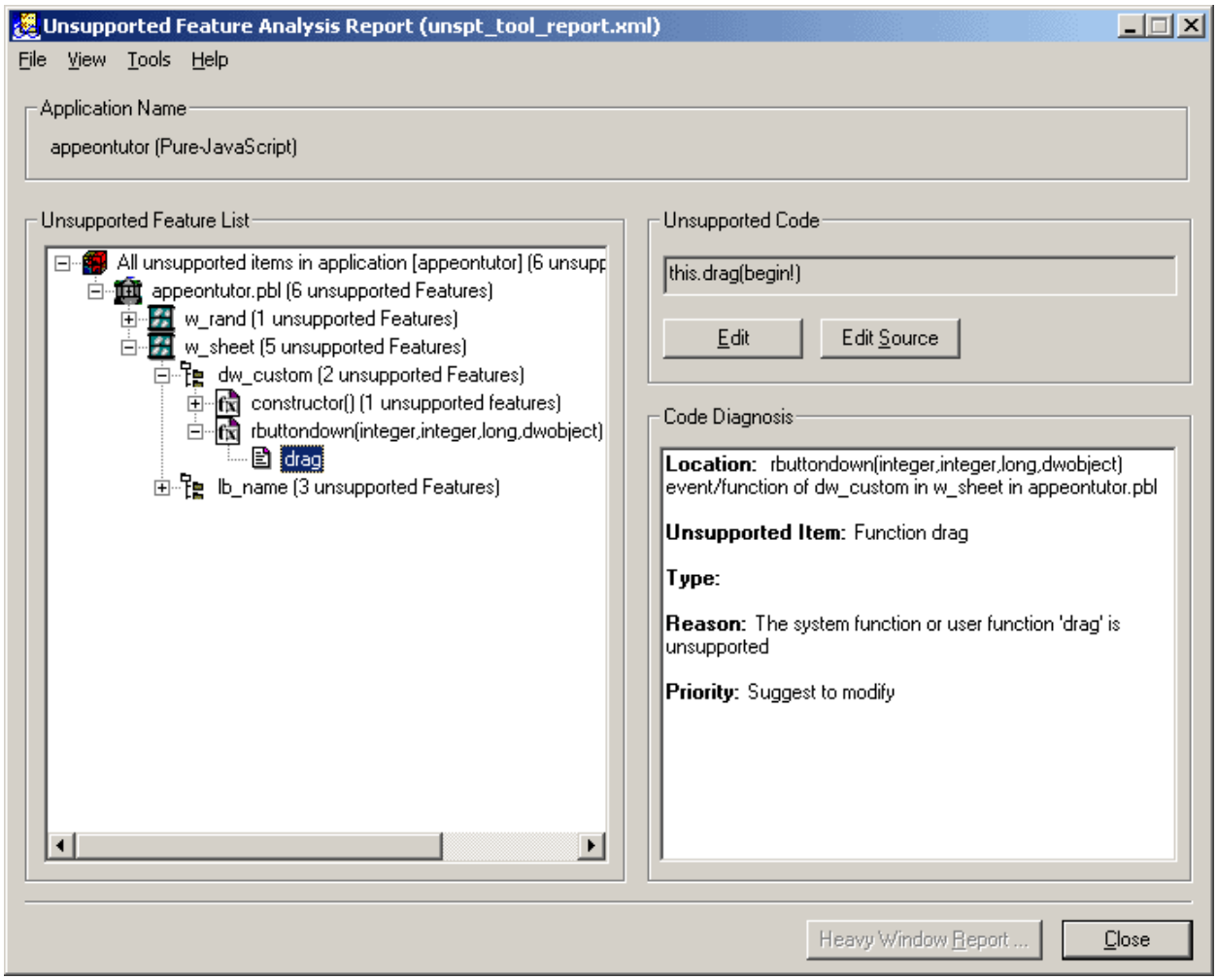


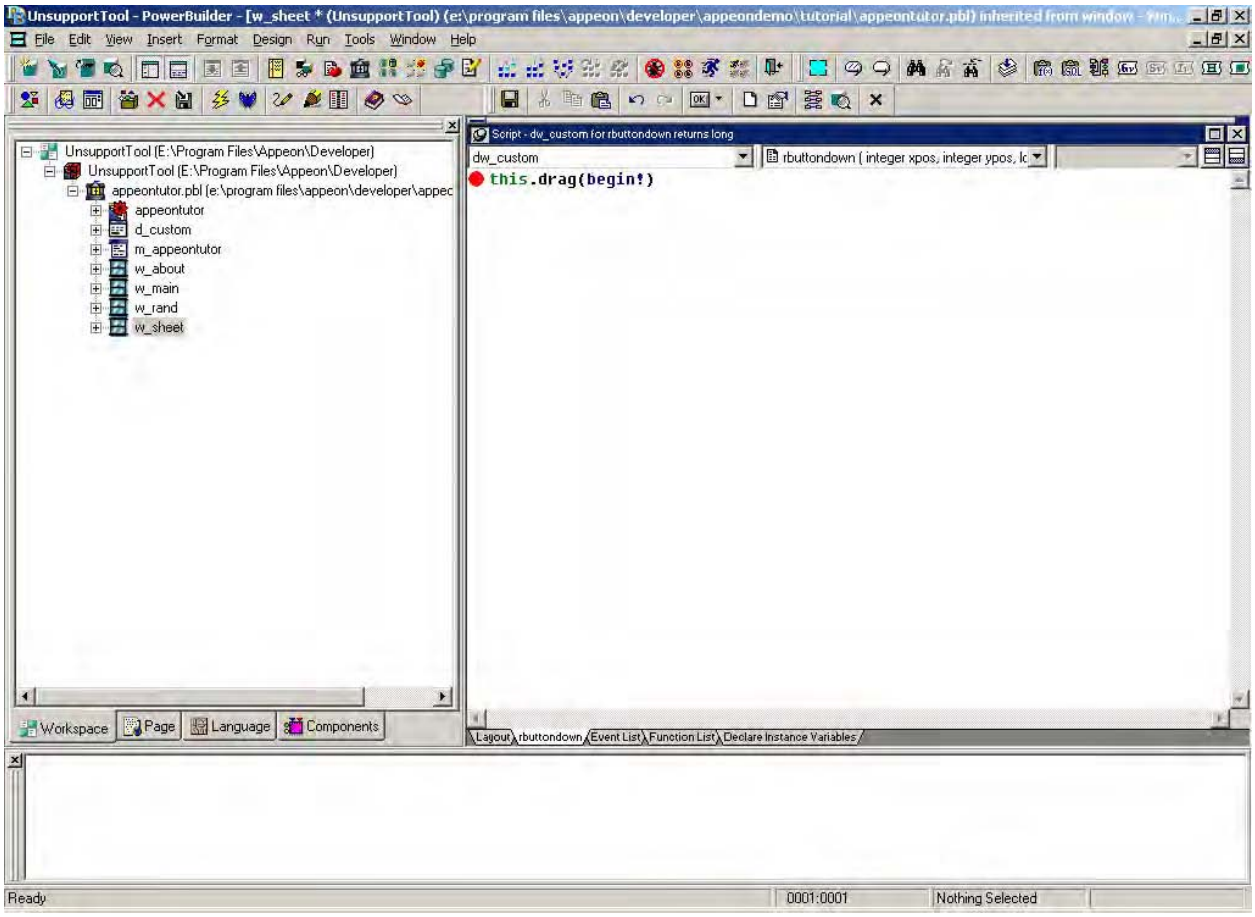
Figure 5-61: UFA Report Window



STEP 2 – Click the *Edit* button to enter the PowerBuilder Script view of the *Drag* function of the *dw_customer* DataWindow in *w_sheet* Window.

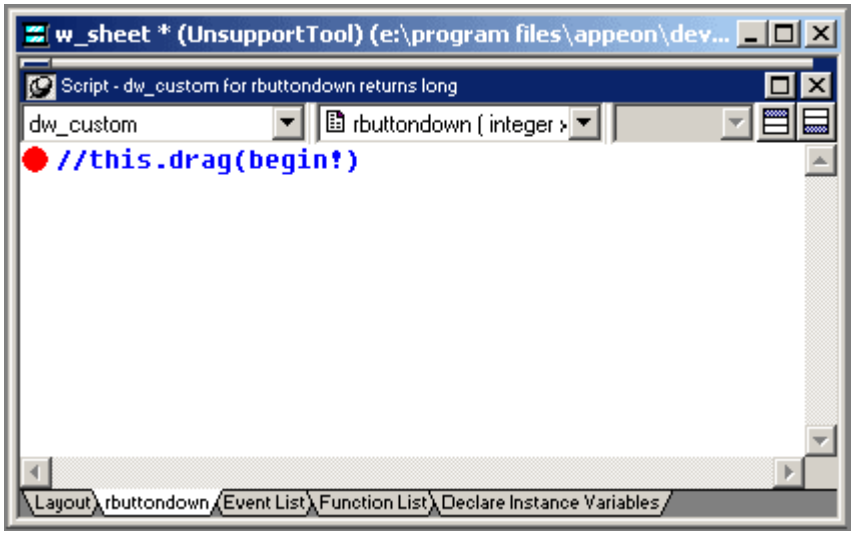
A red bullet is placed at the left side of the *Drag* function to indicate that this coding feature is unsupported by Appeon, as shown in Figure 5-62.

Figure 5-62: Unsupported code in the PowerBuilder painter



STEP 3 – Click in the first line in the Script view, and then click the *Comment* button ( ) in PainterBar2.

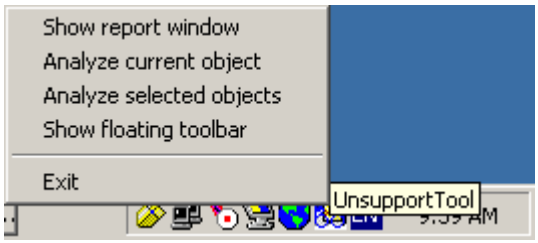
Figure 5-63: Commenting unsupported script



STEP 4 – Click the *Save* button, and then click the *Close* button in PainterBar1.

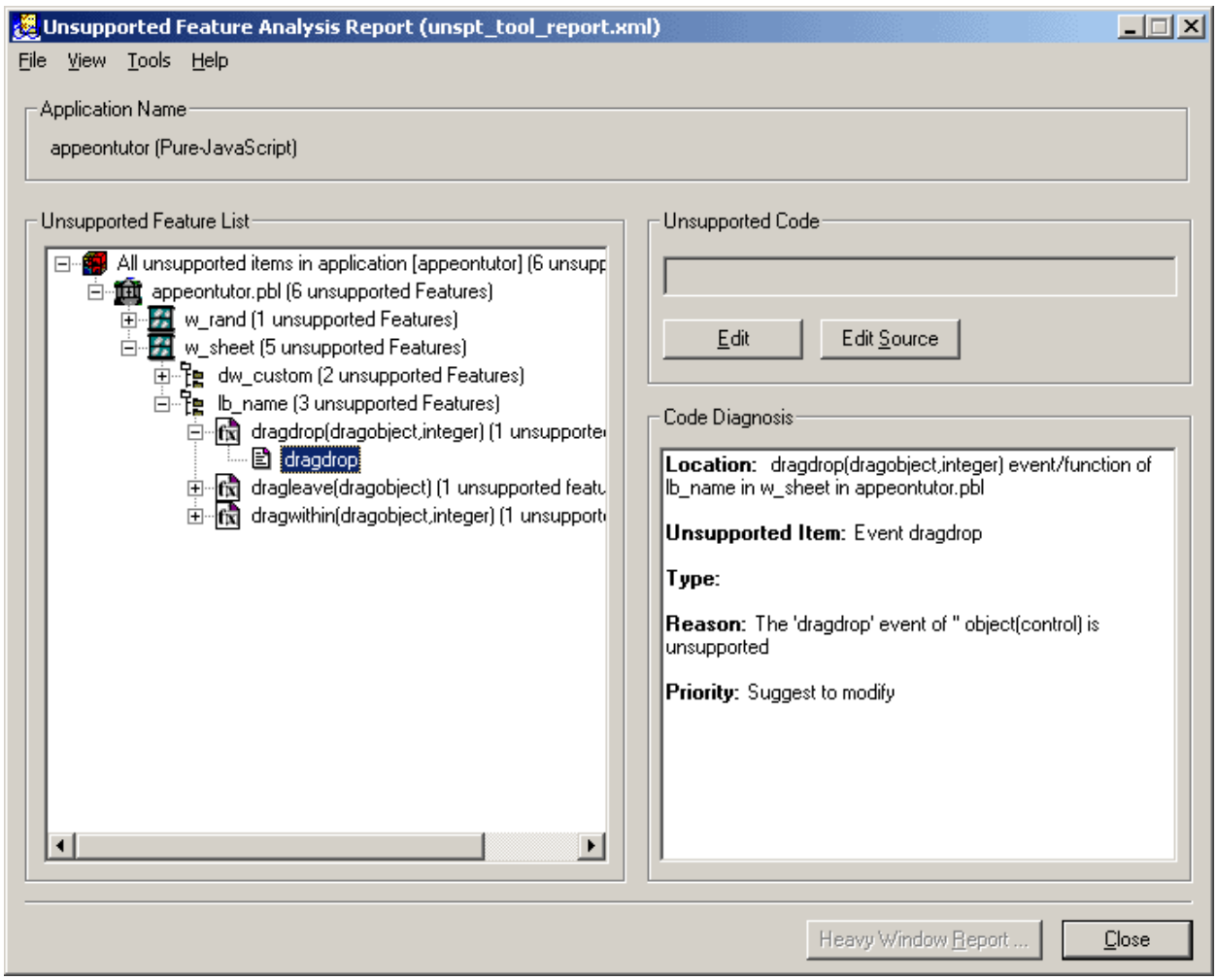
STEP 5 – Click on the UFA tool icon on the status bar and select *Show report window* from the popup menu to display the UFA Report Window, as shown in Figure 5-64.

Figure 5-64: Popup menus



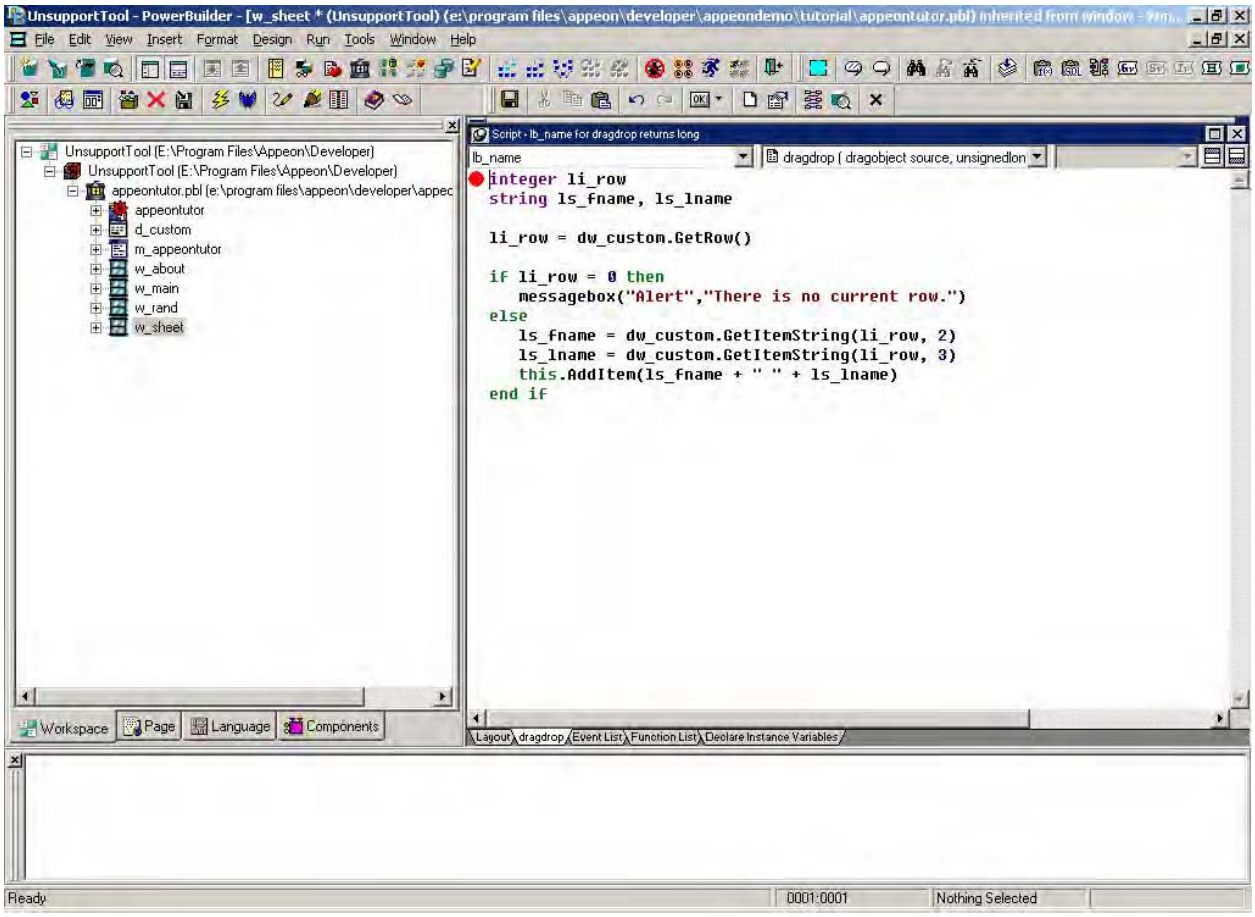
STEP 6 – In the UFA Report Window, expand *w_sheet* | *lb_name* | *dragdrop* and select dragdrop, as shown in Figure 5-65.

Figure 5-65: UFA Report Window



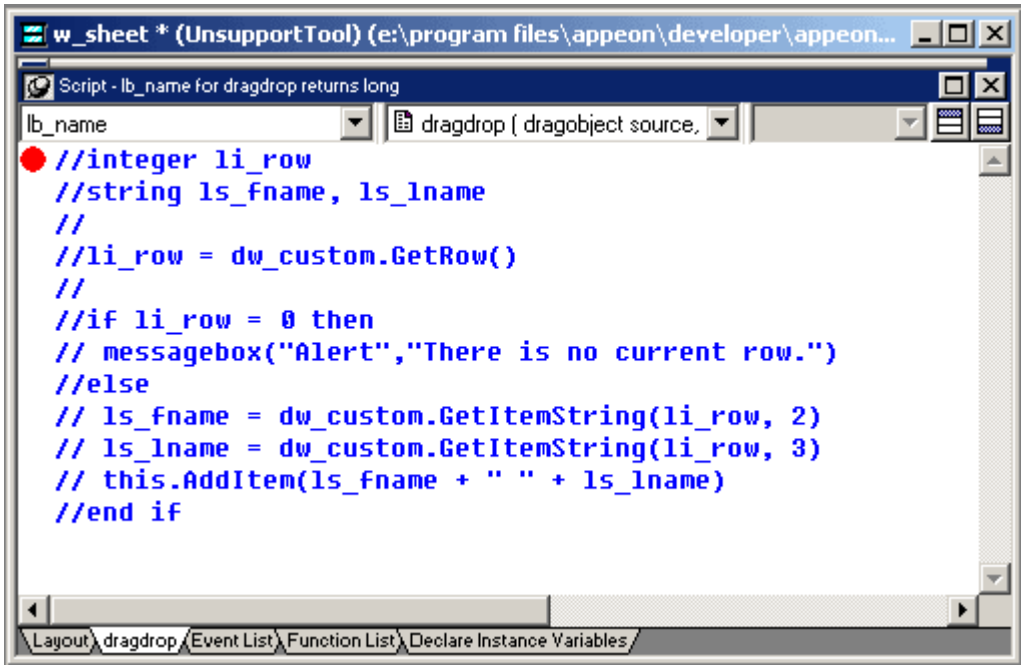
STEP 7 – Click the *Edit* button to enter the PowerBuilder Script view of the *DragDrop* event of the *lb_name* ListBox in *w_sheet* Window, as shown in Figure 5-66.

Figure 5-66: Unsupported code in the PowerBuilder painter



STEP 8 – Select the entire script of the *DragDrop* event, and then click the *Comment* button () in PainterBar2.

Figure 5-67: Commenting unsupported script

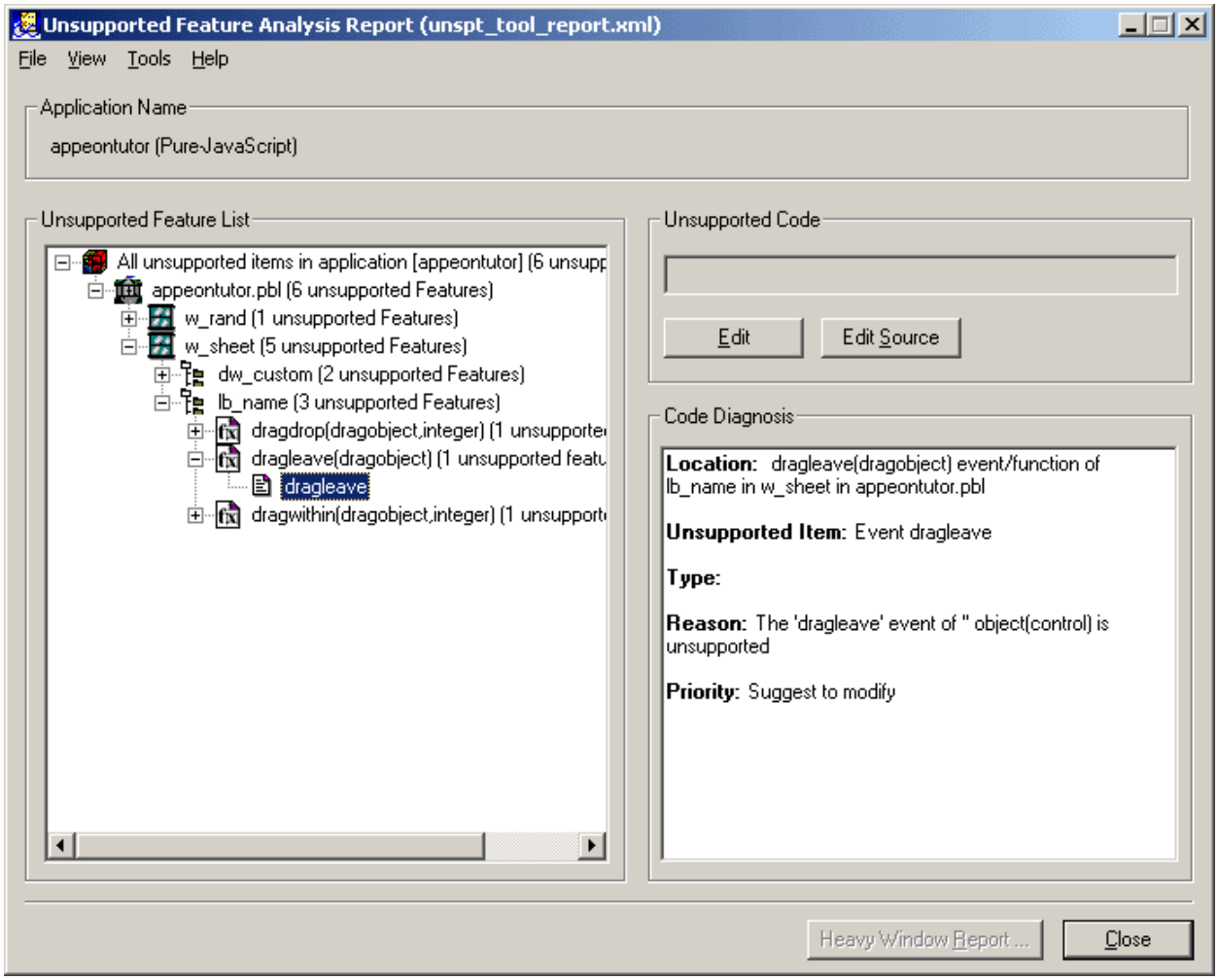


STEP 9 – Click the *Save* button, and then click the *Close* button in PainterBar1.

STEP 10 – Click on the UFA tool icon on the status bar and select *Show report window* from the popup menu to display the UFA Report Window.

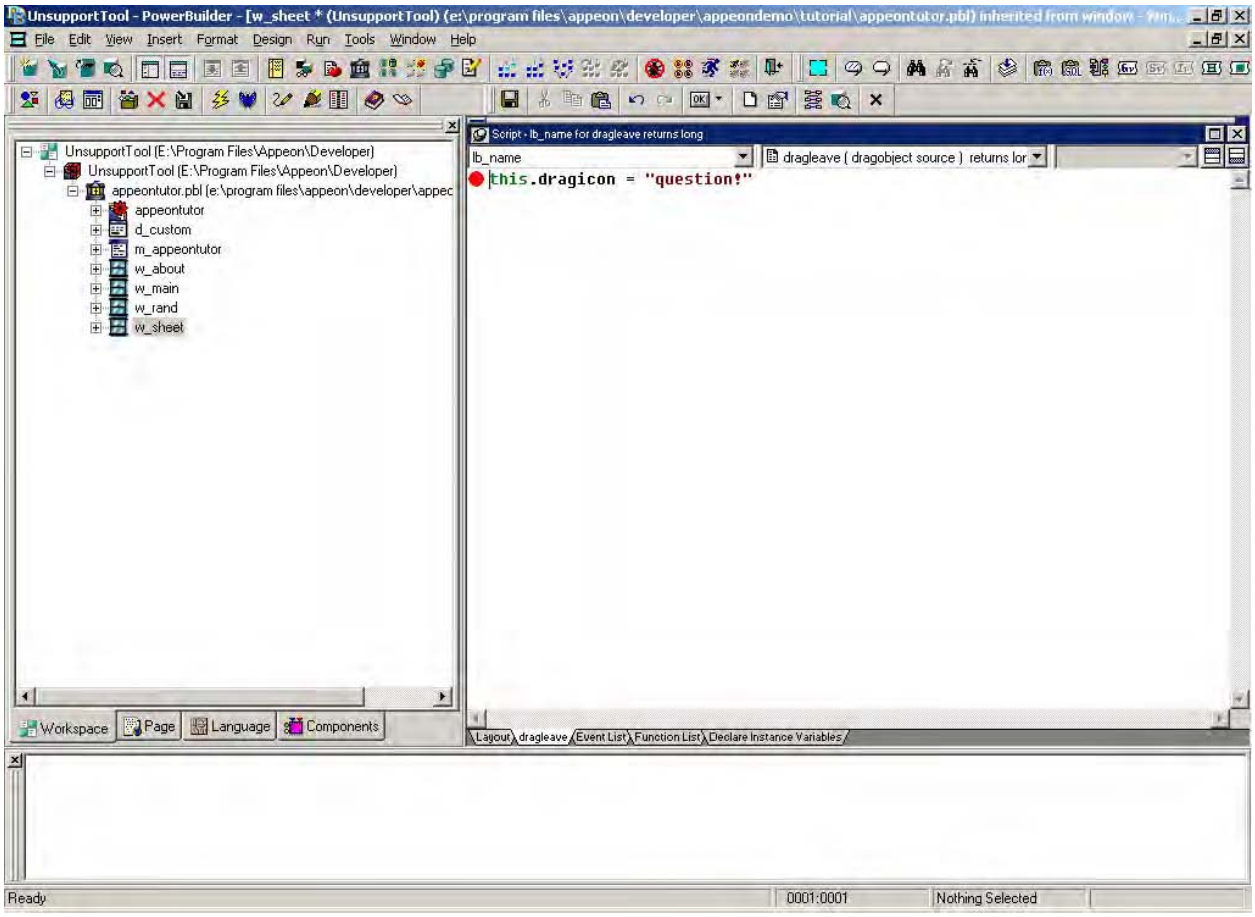
STEP 11 – In the UFA Report Window, expand *w_sheet* | *lb_name* | *dragleave* and select dragleave, as shown in Figure 5-68.

Figure 5-68: UFA Report Window



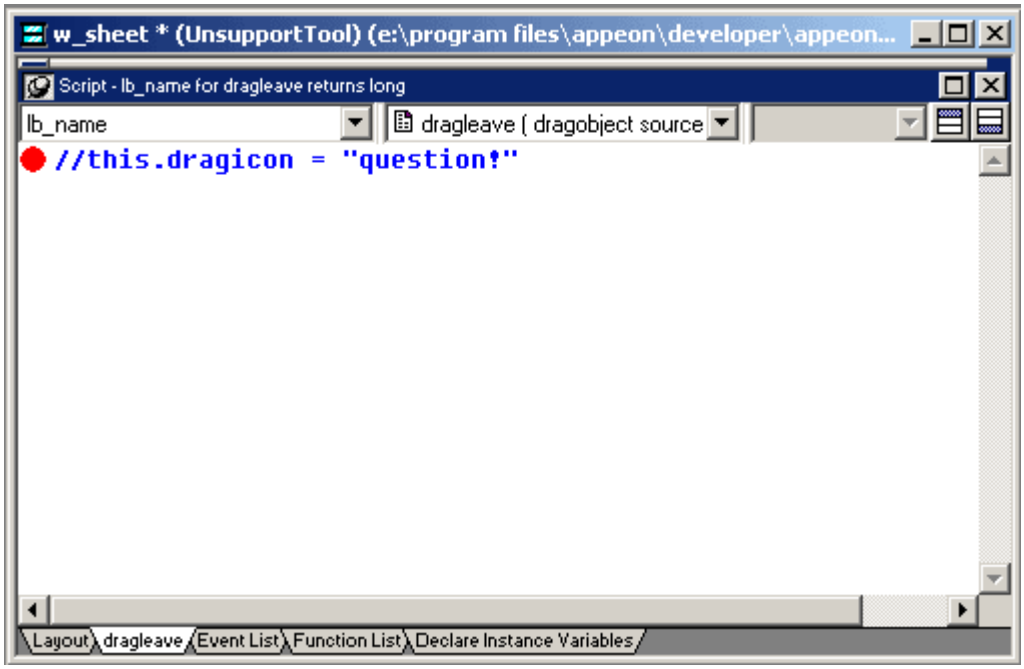
STEP 12 – Click the *Edit* button to enter the PowerBuilder Script view of the *DragLeave* event of the *lb_name* ListBox in *w_sheet* Window, as shown in Figure 5-69.

Figure 5-69: Unsupported code in the PowerBuilder painter



STEP 13 – Click in the first line in the Script view, and then click the *Comment* button ( ) in PainterBar2.

Figure 5-70: Commenting unsupported script

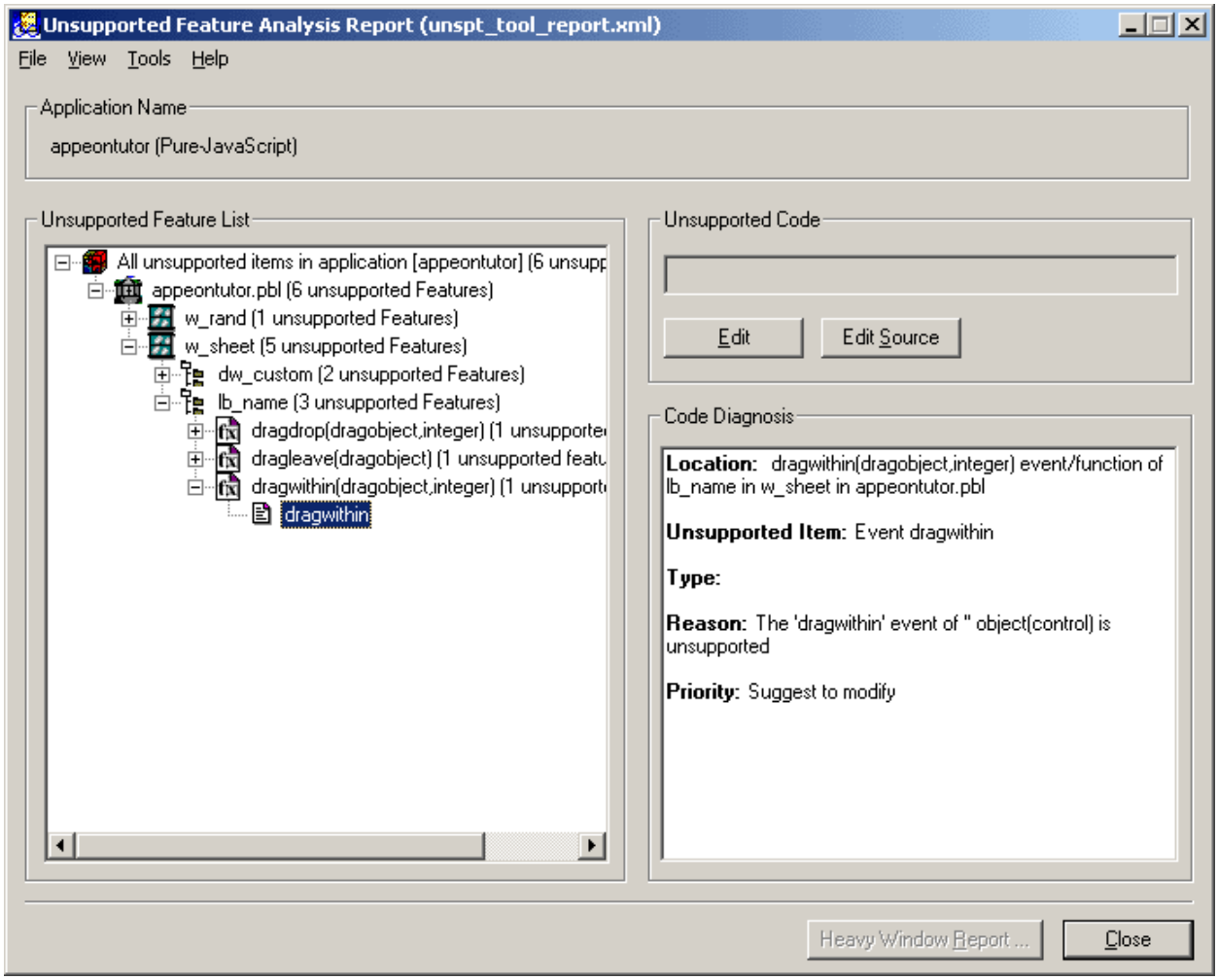


STEP 14 – Click the *Save* button, and then click the *Close* button in PainterBar1.

STEP 15 – Click on the UFA tool icon on the status bar and select *Show report window* from the popup menu to display the UFA Report Window.

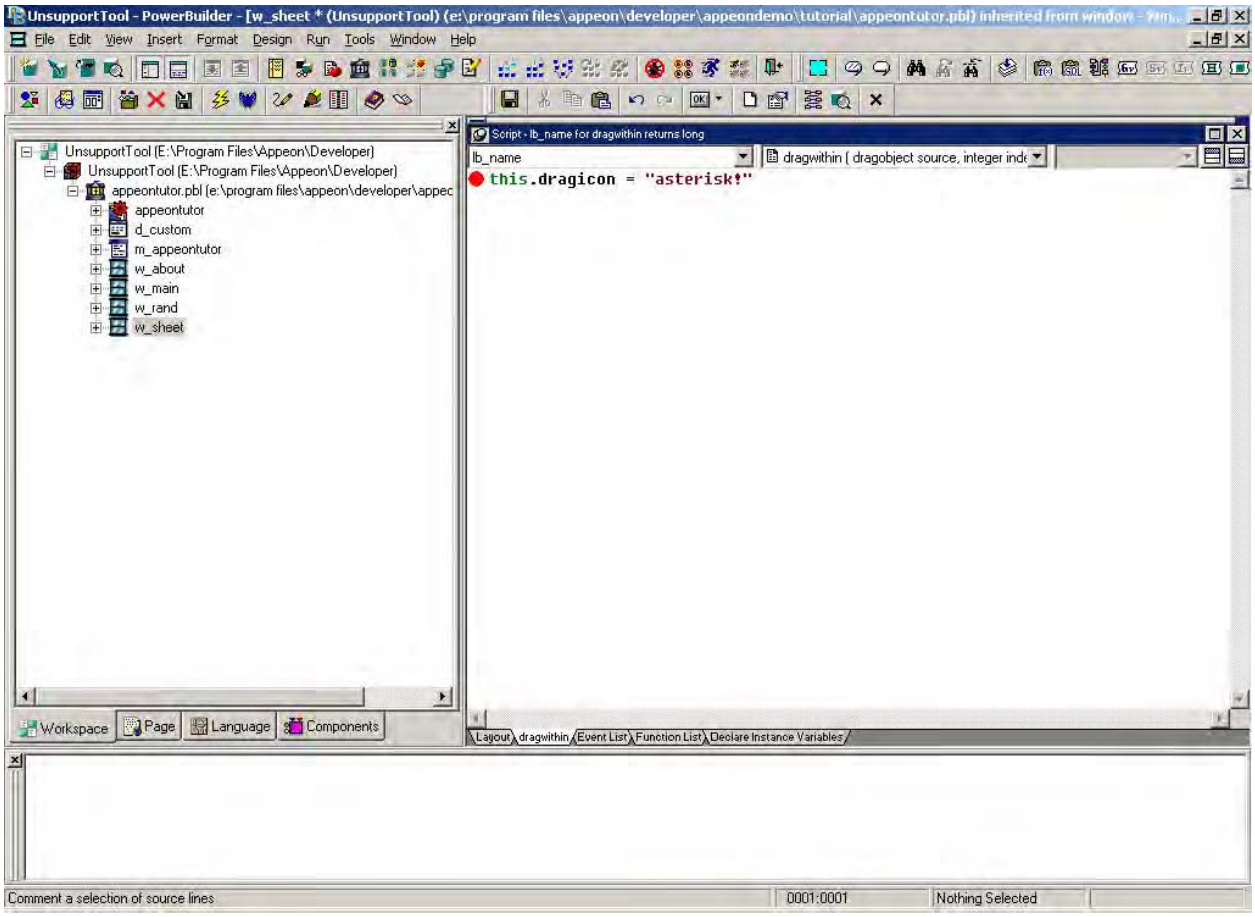
STEP 16 – In the UFA Report Window, expand *w_sheet* | *lb_name* | *dragwithin* and select dragwithin, as shown in Figure 5-71.

Figure 5-71: UFA Report Window



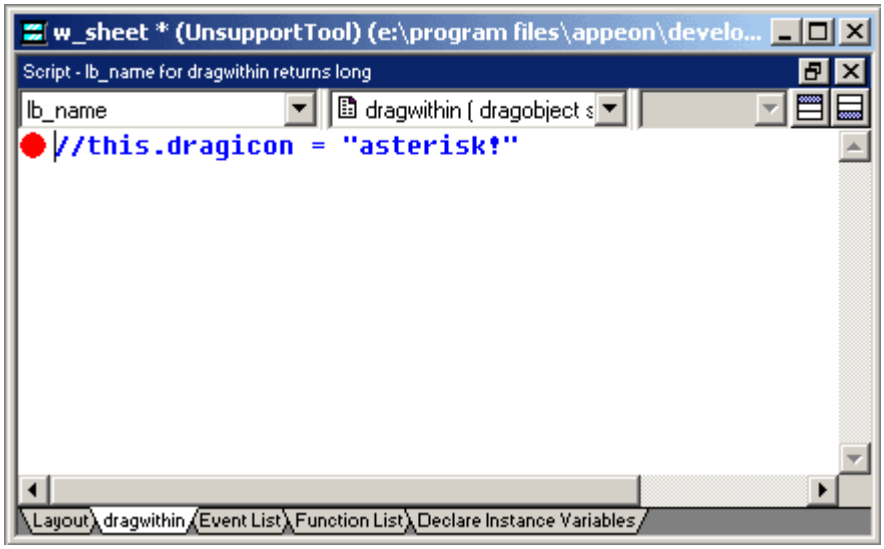
STEP 17 – Click the *Edit* button to enter the PowerBuilder Script view of the *DragWithin* event of the *lb_name* ListBox in the *w_sheet* Window, as shown in Figure 5-72.

Figure 5-72: Unsupported code in the PowerBuilder painter



STEP 18 – Click in the first line in the Script view, and then click the *Comment* button ( ) in PainterBar2.

Figure 5-73: Commenting unsupported script



STEP 19 – Click the *Save* button, and then click the *Close* button in PainterBar1.

STEP 20 – Click on the UFA tool icon on the status bar and select *Analyze selected objects* from the popup menu to analyze the tutorial application again and refresh the UFA Report.

STEP 21 – Make sure that no Drag N' Drop unsupported features are listed in the UFA Report.

To add additional buttons and script to perform the task of copying content from the DataWindow and pasting it into the ListBox:

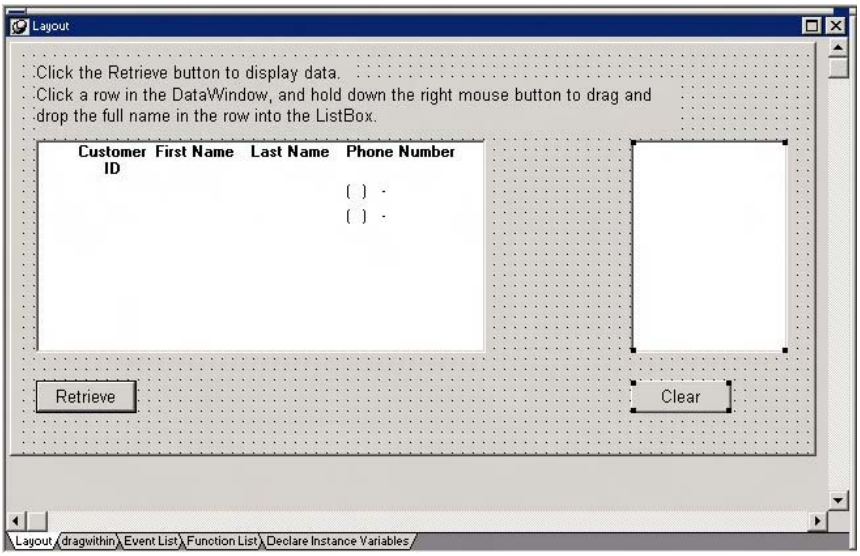
STEP 1 – Double click *w_sheet* Window in the system tree to open the Window painter for the *w_sheet* Window object.

The Window painter displays different views of the *w_sheet* Window object. The currently selected Layout view scheme may look different. To display the default Layout, select *View* | *Layouts* | *Default* from the PowerBuilder menu.

STEP 2 – Place the mouse pointer over the right edge of the *w_sheet* Window and the mouse pointer becomes a double arrow. Now hold down the left mouse button and drag to the right to increase the length of the *w_sheet* Window.

STEP 3 – Hold down the Ctrl key to select the *lb_name* ListBox and the Clear button. Then, press the right arrow key to move the two controls to the right, so that there is space between the DataWindow control and the ListBox control.

Figure 5-74: Layout

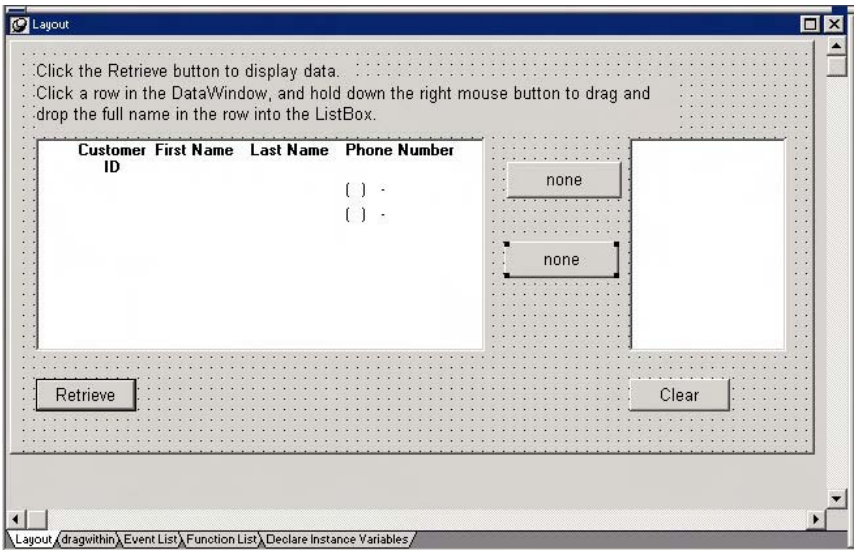


STEP 4 – Select *Insert* | *Control* | *CommandButton* from the PowerBuilder menu bar.

STEP 5 – Click between the DataWindow control and the ListBox control in the Layout view. A new CommandButton is produced.

STEP 6 – Repeat the above process to create another CommandButton just below the previously created CommandButton.

Figure 5-75: Layout



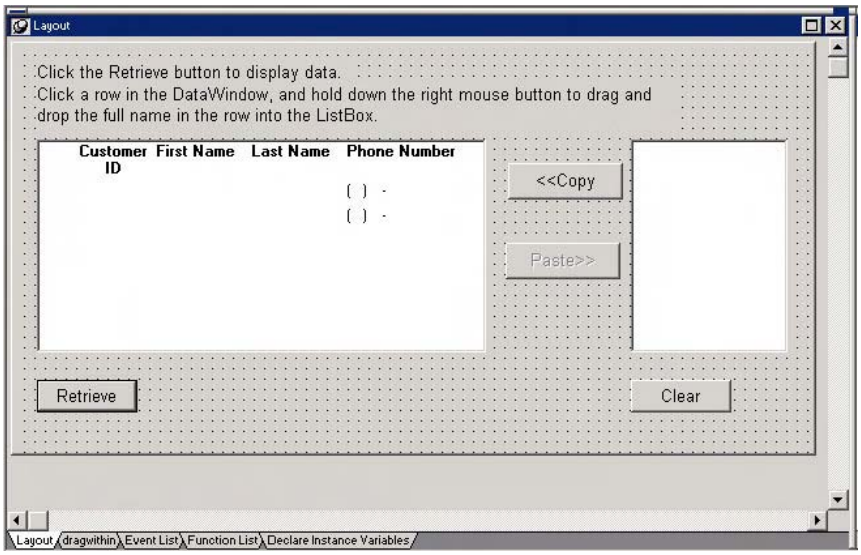
STEP 7 – Now change the properties of the two buttons you have just created:

Button Name	cb_copy	cb_paste
Button Text	"<<Copy"	"Paste>>"
Enabled	True	False

To change the properties of a CommandButton, click the button in the Layout view and edit its properties in the Properties view in the Windows painter.

After these modifications are made, the Layout view will look similar to Figure 5-76.

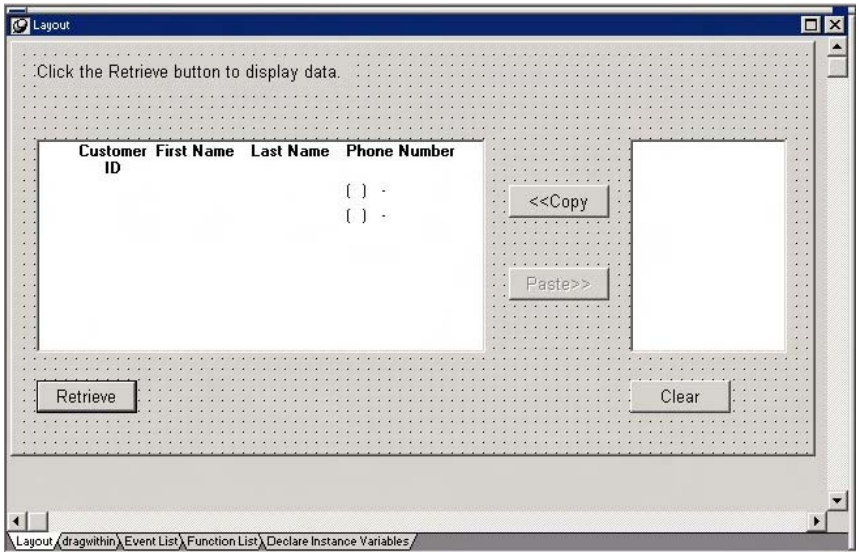
Figure 5-76: New layout



STEP 8 – Make additional small UI changes to the *w_sheet* Window.

1. Delete the second sentence ("Click a row in the DataWindow, and…") at the top of the *w_sheet* Window, since it no longer applies after the drag and drop feature has been removed.
2. Adjust the button sizes and positions, and align the controls properly to make the UI arrangement look acceptable.

Figure 5-77: Adjusted layout

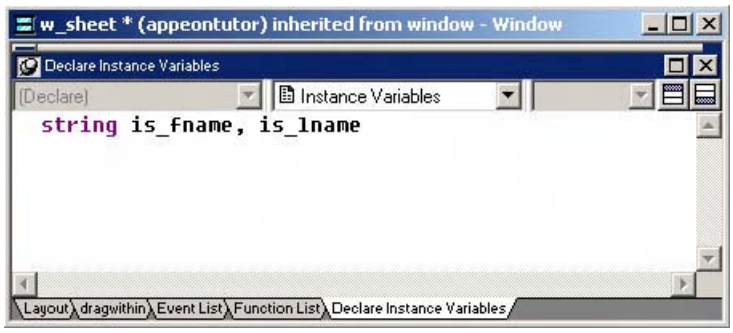


STEP 9 – Now write the necessary code to make the "<<Copy" and "Paste>>" buttons functional. Click the *Declare Instance Variables* tab to open the Variable view. Select *Instance Variables* from the second dropdown listbox. Type the following code in the script box:

```
string is_fname, is_lname
```

This line declares two instance strings: *is_fname* and *is_lname*. These two variables are used to store the value of the first name and the last name obtained from the current row in the DataWindow.

Figure 5-78: Declare instance variables

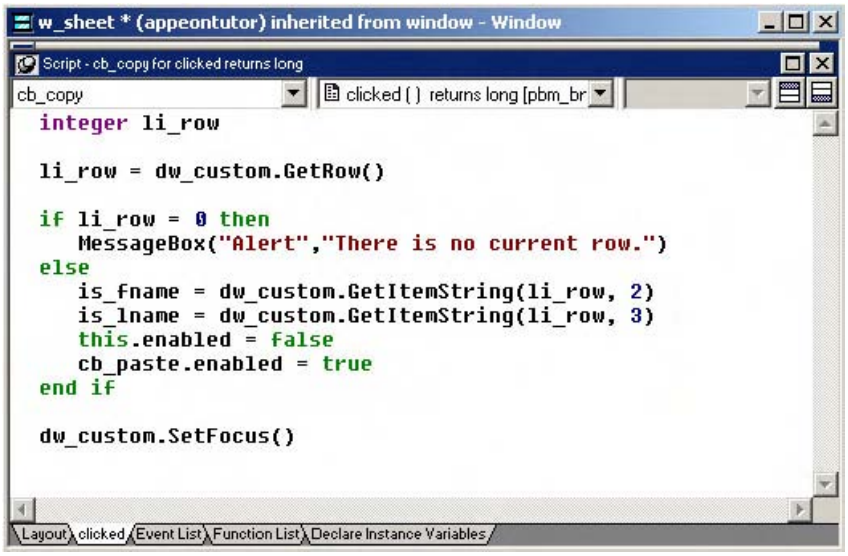


STEP 10 – Click the Layout tab. Double click the "<<Copy" CommandButton to open the Script view for the *Clicked* event of the *cb_copy* CommandButton.

STEP 11 – Type the following code in the script box:

```
integer li_row
li_row = dw_custom.GetRow()
if li_row = 0 then
   MessageBox("Alert","There is no current row.")
else
   is_fname = dw_custom.GetItemString(li_row, 2)
   is_lname = dw_custom.GetItemString(li_row, 3)
   this.enabled = false
   cb_paste.enabled = true
end if
dw_custom.SetFocus()
```

Figure 5-79: Script in cb_copy Clicked event



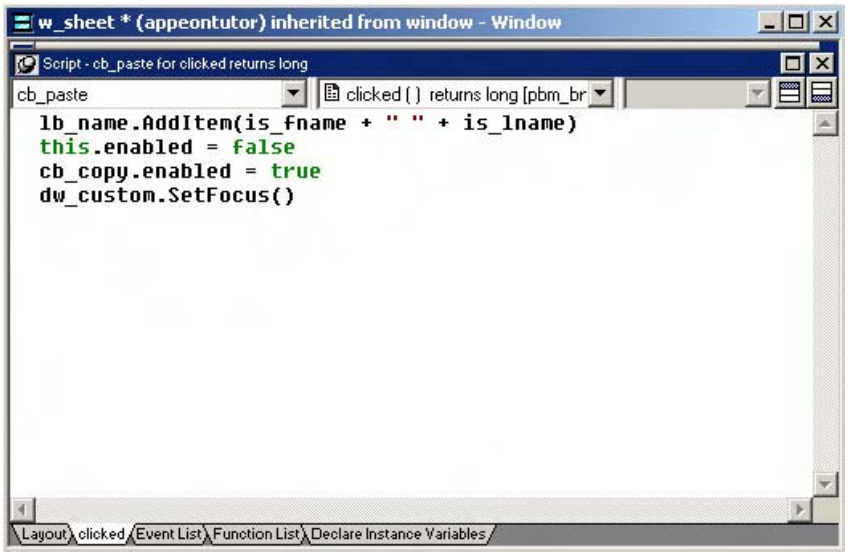
When the "<<Copy" button is clicked, the current row number in the *dw_custom* is saved into the *li_row* local integer. The DataWindow function *GetItemString* is then used to read the first name and the last name in the current row, and assign them to the two instance strings: *is_fname* and *is_lname*.

STEP 12 – Click the Layout tab again. Double click the "Paste>>" CommandButton to open the Script view for the *Clicked* event of the *cb_paste* CommandButton.

STEP 13 – Type the following code in the script box:

```
lb_name.AddItem(is_fname + " " + is_lname)
this.enabled = false
cb_copy.enabled = true
dw_custom.SetFocus()
```

Figure 5-80: Script in cb_paste Clicked event



When the "Paste>>" button is clicked, the script concatenates the first and last name obtained from the DataWindow to form a full name, then adds the full name to the *lb_name* ListBox.

STEP 14 – Click the *Save* button and then click the *Close* button in PainterBar1.

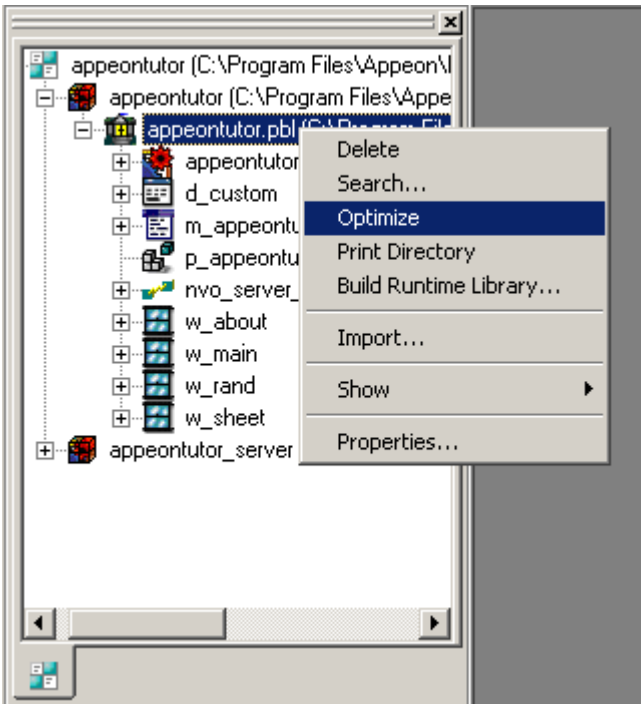
5.3 Optimizing and full build

Before any PowerBuilder application is converted by Appeon, ensure the PBLs have been optimized. You must also "full build" the PowerBuilder application successfully before Appeon will be able to convert it.

To optimize and full build the tutorial application:

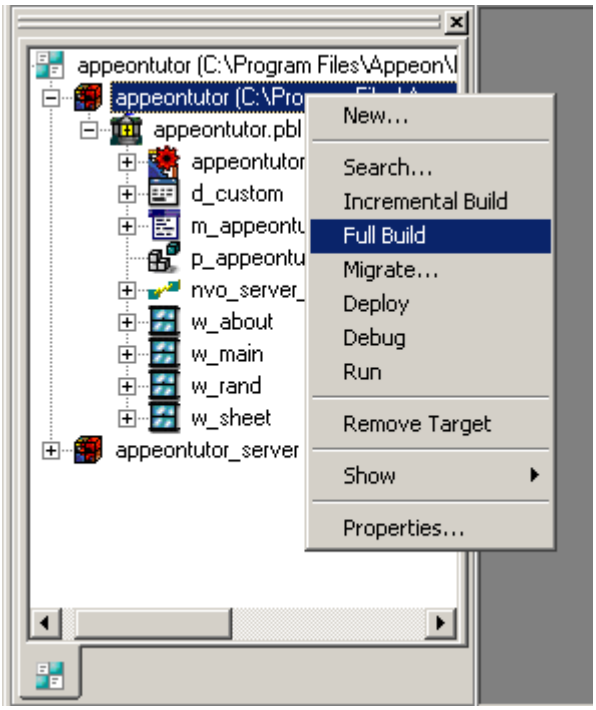
STEP 1 – Right click on *appeontutor.pbl* in the PowerBuilder system tree. Click *Optimize*, as shown in Figure 5-81.

Figure 5-81: Optimize



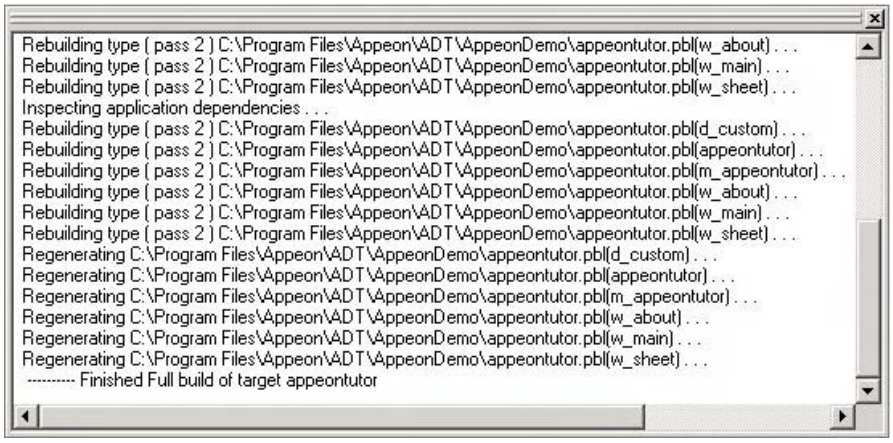
STEP 2 – Right click on the *appeontutor* Target in the system tree and select *Full Build* from the context menu, as shown in Figure 5-82.

Figure 5-82: Full Build



STEP 3 – The full build process begins. Some information is displayed in the Output window. Make sure that when the full build is complete, no error messages occur in the Output window, as shown in Figure 5-83.

Figure 5-83: Finished Full Build of target appeontutor



6 Deploying the Tutorial PowerBuilder Application

In the previous lessons, the Appeon tutorial PowerBuilder application has undergone careful analysis for unsupported features. The unsupported features have been worked around or removed, and the necessary configurations have been made. Now you will launch the Appeon Deployment Wizard to convert the tutorial application onto the Web automatically.

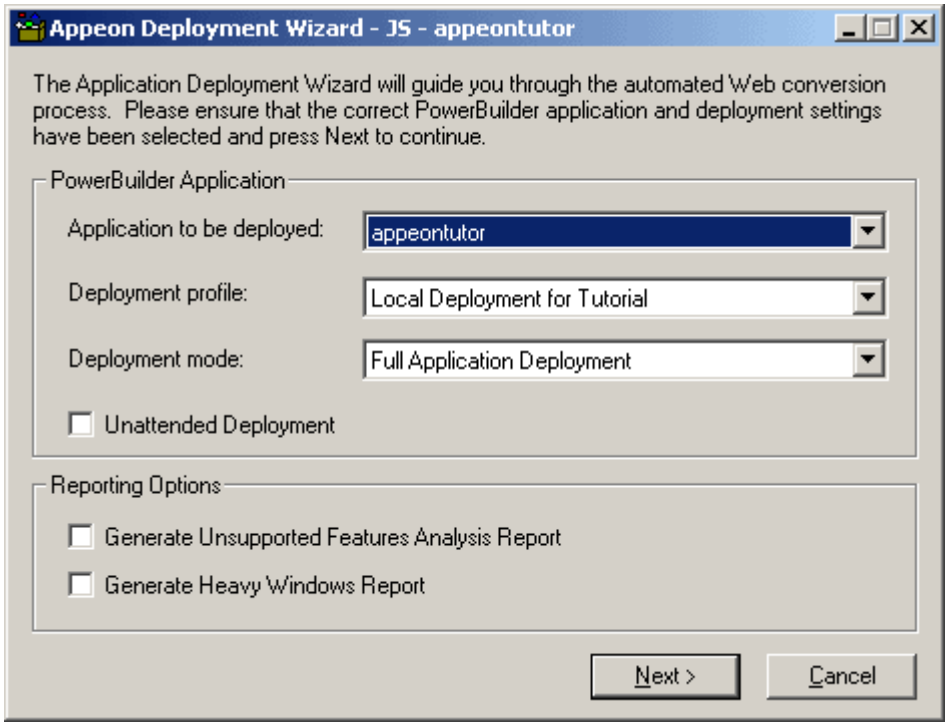
To deploy the tutorial application onto the Web automatically:

STEP 1 – Start Appeon Server from the Windows shortcut. If Appeon Server is already running, shut it down and restart it.

To start Appeon Server, choose *Programs* | *Appeon 3.1 for PowerBuilder* | *Jaguar Server* from the Windows Start menu. Wait until the text line displays "Accepting connections". This means Appeon Server has been started successfully.

STEP 2 – Within the PowerBuilder IDE, click the *Deploy* button ( ) in the Appeon Developer toolbar. The Appeon Deployment Wizard starts.

Figure 6-1: Appeon Web Application Deployment Wizard



The start page of the Wizard displays the following information:

- Application to be deployed – The Current Application (appeontutor) is selected by default from the list of applications that are defined in Appeon Developer application profiles.
- Deployment profile – Make sure the deployment profile, *Local Deployment for Tutorial profile*, is selected.
- Deployment mode – Since it is the first time the tutorial application will be deployed, the *Full Application Deployment* mode will be selected by default. For subsequent deployments, the *Incremental Application Deployment* will be selected by default.

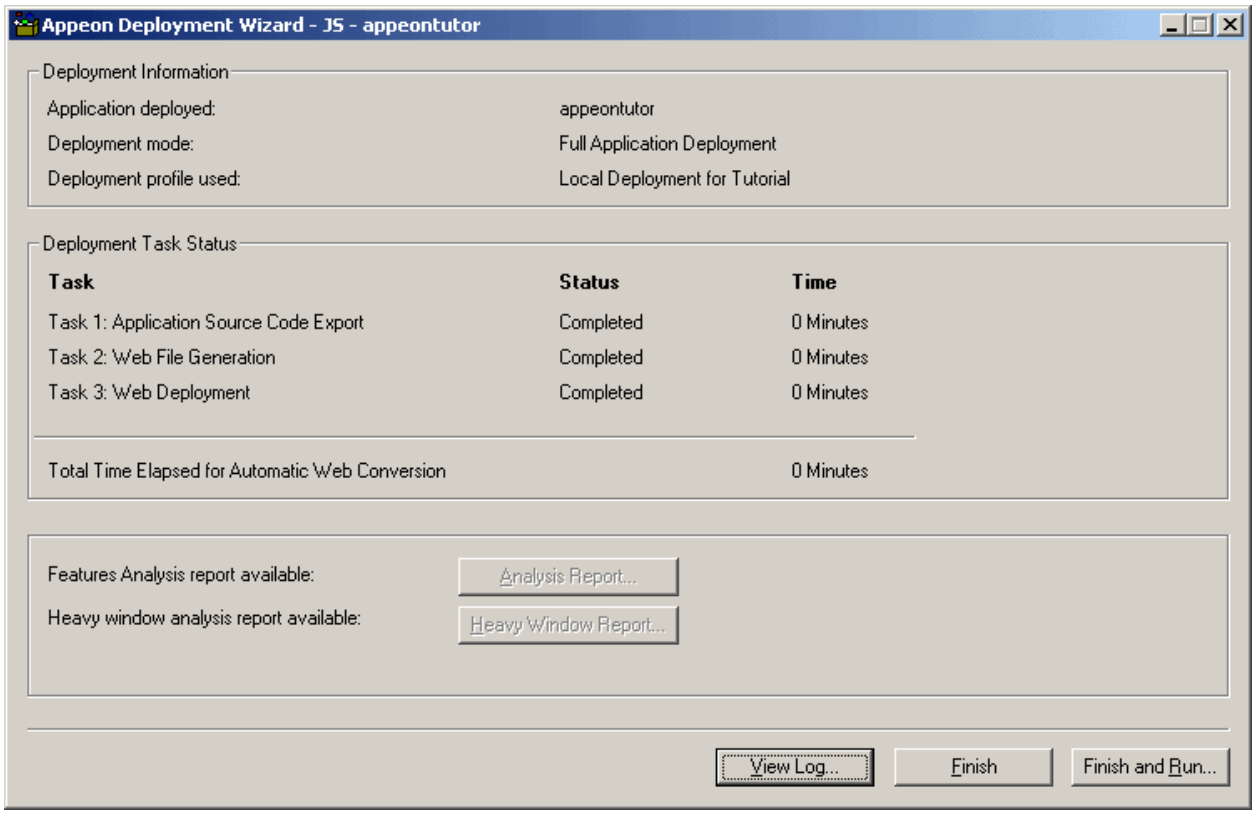
- Unattended Deployment – Check this option so that the entire deployment process is automatic.

STEP 3 – Click *Next* to start the automatic Web deployment process. The Appeon Deployment Wizard will perform three tasks:

1. PowerBuilder application source code export and analysis. (Task 1: Application Source Code Export)
2. Parsing of the PowerBuilder source code (PBL files) into Web application files that are stored on the local machine. (Task 2: Web File Generation)
3. Transfer of local Web files to the Web Server by using either file copy or FTP, and uploads DataWindows to the Appeon Server. (Task 3: Web Deployment)

When the deployment process is complete, the report dialog box is displayed.

Figure 6-2: Appeon Web Application Deployment Wizard



STEP 4 – Click *Finish*.

7 Configuring Database Connection

When the tutorial PowerBuilder application is deployed to the Web, Appeon Server handles the database connection using connection caches rather than transaction objects defined in the PowerBuilder application. After deployment, you must first create proper connection caches in EAServer Manager, and then associate the transaction objects used by the Appeon tutorial PowerBuilder application with the created connection caches using Appeon Enterprise Manager.

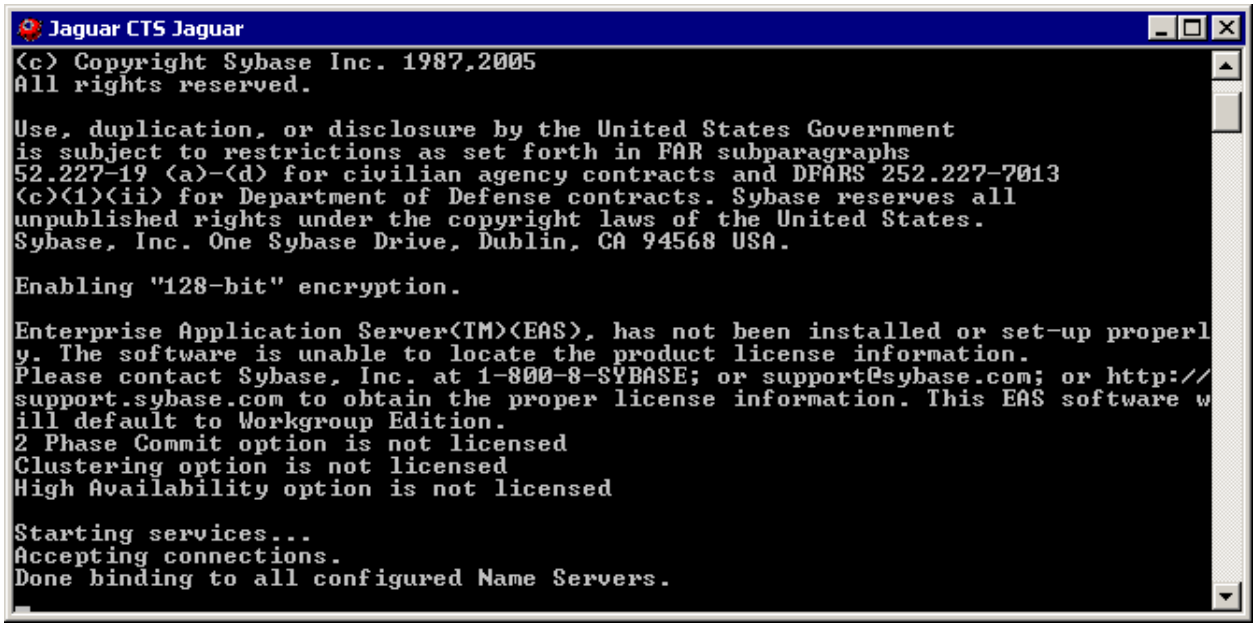
In this section, you will learn how to:

- Login to Appeon Server using EAServer Manager
- Create a connection cache
- Launch and log on to AEM
- Add transaction object mappings in AEM

7.1 Logging in Appeon Server

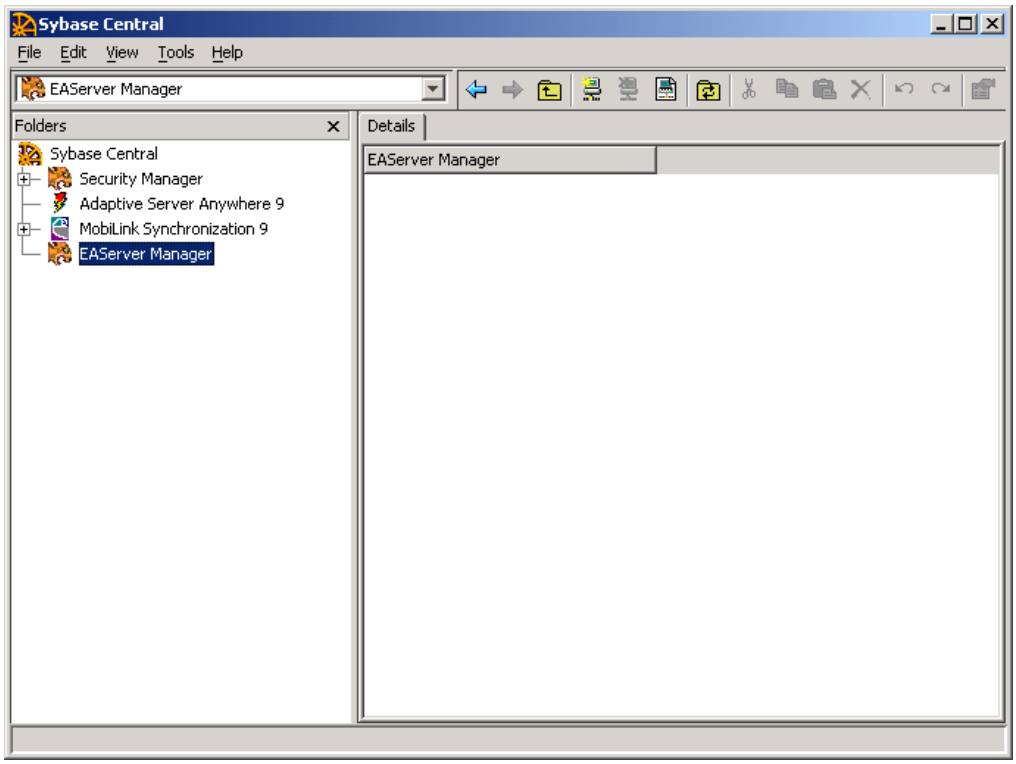
STEP 1 – In Windows: select *Start* | *Programs* | *Appeon 3.1 for PowerBuilder* | *Jaguar Server*. Wait until the "Accepting connections" text line is displayed, as shown in Figure 7-1. This indicates that Appeon Server is ready for use.

Figure 7-1: Appeon Server



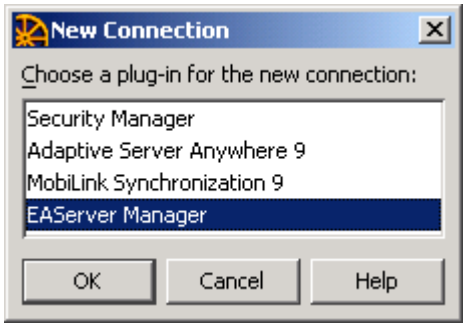
STEP 2 – In Windows: select *Start* | *Programs* | *Appeon 3.1 for PowerBuilder* | *EAServer Manager* from the Windows Start menu. Sybase Central starts, as shown in Figure 7-2.

Figure 7-2: Sybase Central



STEP 3 – Click the *Connect* button ( ) as shown in Figure 7-2, and the New Connection dialog box appears, as shown in Figure 7-3. Select EAServer Manager and click *OK*.

Figure 7-3: New connection

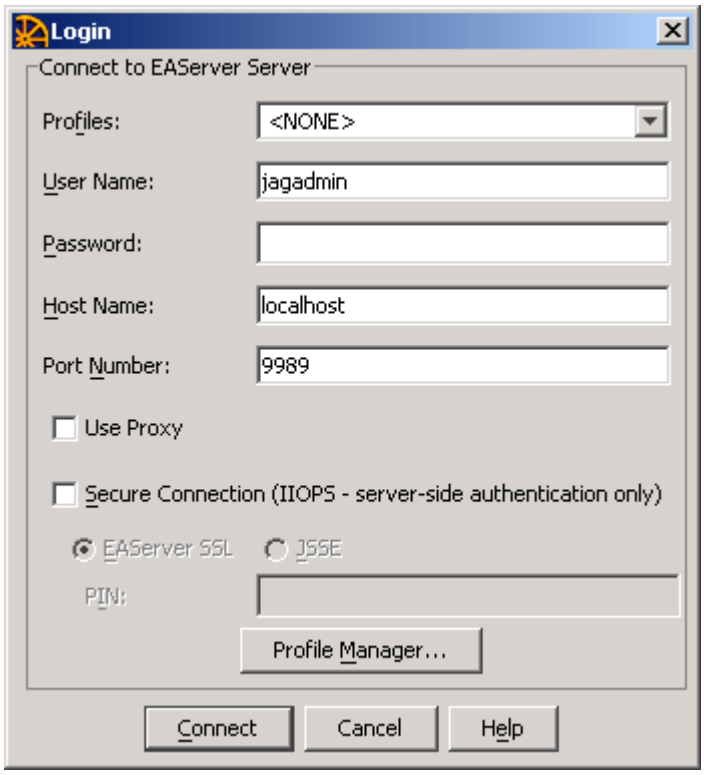


STEP 4 – Type *localhost* as the Host Name and *9989* for the Port Number in the Login dialog box that displays.

The Appeon Server default user name is *jagadmin* and there is no password. If you have changed the default, enter the correct user name and password, as shown in Figure 7-4.

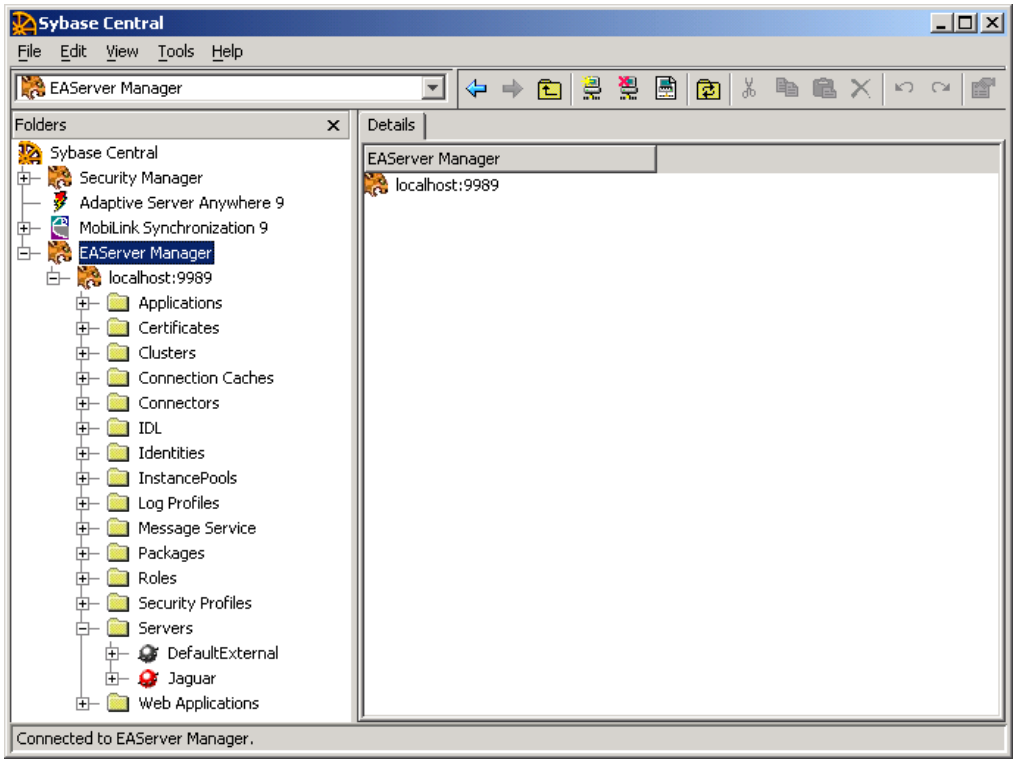
Click *Connect*.

Figure 7-4: Login



EAServer is successfully logged in, as shown in Figure 7-5.

Figure 7-5: Logged in



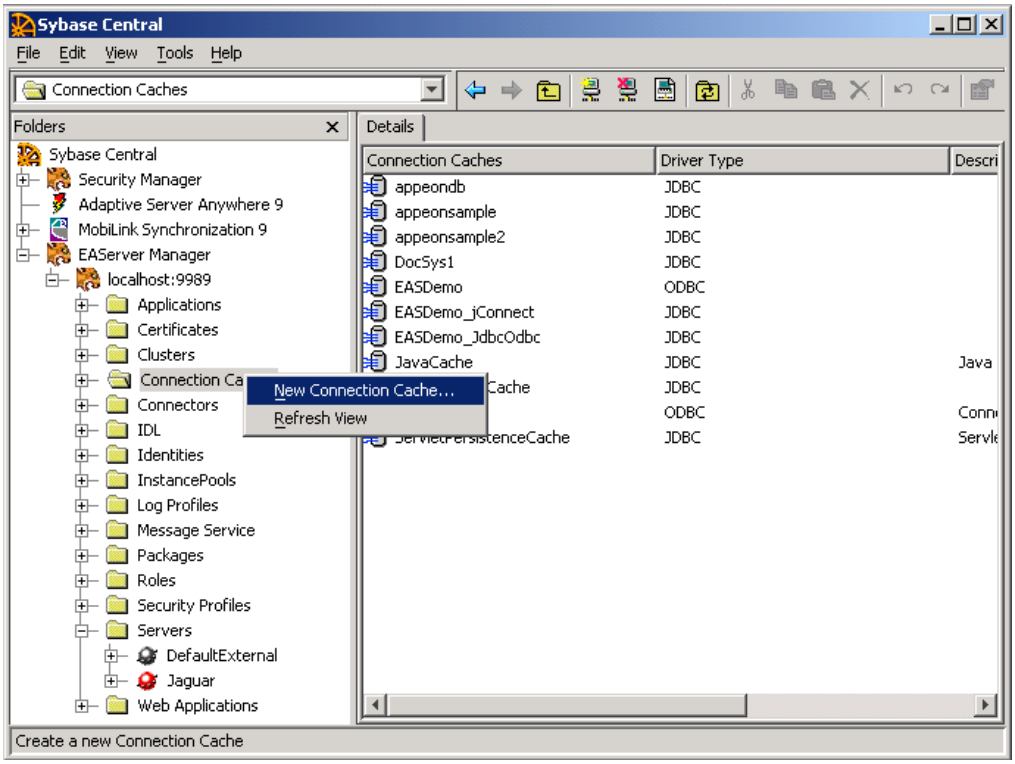
7.2 Creating a connection cache

The following sections show you how to create an EAServer JDBC connection cache that links to the *appeontutor* ODBC data source using the Sun JDBC driver.

To create a connection cache using the Sun JDBC driver:

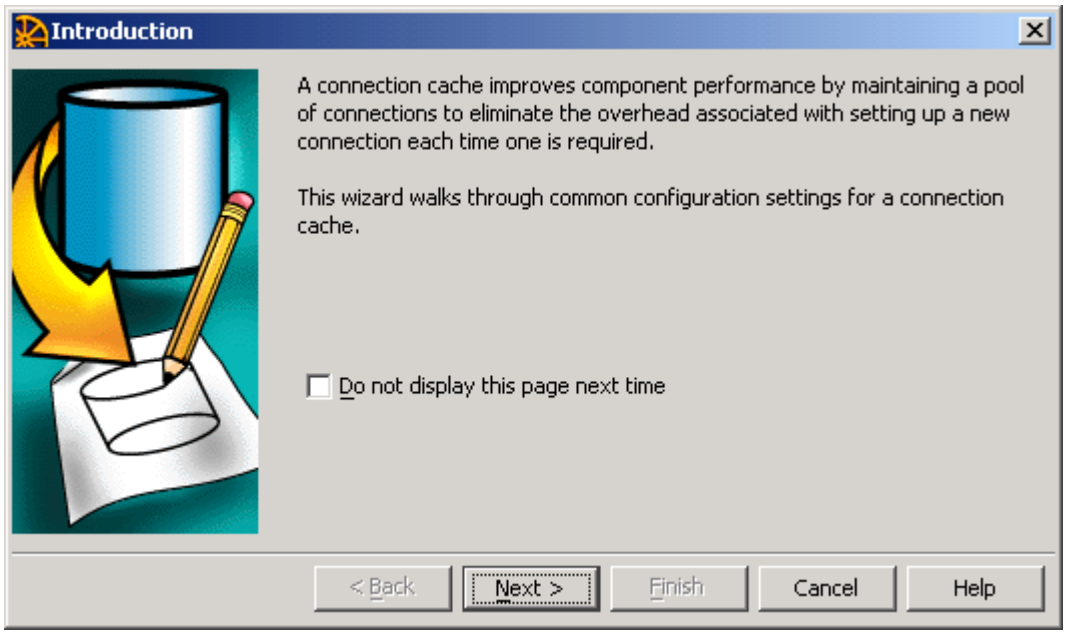
STEP 1 – Navigate to the *Connection Caches* folder under EAServer Manager in the left system tree, right-click on the folder, and select *New Connection Cache* from the popup menu, as shown in Figure 7-6.

Figure 7-6: New Connection Cache



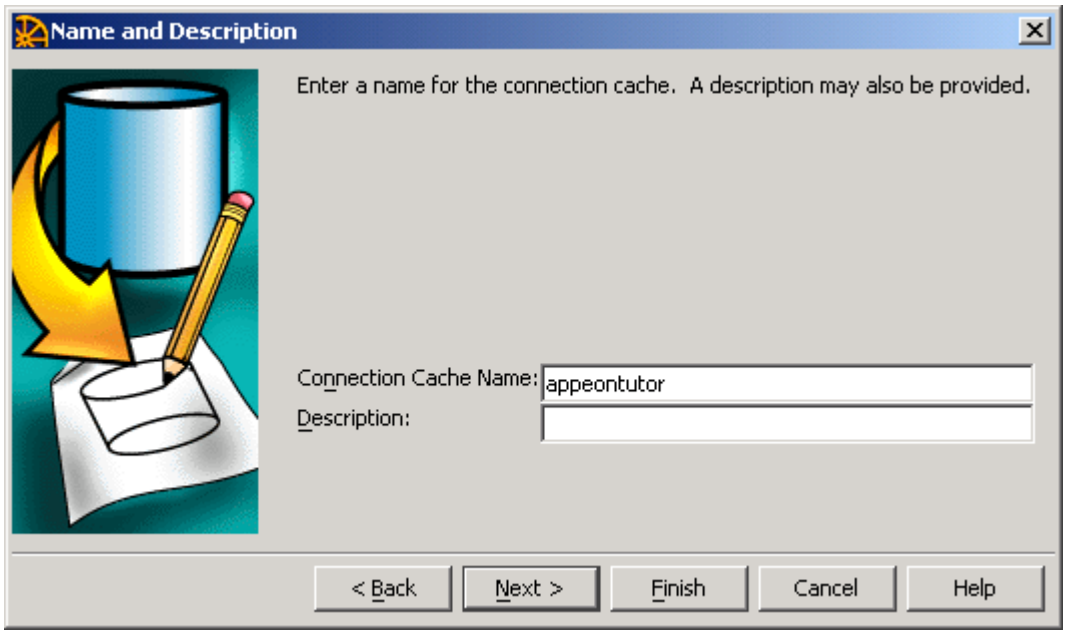
STEP 2 – In the Introduction dialog box that displays, click *Next* to continue, as shown in Figure 7-7.

Figure 7-7: Introduction dialog box



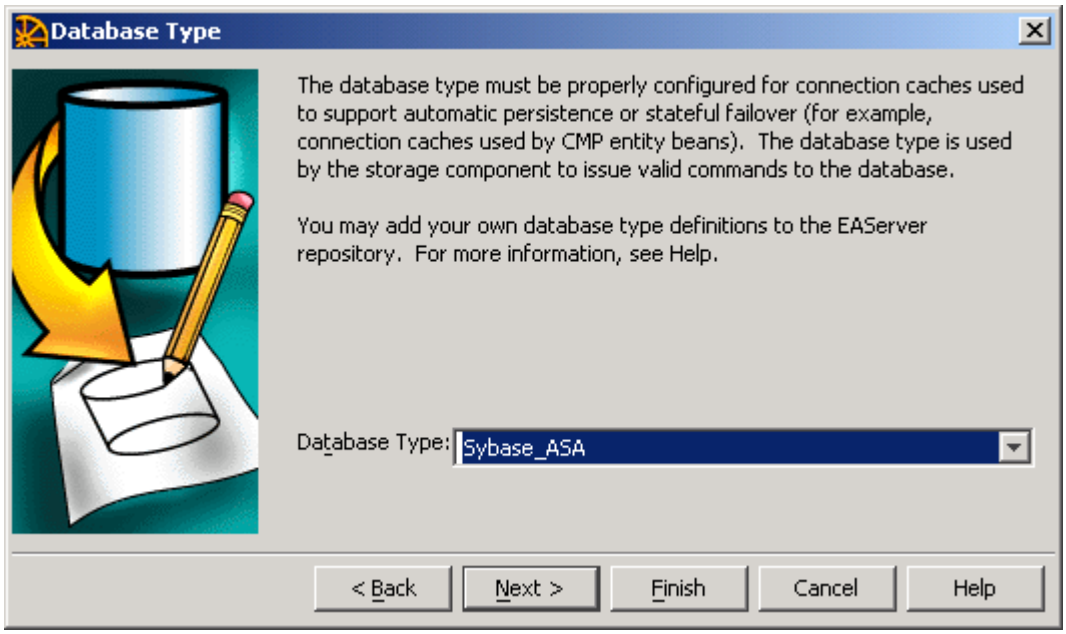
STEP 3 – In the Name and Description dialog box that displays, type *appeontutor* and click *Next* to continue, as shown in Figure 7-8.

Figure 7-8: Name and Description dialog box



STEP 4 – In the Database Type dialog box that displays, select *Sybase_ASA* from the Database Type dropdown listbox and click *Next* to continue, as shown in Figure 7-9.

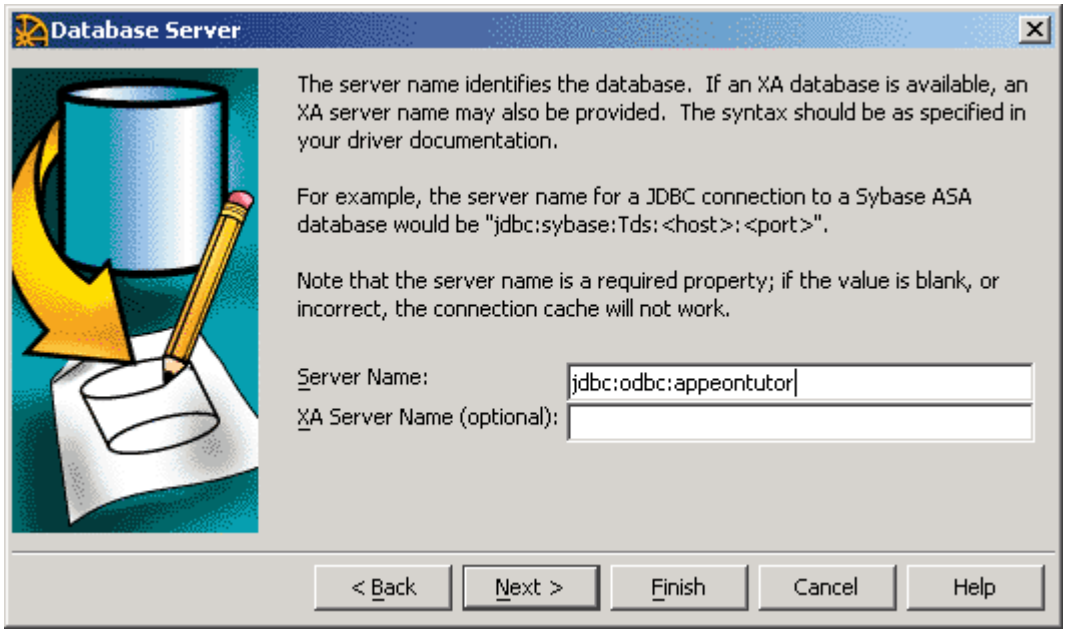
Figure 7-9: Database Type dialog box



STEP 5 – In the Database Server dialog box that displays, type *jdbc:odbc:appeontutor* in the *Server Name* field and click *Next* to continue, as shown in Figure 7-10. This will connect to the *appeontutor* ODBC data source that was created in Section 3.3: *Configuring ODBC data source*.

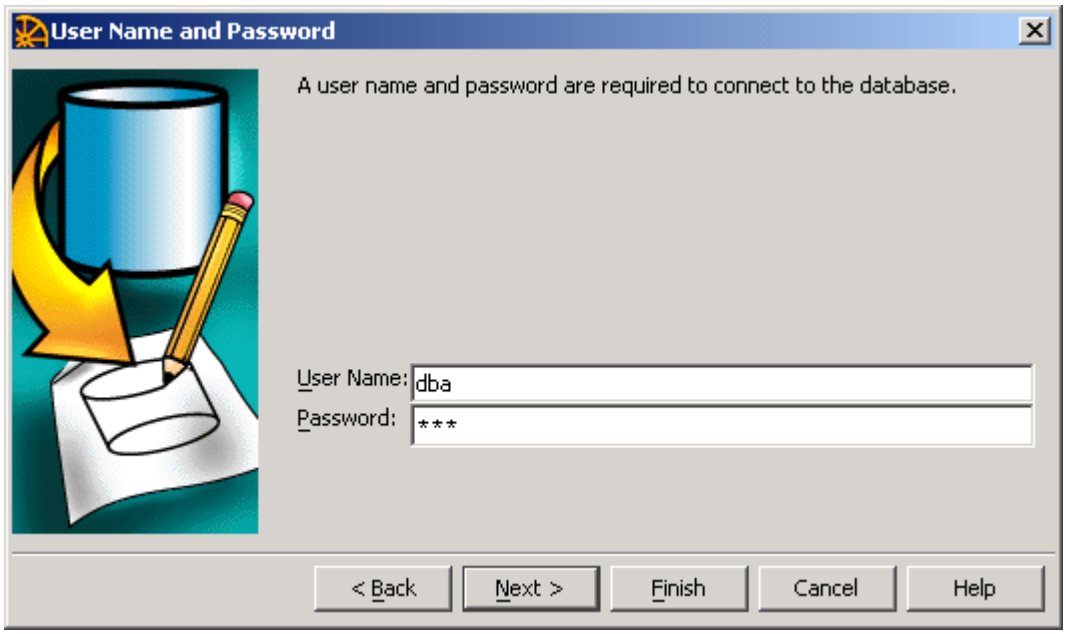
The server name syntax format is: jdbc:odbc:*DSNName*. *DSNName* indicates the data source name. Note that the server name is case sensitive.

Figure 7-10: Database Server dialog box



STEP 6 – In the User Name and Password dialog box, type *dba* in the *User Name* field and *sql* in the *Password* field, and click *Next* to continue, as shown in Figure 7-11.

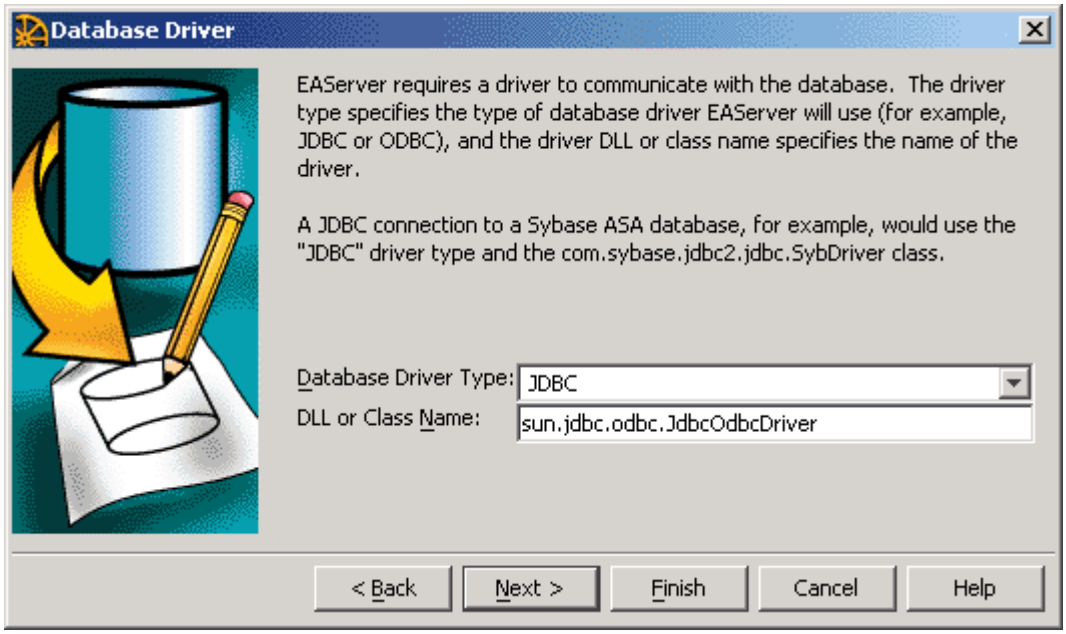
Figure 7-11: User Name and Password dialog box



STEP 7 – In the Database Driver dialog box, select *JDBC* from the *Database Driver Type* dropdown listbox, type *sun.jdbc.odbc.JdbcOdbcDriver* in the *DLL or Class Name* field, and click *Next* to continue, as shown in Figure 7-12.

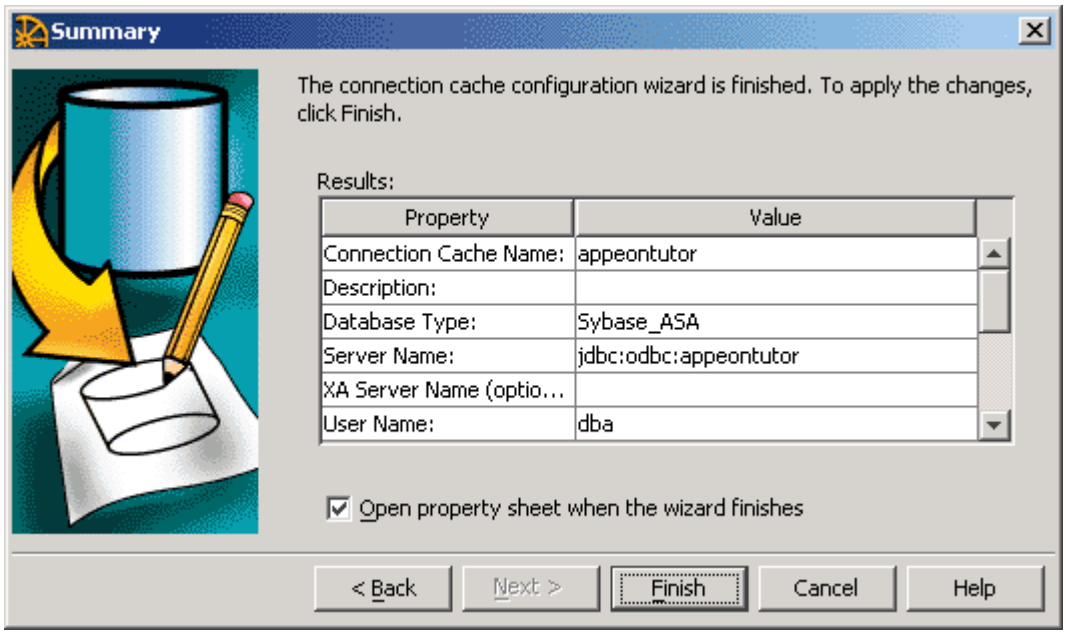
The typed string is case sensitive. Be careful not to leave leading or ending spaces.

Figure 7-12: Database Driver



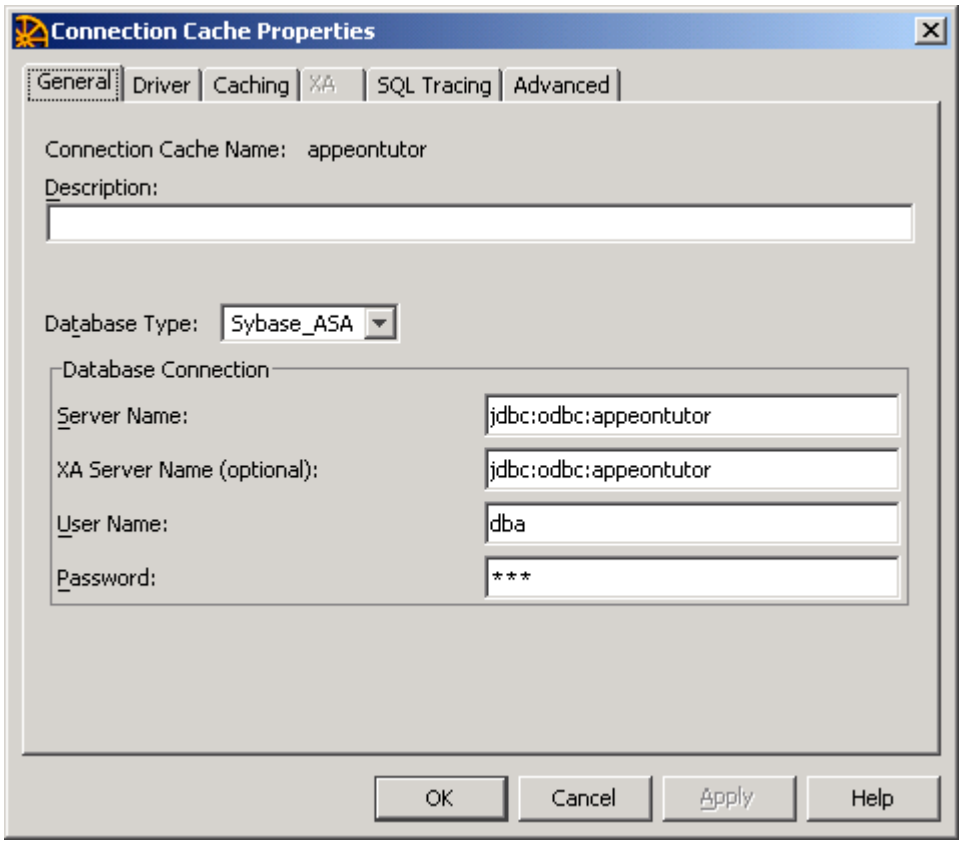
STEP 8 – Click *Next* until you get to the Summary dialog box and click *Finish* to complete the connection cache creation, as shown in Figure 7-13.

Figure 7-13: Summary dialog box



STEP 9 – Click *OK* to close the Connection Cache Properties dialog box, as shown in Figure 7-14.

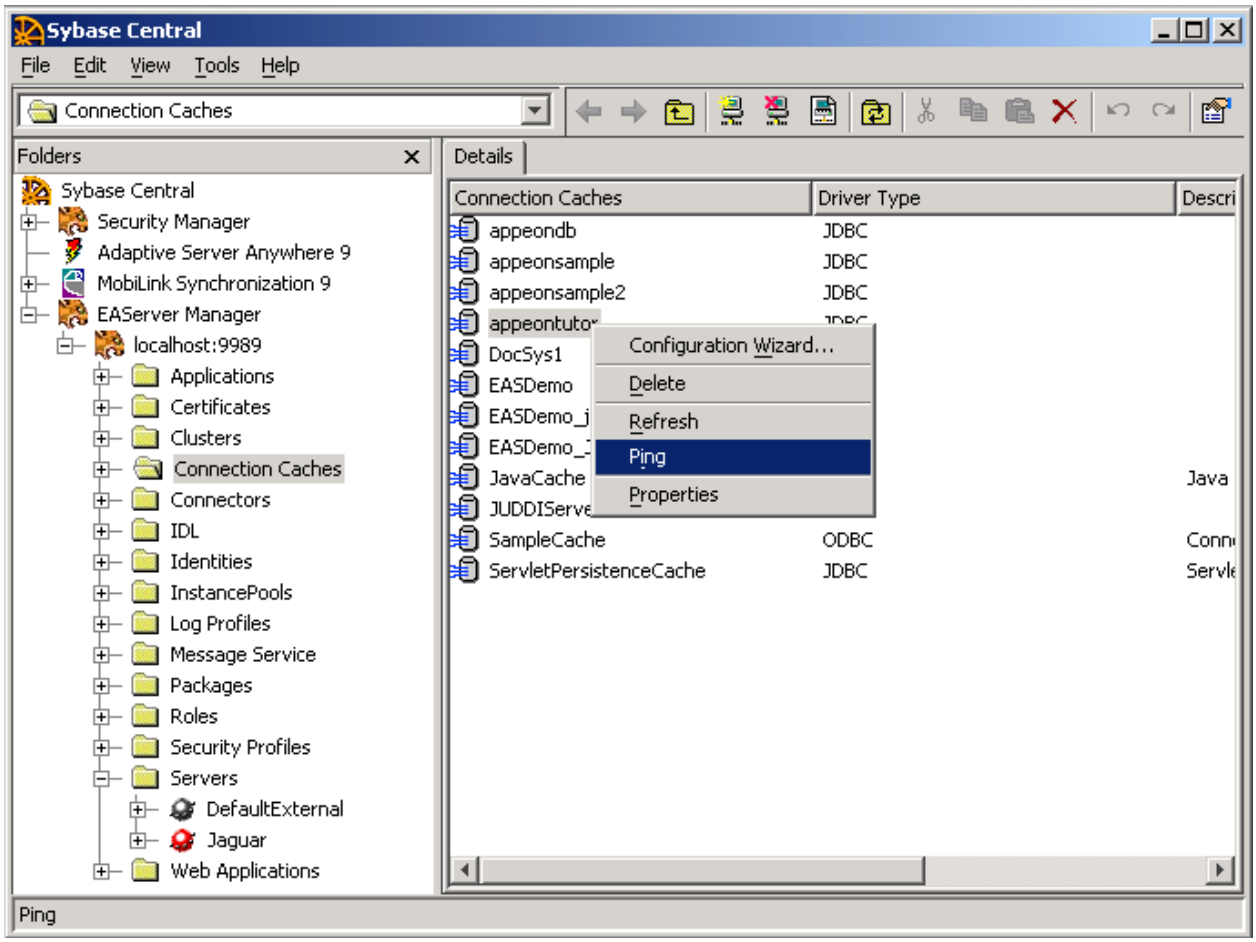
Figure 7-14: Connection Cache Properties dialog box



The *appeontutor* connection cache has been added, as shown in Figure 7-15.

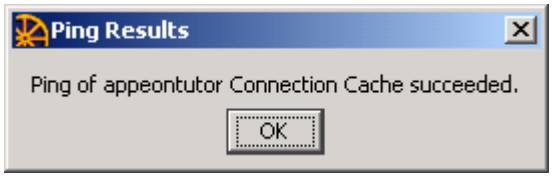
STEP 10 – Right click *appeontutor* connection cache and select *Ping* from the popup menu to test it, as shown in Figure 7-15.

Figure 7-15: Ping



STEP 11 – If the Ping succeeds, click *OK* to close the Ping Results message box, as shown in Figure 7-16.

Figure 7-16: Ping Results



STEP 12 – Close the Sybase Central window and shut down Appeon Server.

For more details on how to use the Sun JDBC driver for connecting ASA or ASE databases, refer to Section 8.2: *Setting up Appeon Server connection caches* in the *Appeon Migration Guide*.

7.3 Launching and logging in AEM

Appeon Enterprise Manager (AEM) is the Web based application that provides the portal to configure and maintain Appeon Web applications and the entire Appeon system.

To launch and login AEM:

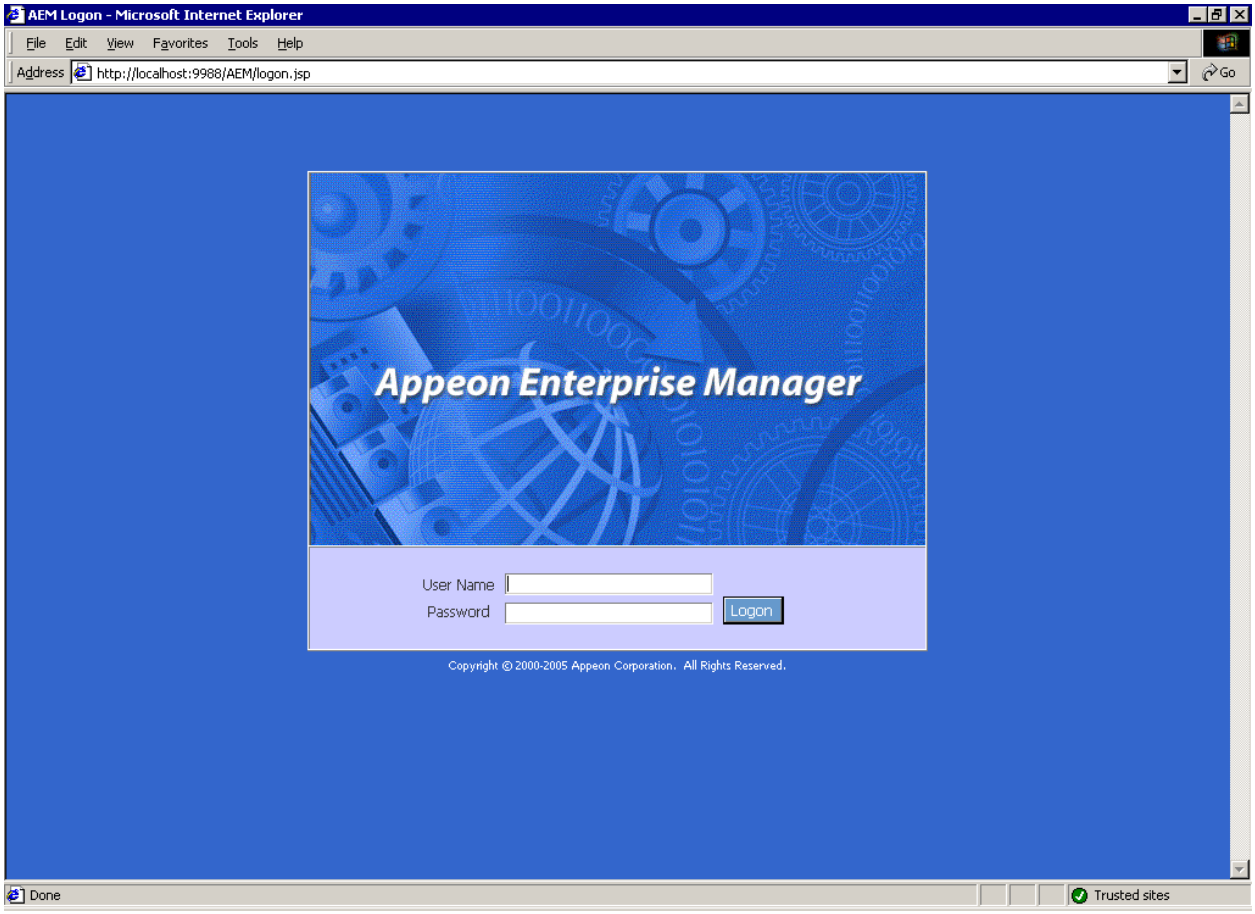
STEP 1 – Start Appeon Server using the Windows shortcut. If Appeon Server is already running, shut it down and restart it.

To start Appeon Server: in Windows, select *Start* | *Programs* | *Appeon 3.1 for PowerBuilder* | *Jaguar Server*. Wait until the "Accepting connections" text line is displayed. This means Appeon Server has been started successfully.

STEP 2 – In Windows, select *Start* | *Programs* | *Appeon 3.1 for PowerBuilder* | *Appeon Enterprise Manager*, or click the *AEM* button in the Appeon Developer toolbar.

The AEM login page is loaded into a new browser window, as shown in Figure 7-17.

Figure 7-17: AEM Login



STEP 3 – Type *admin* for the username and *admin* for the password and click *Logon*, and the AEM Welcome page is displayed, as shown in Figure 7-18.

Figure 7-18: Appeon Enterprise Manager

Note: The username and password - both “admin” - are the default values to log on to AEM. If the username and the password have been changed, enter the new username and password on the AEM login page.

7.4 Adding transaction object mappings in AEM

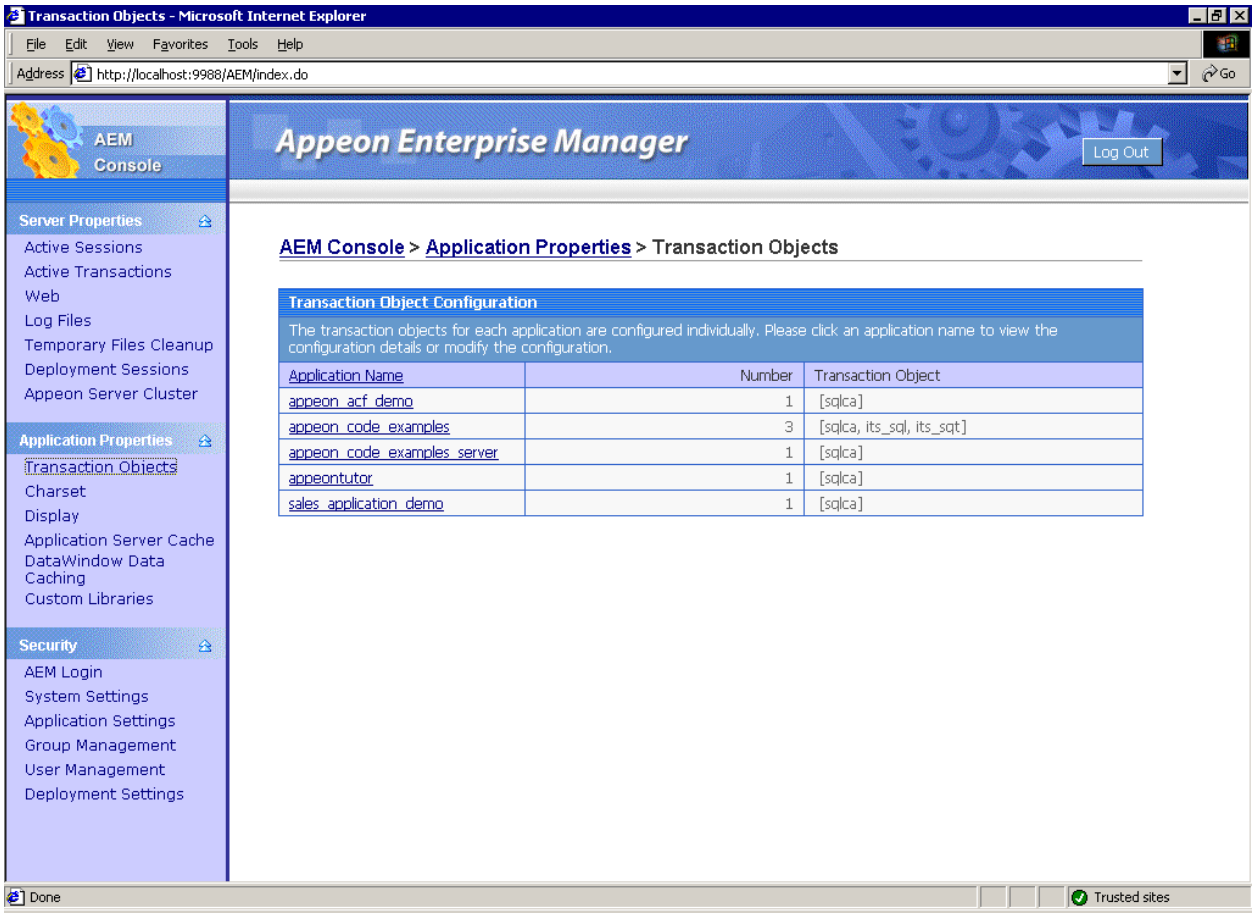
When the Appeon tutorial PowerBuilder application is deployed to Appeon Server, AEM automatically adds the application name (appeontutor) into the application list of the Transaction Objects tool. You can now configure the transaction object mapping for the appeontutor Web Application.

To configure Transaction object mapping for the Web application:

STEP 1 – Expand the *Application Properties* node in the left navigation frame.

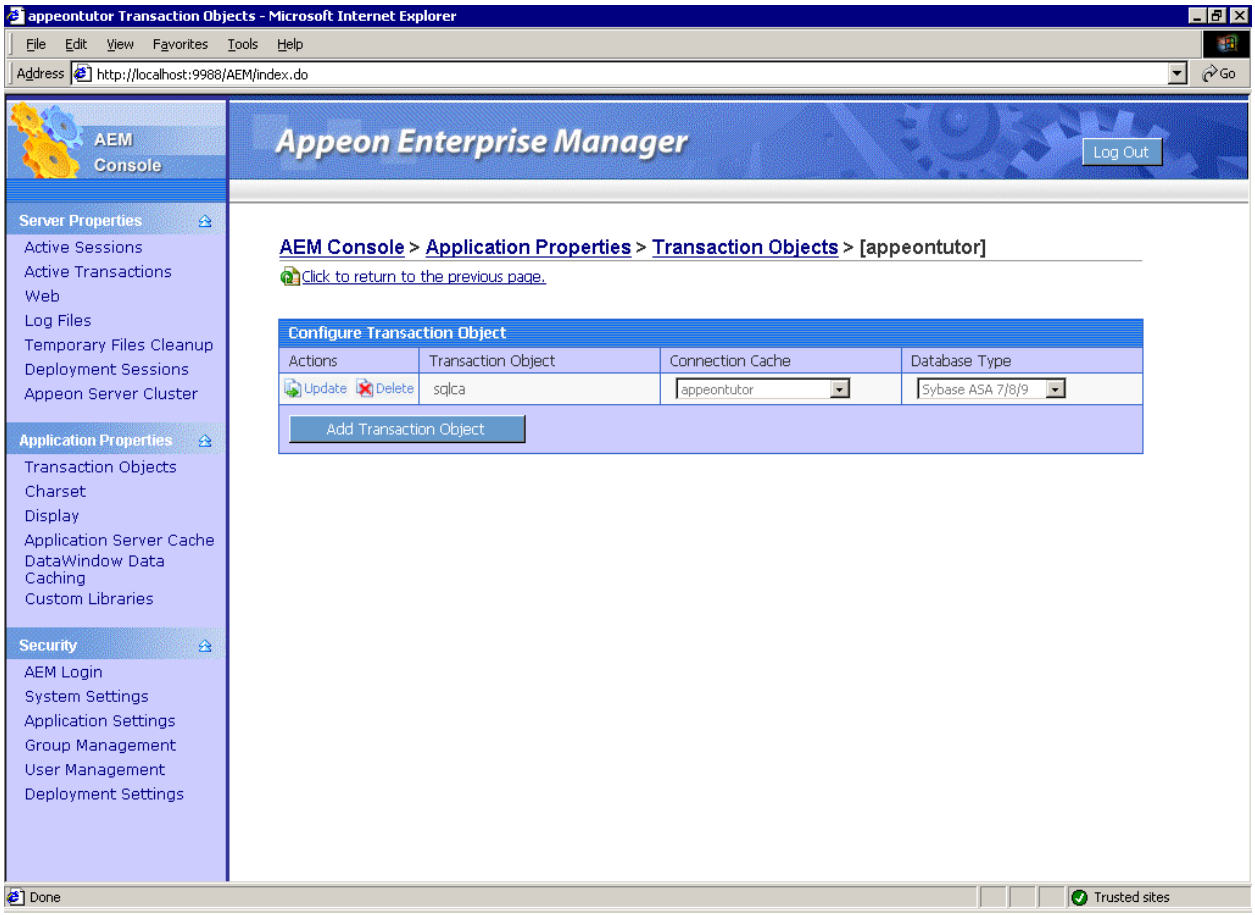
STEP 2 – Click *Transaction Objects* under *Application Properties*. The Transaction Objects page is displayed in the right frame, as shown in Figure 7-19.

Figure 7-19: Transaction Objects



STEP 3 – Click the *appontutor* in the Transaction Object Configuration table. The appeontutor page is displayed, as shown in Figure 7-20.

Figure 7-20: Transaction Object Configuration



STEP 4 – Select *appeontutor* from the Connection Cache dropdown listbox, select *Sybase ASA 7/8/9* from the Database Type dropdown listbox, and click *Update* to save the settings.

The Transaction Object has now been configured.

STEP 5 – Click *Log Out* (

).

STEP 6 – Close the AEM browser window and shut down Appeon Server.

8 Running the Web Application

The Appeon tutorial PowerBuilder application has been converted into a standard Web application with proper database connection which can now be run.

To launch the Web application after the tutorial application has been converted:

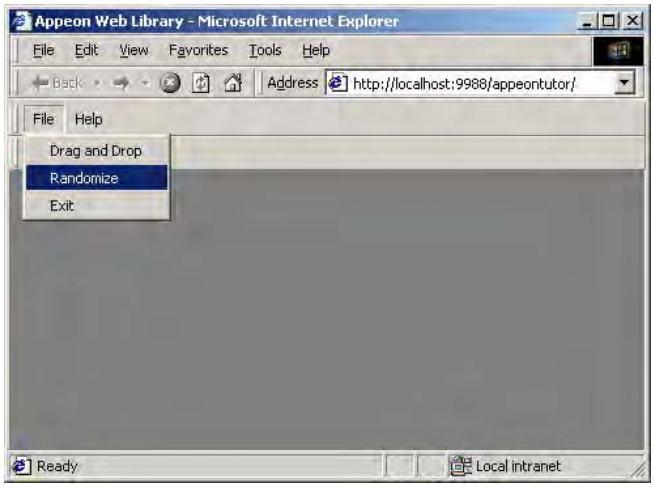
STEP 1 – Make sure Appeon Server is running.

To start Appeon Server, choose *Programs* | *Appeon 3.1 for PowerBuilder* | *Jaguar Server* from the Windows Start menu. Wait until the text line displays "Accepting connections". This means Appeon Server has been started successfully.

STEP 2 – In the PowerBuilder IDE, click the *Run* button () in the Appeon Developer toolbar and select the *appeontutor* application in the popup window, or manually open a browser window and type "http://localhost:9988/appeontutor/" into the Address bar and press Enter.

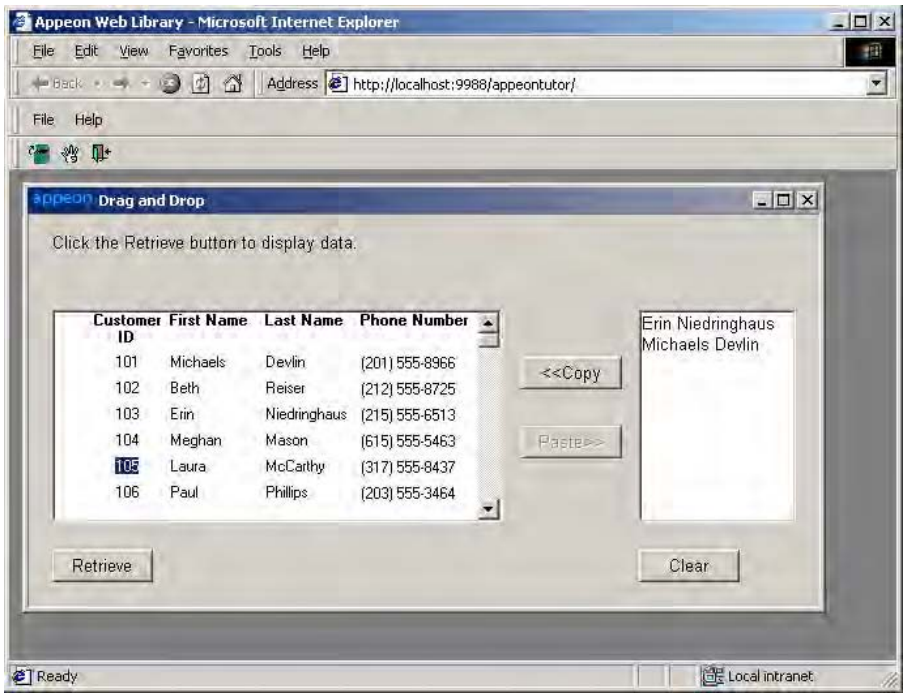
The Web application starts, as shown in Figure 8-1.

Figure 8-1: Randomize



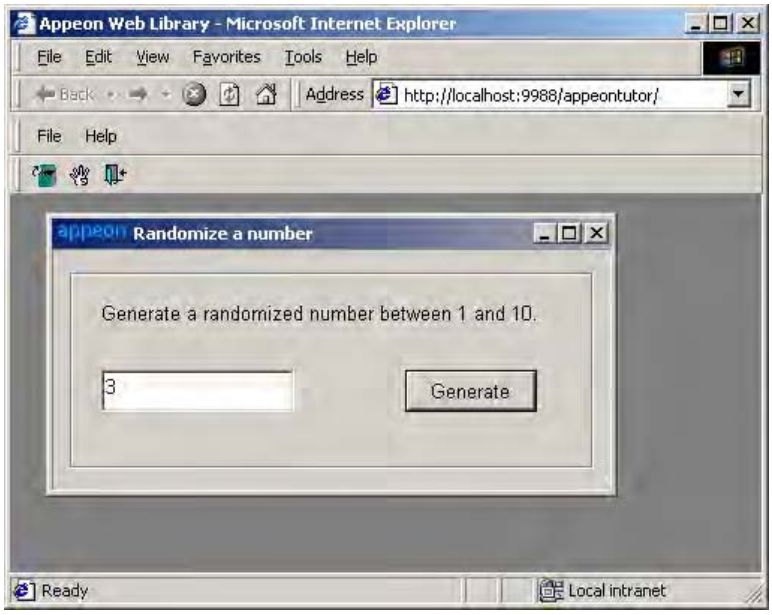
STEP 3 – Open the Drag and Drop window, as shown in Figure 8-2.

Figure 8-2: Appeon Tutorial Application



STEP 4 – Open and test the *Randomize a number* Window, as shown in Figure 8-3.

Figure 8-3: Appeon Tutorial Application



Index