# PowerServer Mobile Tutorials

Appeon® PowerServer® 2017

FOR WINDOWS & UNIX & LINUX

DOCUMENT ID: ADC50001-01-0700-01

LAST REVISED: October 27, 2017



Appeon Limited, 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 intended for PowerBuilder developers who want to install PowerServer Mobile, deploy and run the mobile application, package and distribute the application, or configure the PowerServer cluster.

## 1.2 How to use this book

There are six chapters in this book.

Chapter 1: About This Book

A general description of this book

Chapter 2: Tutorial 1: Set up the Environment

Detailed instructions for setting up the PowerServer Mobile environment.

Chapter 3: Tutorial 2: Config, Deploy & Run the Application

Detailed instructions for configuring and deploying an existing PowerBuilder application, and then running the application on mobile device.

Chapter 4: Tutorial 3: Develop & Debug with PowerServer Mobile

High-level guidelines for developing and debugging the application with PowerServer Mobile.

Chapter 5: Tutorial 4: Package & Distribute Native Mobile Apps

Detailed instructions for packaging and distributing the app as a native iOS or Android app.

Chapter 6: Tutorial 5: Configure PowerServer Cluster

Detailed instructions for configuring a PowerServer cluster.

## 1.3 Related documents

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

- Introduction to Appeon:

  Gives general introduction to Appeon PowerServer and its editions.

- Getting Started (for PowerServer Mobile):

  Guides you though installing PowerBuilder and Appeon PowerServer, and developing and deploying a mobile application.

- New Features Guide:

  Introduces new features and changes in Appeon PowerServer.

- PowerServer Mobile Tutorials:

Gives instructions on deploying, running, and debugging the mobile application, distributing native mobile apps, and configuring the PowerServer cluster.

- PowerServer Mobile (Offline) Tutorials:

  Gives instructions on setting up the PowerServer Mobile (Offline) environment, and configuring, deploying, running, updating, and debugging the offline application.

- Appeon Installation Guide:

  Provides instructions on how to install Appeon PowerServer successfully.

- Mobile UI Design & Development Guide:

  Introduces general guidelines on designing and developing the mobile app and UI.

- Migration Guidelines for PowerServer Web:

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

- Supported PB Features:

  Provides a detailed list of supported PowerBuilder features which can be converted to the Web/Mobile with Appeon as well as which features are unsupported.

- Workarounds & API Guide:

  Provides resolutions for unsupported features and various APIs to facilitate you to implement the features (including Web and mobile) that are not easy or impossible to implement in the PowerBuilder IDE.

- Appeon Workspace User Guide:

  Gives a general introduction on Appeon Workspace and provides detailed instructions on how to use it.

- PowerServer Configuration Guide:

  Provides instructions on how to configure PowerServer Monitor, establish connections between PowerServer and database servers, and configure AEM for maintaining PowerServer and the deployed applications.

- Web Server Configuration Guide:

  Describes configuration instructions for different types of Web servers to work with the PowerServer.

- Troubleshooting Guide:

  Provides information on troubleshooting issues; covering topics, such as product installation, application deployment, AEM, and Appeon application runtime issues.

- Appeon Performance Tuning Guide:

Provides instructions on how to modify a PowerBuilder application to achieve better performance from its corresponding Web/mobile application.

- Testing Appeon Web Applications with UFT:

  Provides instructions on how to test Appeon Web applications with UFT.

## 1.4 If you need help

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

# 2 Tutorial 1: Set up the Environment

You will need to go through the following tasks to set up the PowerServer Mobile 2017 environment for deploying and running the mobile application:

## 2.1 Task 1: Prepare the machine

The simplest scenario will be used in this guide, which requires 1 Windows PC and 1 Android or iOS device.

- **1 Windows PC**: used as the development machine and the server
- **1 Android or iOS device**: used as the mobile client

The 32-bit version of **PowerServer Mobile for .NET** edition will be used to walk you through this guide, so please prepare the environment according to the following requirements.

Software requirements for **Windows PC**:

- Windows 7 or 8.1 (32-bit or 64-bit)

  **PowerServer Mobile 32-bit** version can be installed to the 64-bit OS without any special considerations. But if you have installed **Appeon PowerServer 64-bit** version before please uninstall it first. See Task 1.2: Uninstall previous version of 64-bit Appeon.

- Internet Explorer 9, 10, or 11
- (Optional) SAP SQL Anywhere (32-bit engine) 11.0, 12.0, 16.0, or 17 (for running the demos included with the Appeon installation)
- .NET Framework 4.x

  For Windows 7, you will need to first download the .NET Framework 4.0 setup program from http://www.microsoft.com/en-us/download/details.aspx?id=17718.

- IIS 7.x or 8.x

  See Task 1.3: Install IIS and Task 1.4: Configure IIS.

- PowerBuilder 2017 & PowerServer Mobile 2017

  You can download **PowerBuilder 2017 Universal Edition** from the Appeon web site. The universal edition includes the installation of PowerBuilder, PowerServer Mobile, and PowerServer Toolkit. Or you can download and install PowerBuilder 2017 Standalone Edition first and then download and install PowerServer Mobile 2017.

  See Task 2: Install PowerBuilder and PowerServer for installation instructions.

Software requirements for the Android or iOS device:

- Google Android 4.2, 4.3, 4.4, 5.0, 6.x, or 7.0, all 32-bit, or

Apple iOS 9 or 10

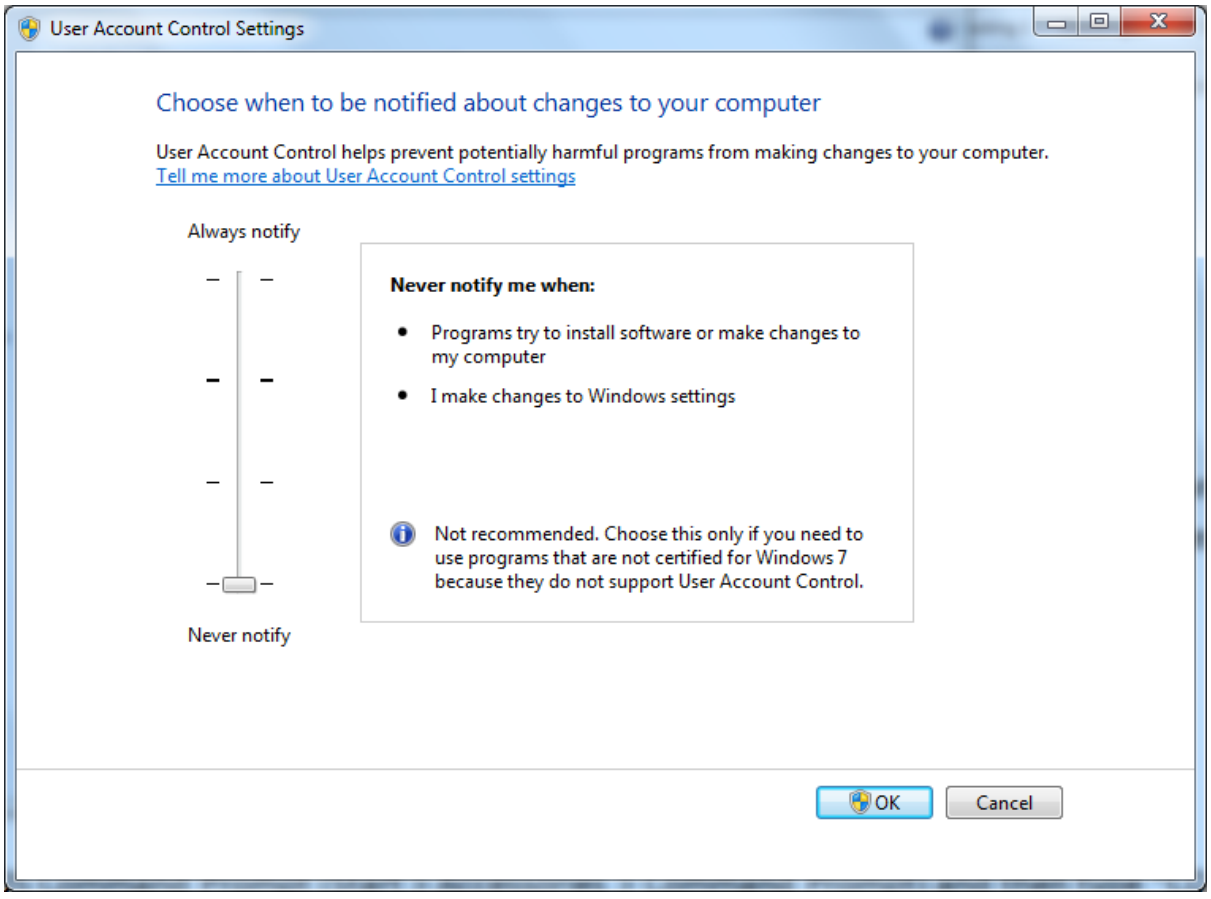
- Appeon Workspace

  See Task 3: Install Appeon Workspace (on the Android or iOS device)

### 2.1.1 Task 1.1: Disable UAC (User Account Control)

On the Windows PC, go to the **Control Panel** > **User Accounts** and disable UAC by setting the slider to "**Never notify**" (see screenshot below). After you have disabled UAC you **MUST** restart the computer. If you do not restart the computer the change will NOT take effect.

**Figure 2.1: Disable UAC**

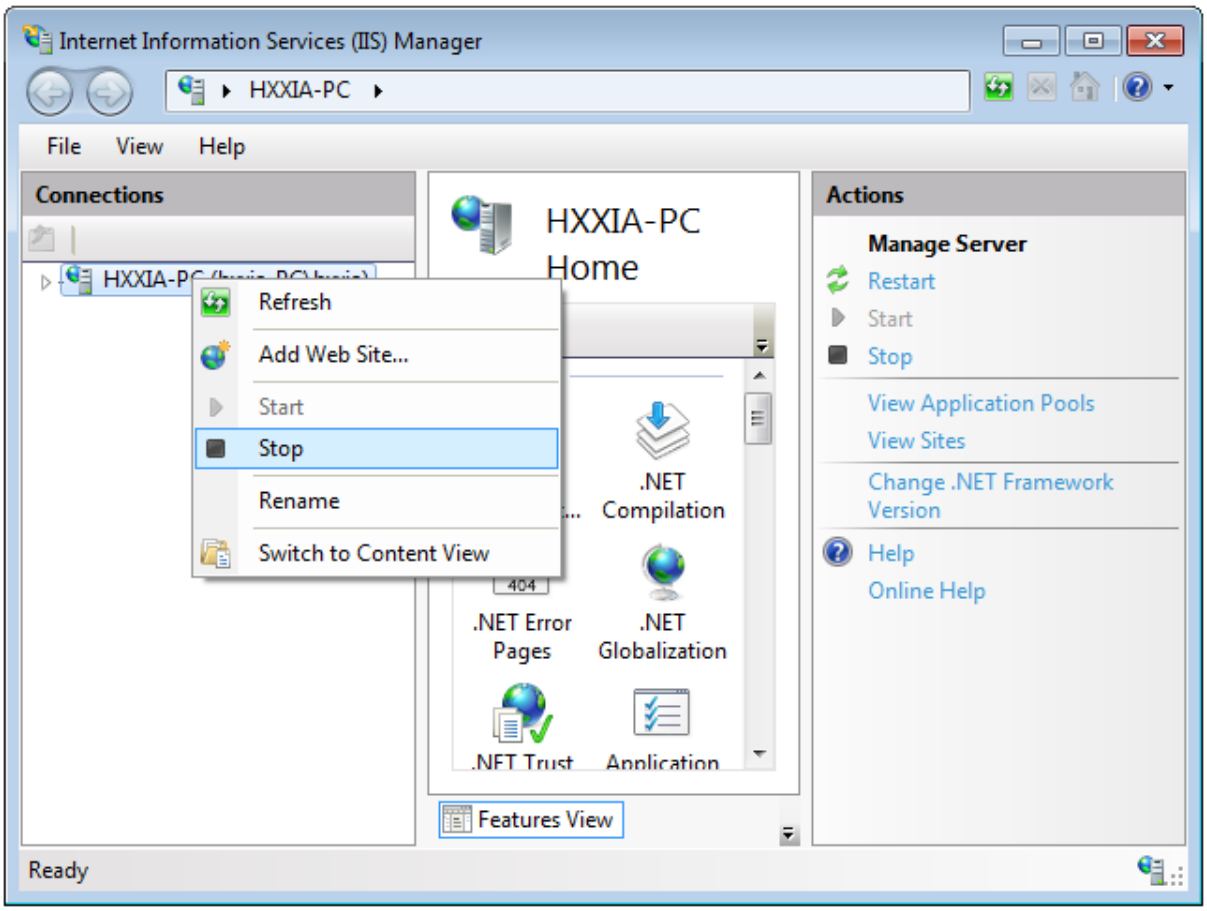



### 2.1.2 Task 1.2: Uninstall previous version of 64-bit PowerServer

If you are using a 64-bit machine you can install **PowerServer Mobile (32-bit)** to the 64-bit OS without any special configurations. But if you have a previous version of **Appeon PowerServer (64-bit)** already installed on this machine then you must uninstall it first.

Step 1: Open **IIS Manager**, right click the **top** node (not the website node) in the treeview and select **Stop** from the popup menu. This will stop the entire IIS.

**Figure 2.2: Stop the entire IIS Manager**




Step 2: Close any opened window, especially **PowerBuilder** and **IIS Manager**.

Step 3: Uninstall all of the Appeon components including **PowerServer**, **PowerServer Toolkit**, and **PowerServer Help**. You will need to uninstall these components one by one.

Step 4: Verify Appeon is cleanly uninstalled by the following two steps:

- Double check the **Control Panel\Programs\Programs and Features** and make sure no Appeon component is listed.
- Open a command prompt window and then type **regedit**<Enter>. Double check that no **ADT** or **ASN** keys are listed under HKEY_LOCAL_MACHINE\SOFTWARE\Appeon \<version_number>.

Step 5: Clear the Internet Explorer cache and temporary files.

Step 6: Delete the entire Appeon folder from C:\Users\User_Name\AppData\Roaming\.

Step 7: Delete all the Appeon application folders from the IIS Web root. For example, under *C:\inetpub\wwwroot* at minimum you should delete the following folders: appeon, appeon_acf_demo, appeon_code_examples, pet_world, sales_application_demo.

Step 8: Restart the machine.

Step 9: Start IIS by right-clicking the **top** node in the treeview in the IIS Manager and selecting **Start** from the popup menu.

After that, you can proceed to install **PowerServer Mobile** by following steps in Task 2: Install PowerServer & PowerServer Toolkit.

### 2.1.3 Task 1.3: Install IIS

IIS is not installed on Windows 7 or 8.1 by default. You need to manually install it.

Before you install IIS, make sure you have installed .NET Framework 4.x. To install .NET Framework 4.x for Windows 8.1, please go to the **Turn Windows features on or off** and check the box for .NET 4.5; for Windows 7, please download the setup program from http://www.microsoft.com/en-us/download/details.aspx?id=17718 and install .NET Framework 4.0.

Below are steps for installing IIS. If you already have IIS installed, please carefully verify the correct IIS options are checked (see screenshots below). By default, not all the required components will be installed.

Step 1: Click **Start** and then click **Control Panel**. In Control Panel, click **Programs** and then click **Turn on or off Windows features**.

Step 2: Select the check box of **Internet Information Services**, then expand the list and select the items under **Web Management Tools, Application Development Features** and **Common HTTP Features** according to the figure below. Click **OK** to let Windows finish the install.

**Figure 2.3: Select the Web Management Tools**

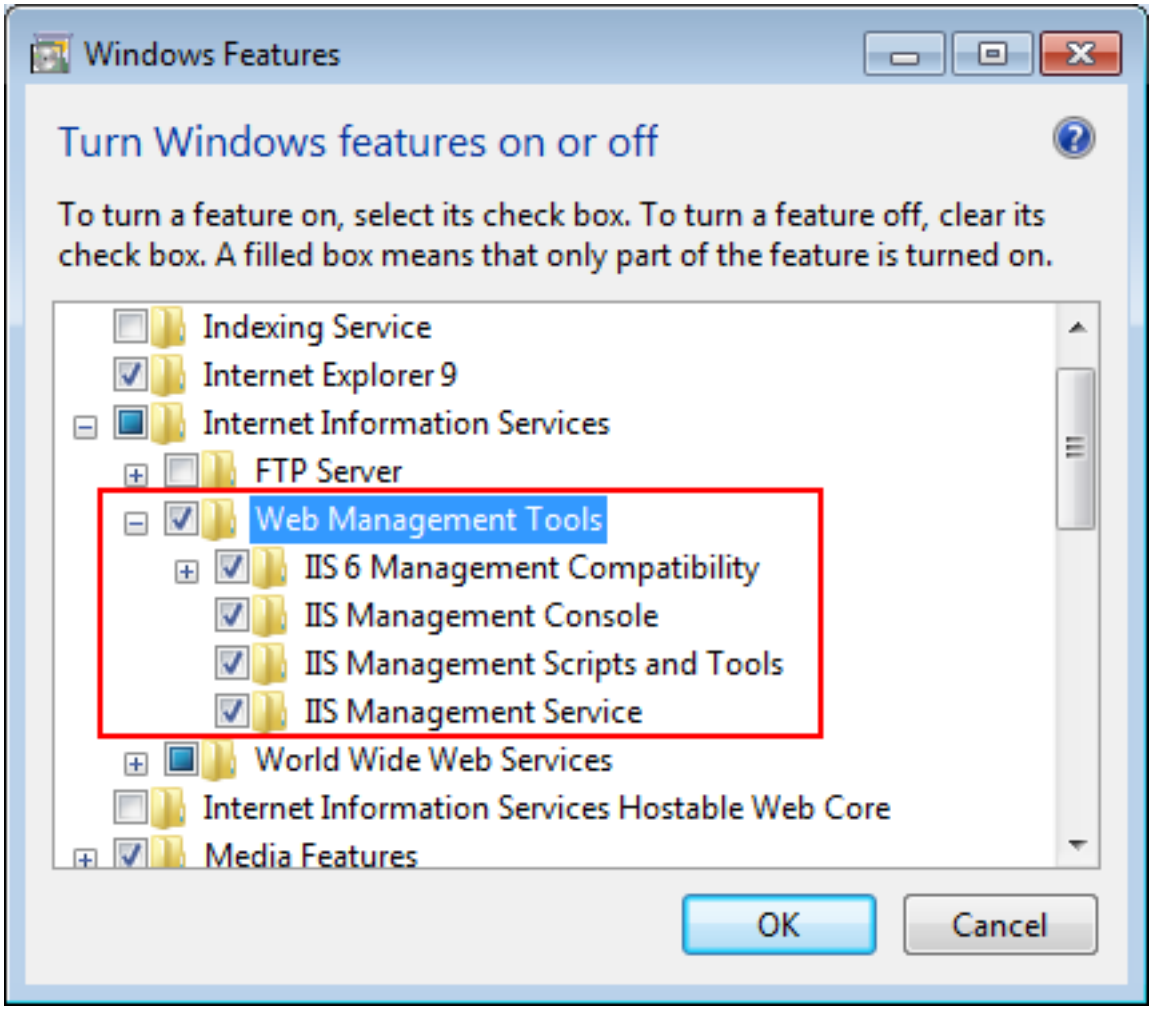


**Figure 2.4: Select the Application Development Features**

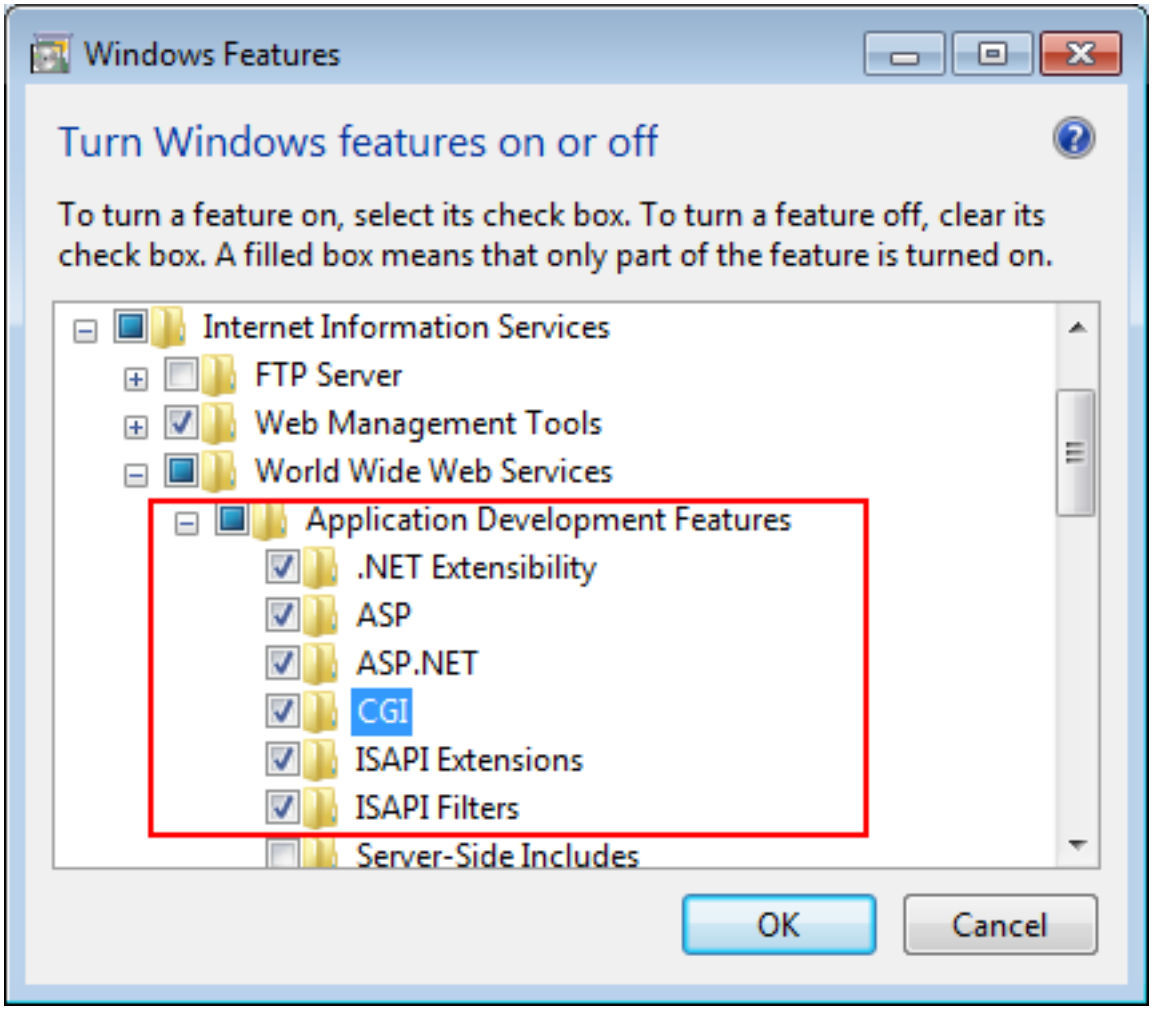
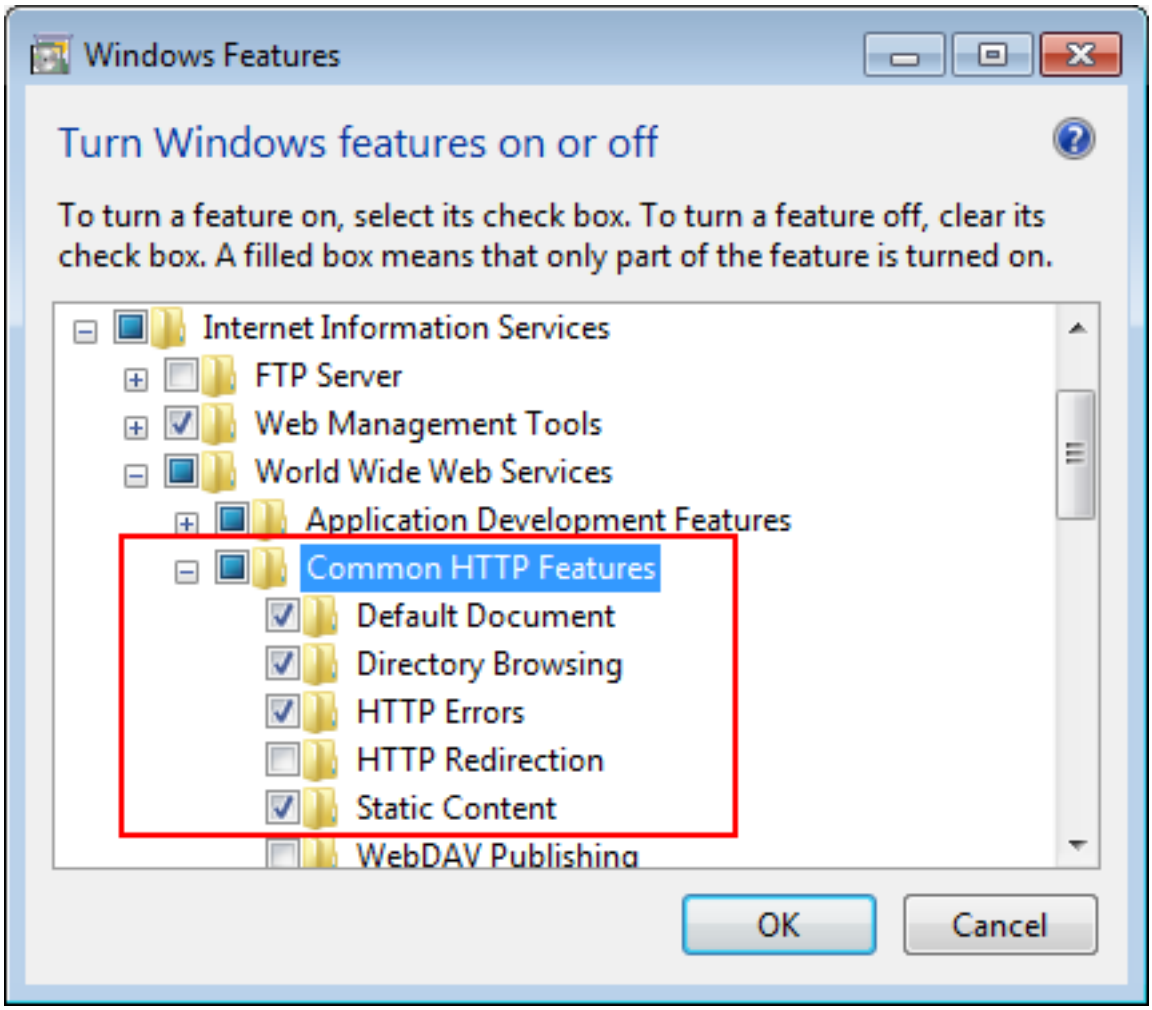


**Figure 2.5: Select the Common HTTP Features**




Step 3: After IIS is installed, go to **IIS Manager**, right click the **Default Web Site**, select **Binding**, and make sure **IP Address** is indicated with an asterisk "*". If not, please click **Edit** and select **All Unassigned** for the IP address, this will display **IP Address** as an asterisk "*".

Step 4: Run http://*IP_Address*:80/ in Internet Explorer. If the IIS welcome screen displays, then IIS is working properly.

**Tip**: to obtain the IP address of the server, open a command prompt window and then type **ipconfig**<Enter>. Remember this IP address as it is also needed when you configure the mobile app in Appeon Workspace.

If IIS is not working, please re-install IIS or fix the IIS configuration by following the IIS help.

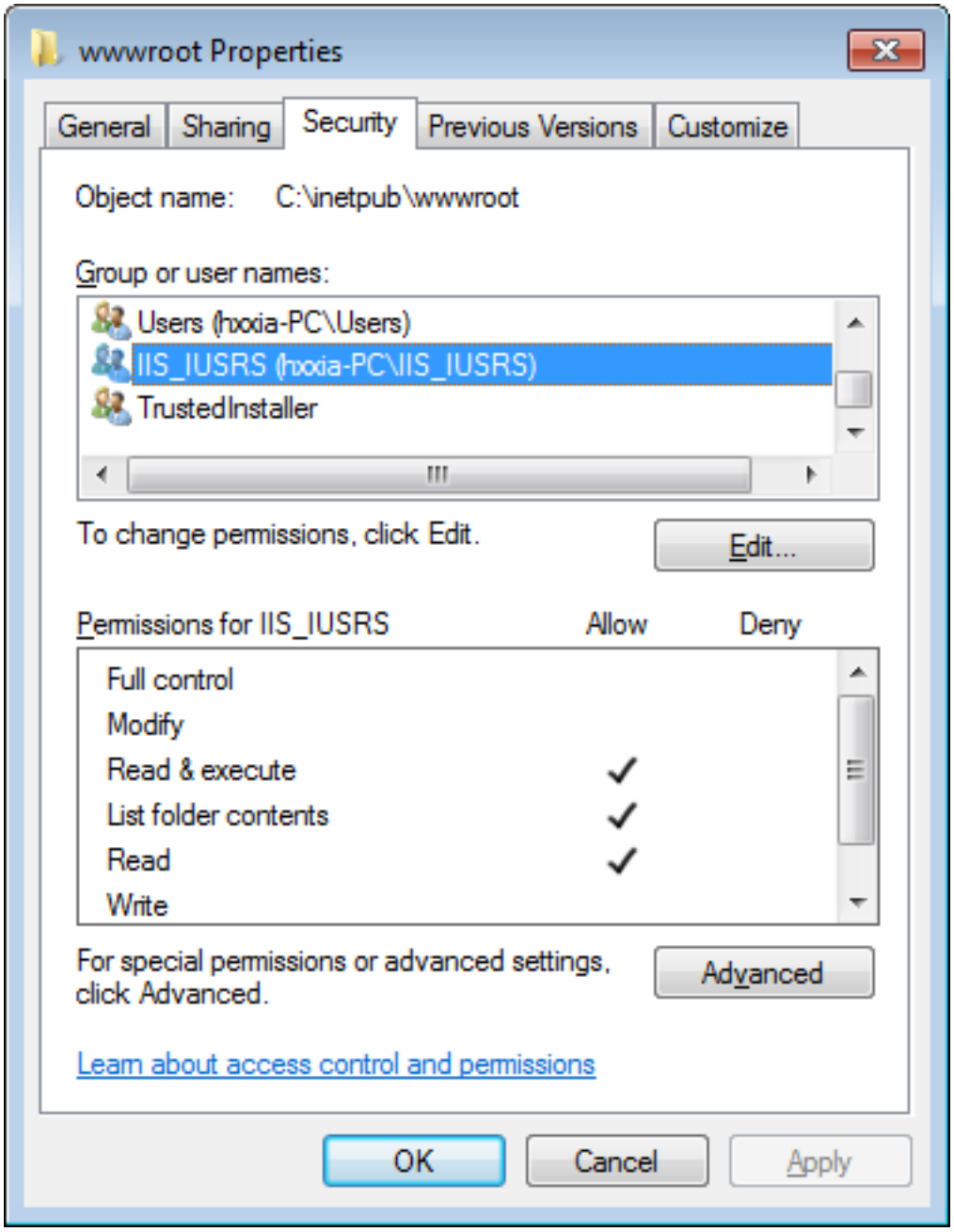
### 2.1.4 Task 1.4: Configure IIS

Follow steps below to grant **IIS_IUSRS** account with full controls to the IIS Web root folder:

Step 1: Right click the **C:\Inetpub\wwwroot** folder and select **Properties** from the popup menu.

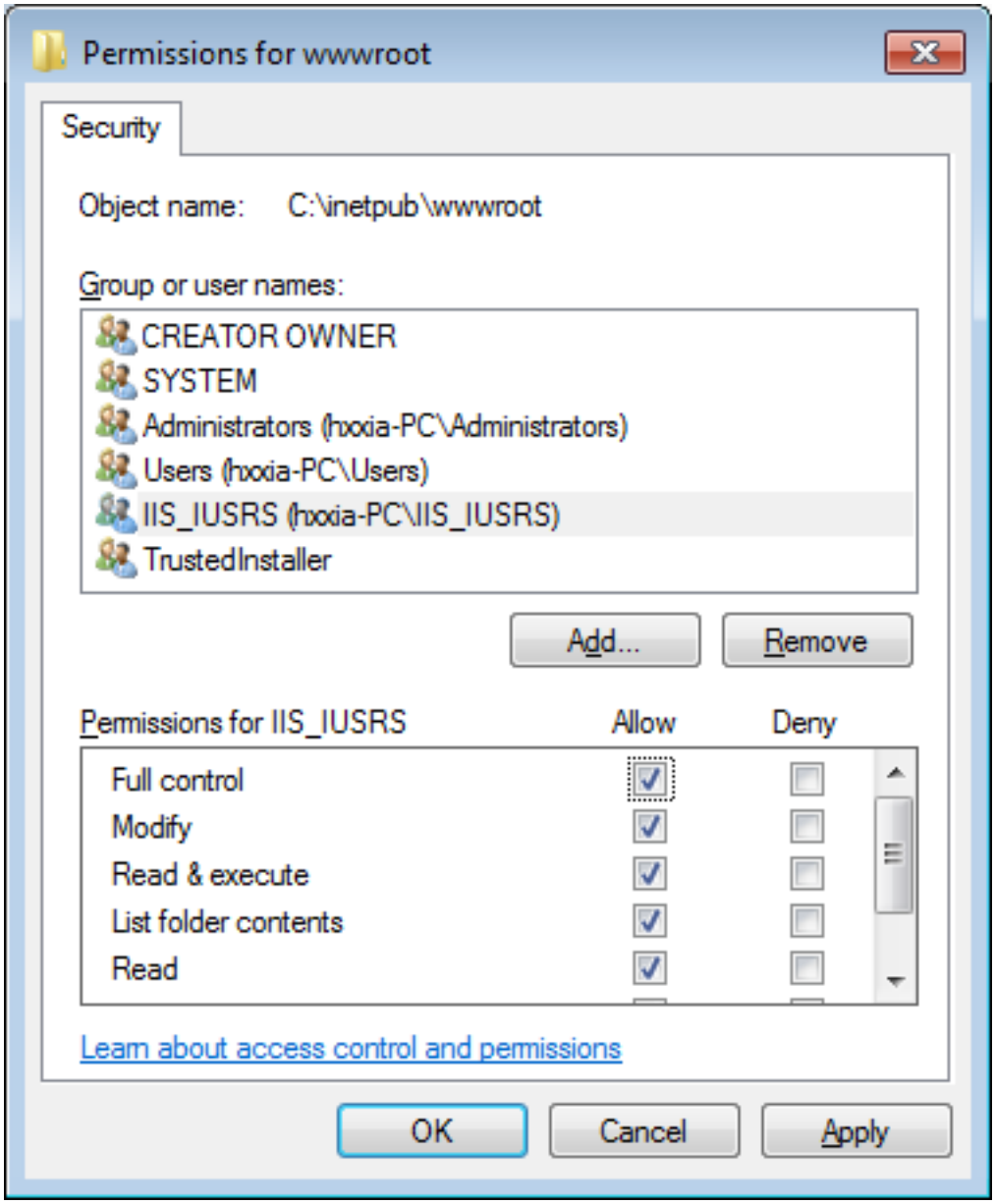
Step 2: On the **Security** tab, select **IIS_IUSRS** in the **Group or user names** list box, and then click the **Edit** button.

**Figure 2.6: Select IIS_IUSRS**




Step 3: Select **IIS_IUSRS** in the **Group or user names** list box, and then select the **Allow** check box for the **Full control**.

**Figure 2.7: Select Full control for IIS_IUSRS**




Step 4: Click **OK** to save the settings.

## 2.2 Task 2: Install PowerBuilder and PowerServer

### 2.2.1 Task 2.1: Install PowerServer Mobile and PowerServer Toolkit

**Prerequisite task**: if PowerBuilder has not been installed yet, you can follow the installation guide online help to install PowerBuilder first.

On the Windows PC where PowerBuilder is already installed, install PowerServer Mobile and PowerServer Toolkit by the following steps.

Step 1: Start IIS server: open IIS Manager, right click the **top** node (not the website node) in the treeview and select **Start** from the popup menu.

Step 2: Launch the PowerServer product setup program by double-clicking **setup.exe**. Click **Next** until you reach the screen for selecting the product components. Make sure the checkboxes for **PowerServer Mobile for .NET** and **PowerServer Toolkit** are selected.

Step 3: Click **Next** until you reach the screen for specifying the IIS Web site where PowerServer Mobile for .NET will be installed. Make sure **Select an existing Web Site** and **Default Web Site** are selected.

Step 4: Click **Next**. Choose whether to install the **Demo Applications**. If you have SQL Anywhere database server installed, select to install the demo; otherwise, do not install the demo since the demo requires SQL Anywhere database server to be previously installed.

Step 5: If you selected to install the **Demo Applications**, specify the path for the SQL Anywhere database server engine.

Step 6: Click **Next** until Setup begins copying files for the PowerServer Mobile for .NET installation.

When PowerServer Mobile for .NET installation is completed, PowerServer Toolkit installation automatically starts.

Step 7: Click **Next** until Setup begins copying files for the PowerServer Toolkit installation.

Step 8: When the installation is completed, click **Finish**.

### 2.2.2 Task 2.2: Verify the installation

On the Windows PC, run http://*IP_Address*:80/AEM/. If AEM is launched successfully, then PowerServer is installed successfully.

## 2.3 Task 3: Install Appeon Workspace (on the Android or iOS device)

Instructions for installing Appeon Workspace on the Android device is different from that on the iOS device.

**Instructions for iOS:**

Step 1: Make sure your iOS device can connect to Internet.

Step 2: On your iOS device, open **App Store**, search for "Appeon Workspace", and then tap **Install**.

Alternatively, you can download Appeon Workspace from the Apple App Store to your Windows PC via a Web browser, and then synchronize it to your iOS device via iTunes.

**Instructions for Android:**

Step 1: Enable the **Unknown resources** option (in **Settings** > **Security**) on the Android device so you can install apps that are not downloaded from Google Play.

Step 2: Make sure your Android device can connect to PowerServer.

Step 3: Visit the Appeon Workspace download center that is posted on PowerServer (http://*Server_IP*:80/AWS), and then click the download button.

## 2.4 Task 4: Configure the network connection

Check and make sure the Windows PC and the mobile device connected to the same Wi-Fi router. To verify this, on the mobile device, open the Web browser and type http://*Server_IP*:80/AEM/. If AEM is launched successfully, it means that the PowerServer is properly installed and that the mobile device is able to connect to the Windows PC.

**NOTE**: Mobile Internet is supported by PowerServer Mobile. However, in order for the Appeon Workspace to connect to your PowerServer you will need an external IP address. You can verify that your external IP address is properly working by typing *http://External_IP_Address:80* into a Web browser of any device connected to the Internet. If you get a page not found error or other HTTP error then your network is not configured properly for external access.

# 3 Tutorial 2: Config, Deploy & Run the Application

This tutorial will use the PowerServer Mobile for .NET edition to walk you through configuring, deploying and running the mobile application.

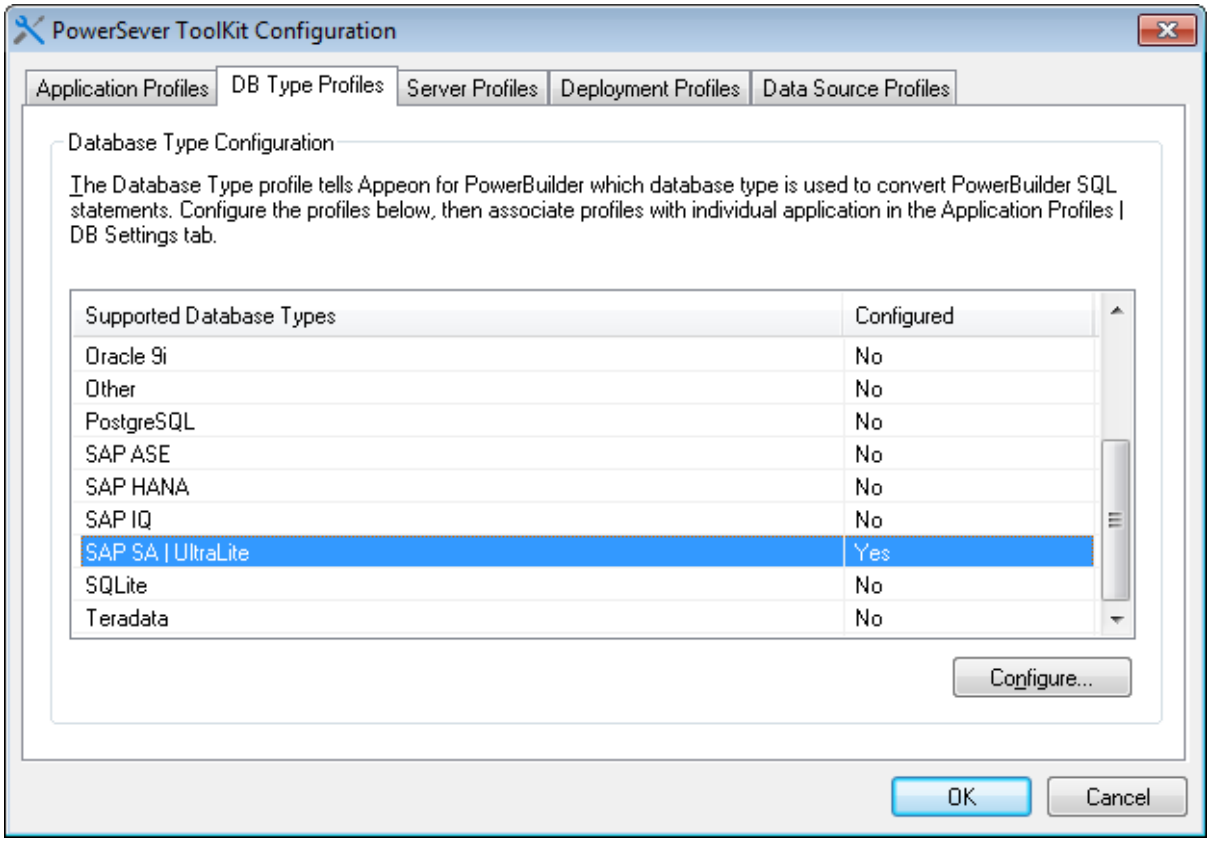
## 3.1 Task 1: Configure the database type

Step 1: On the PowerServer Toolkit, click the **Configure** icon ( ).

Step 2: On the **PowerServer Toolkit Configuration** window, click the **DB Type Profile** tab.

Step 3: View the **Configured** column of the database type used by your application, if it displays **Yes**, you can skip the remaining steps in this task. Otherwise, select the database type used by your application, and then click **Configure**.

**Figure 3.1: DB Type Profile in ADT**




Step 4: On the **Database Type Profile Configuration** window, do the following (take **SAP SA | UltraLite** as an example):

1. Select the **ODBC Interface** radio button from the **Database Interface** option.

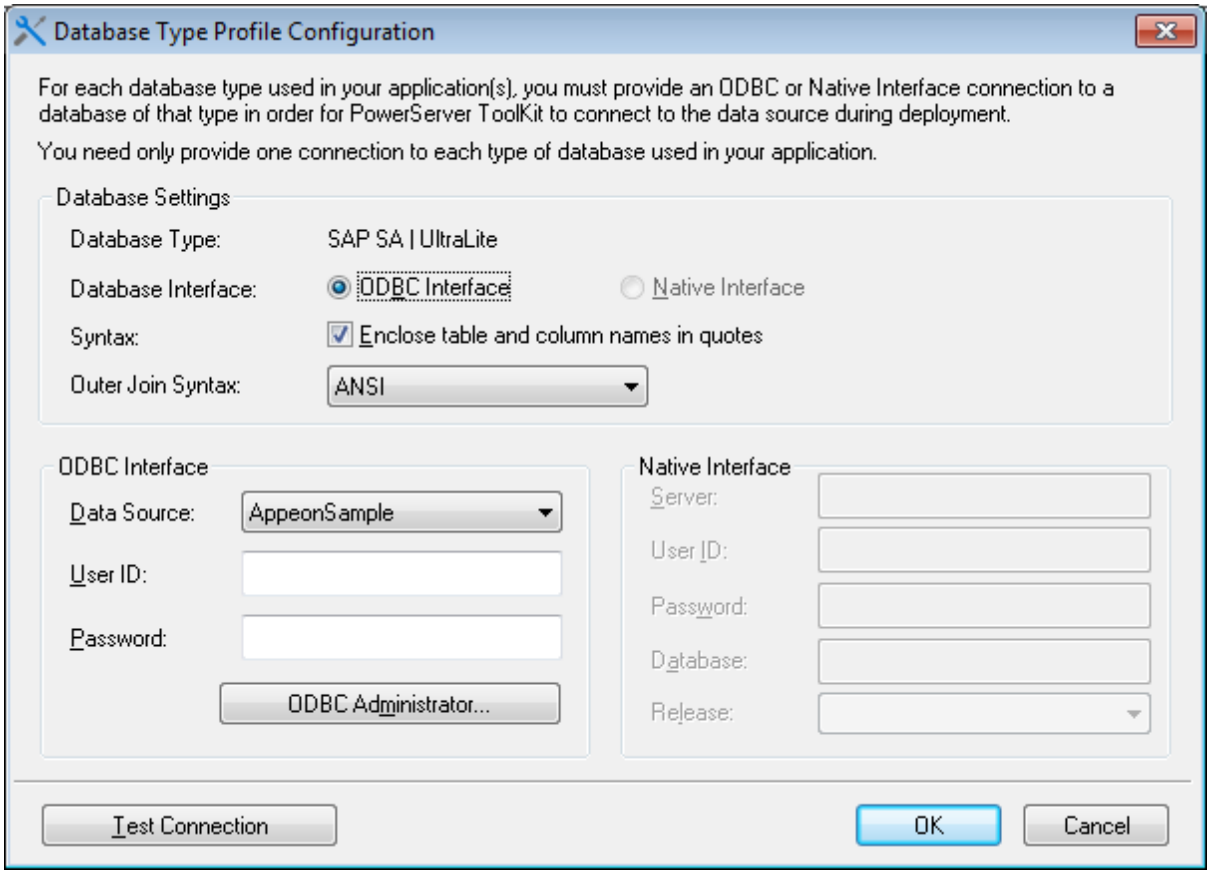
2. Select a data source from the **Data Source** dropdown list box. It can be any data source (ODBC DSN) provided it is the same type as used by your application. It does NOT have to be the actual ODBC DSN used by your particular application. The database type is what is important and that a connection can be established to the specified ODBC DSN.

3. Keep the rest as default, and then click **Test Connection**.

4. Make sure that the database connection is successful.

5. Click **OK**.

**Figure 3.2: Database Type Profile Configuration**




Now the **Configured** column of your database type will be indicated by **Yes**. You can select the database type when you create the application profile for your application in Task 3: Configure and deploy the application.

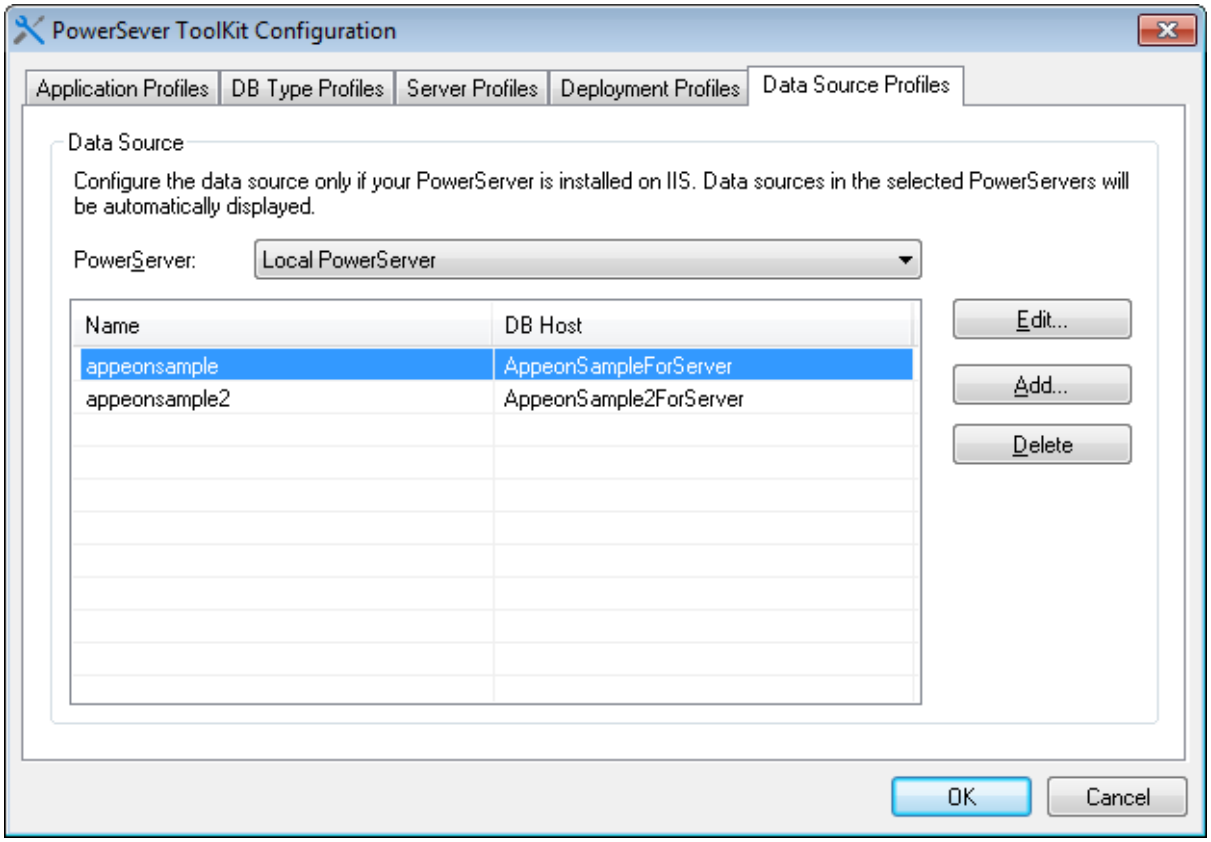
## 3.2 Task 2: Configure the database connection

Step 1: In the **PowerServer Toolkit Configuration** window, click the **Data Source Profile** tab.

Step 2: Select **Local PowerServer** from the **PowerServer** dropdown list box.

Step 3: Click the **Add** button to create a data source in the selected PowerServer.

**Figure 3.3: Connection Cache Profiles**




Step 4: On the **Add Data Source** window, specify the data source settings. Below we take **SAP SQL Anywhere** and **Microsoft SQL Server** database as examples.

Steps for configuring a data source for **SAP SQL Anywhere** database:

1. In the **Name** text box, input any text you like as the name of the data source.

2. Select **ODBC Driver** from the **Driver** dropdown listbox.

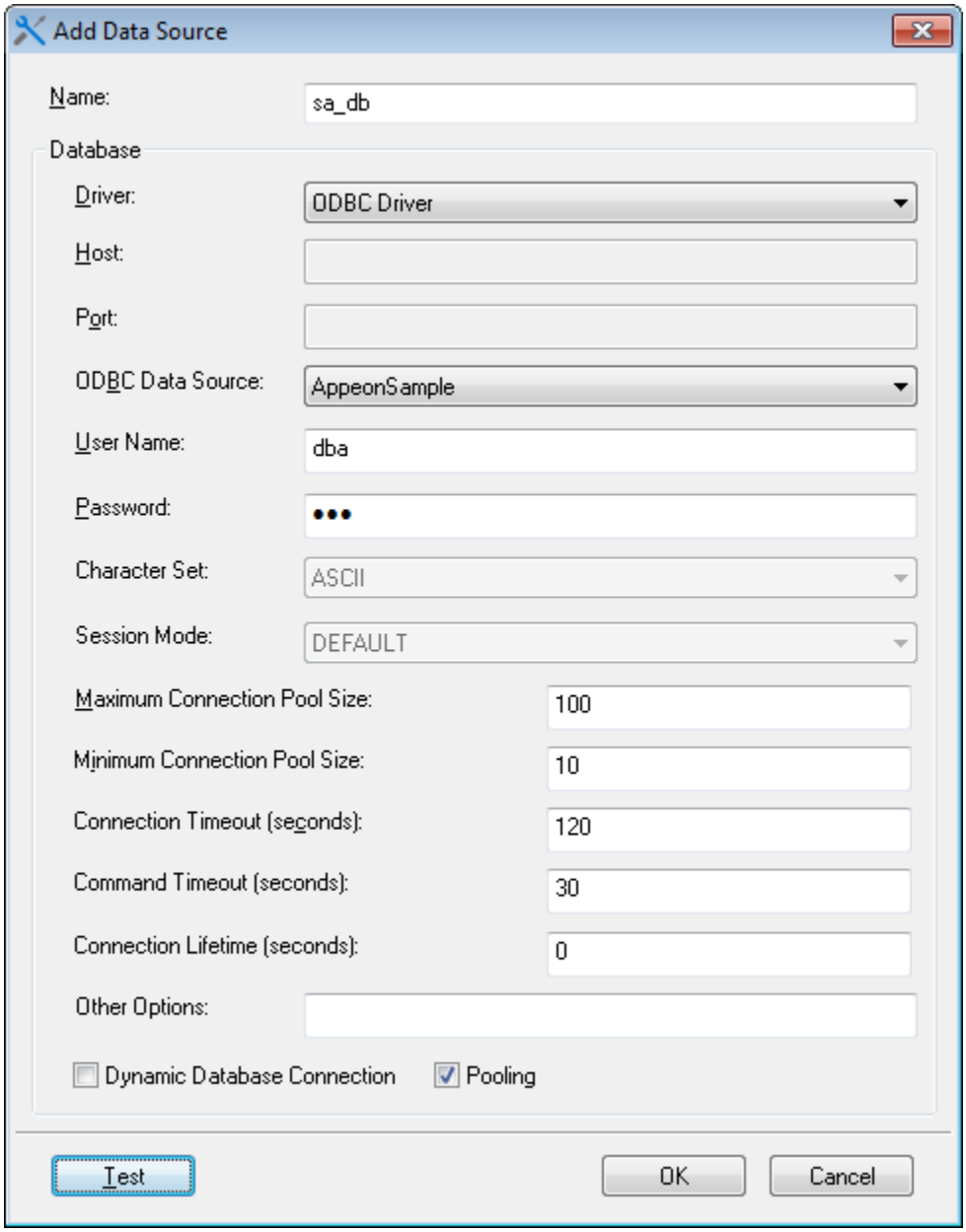
3. Select the data source from the **ODBC Data Source** dropdown listbox.

4. Input the database login user name and password respectively in the **User Name** text box and the **Password** text box.

5. Keep the rest as default, and then click **Test**. Make sure the test is successful.

6. Click **OK**.

**Figure 3.4: SAP SQL Anywhere data source**




Steps for configuring a data source for **Microsoft SQL Server** database:

1. In the **Name** text box, input any text you like as the name of the data source.

2. Select **MS SQL Server Native Driver** from the **Driver** dropdown listbox.

3. Input the IP address or machine name of the database server in the **Host** field.

4. Input the port number of the database server in the **Port** field.

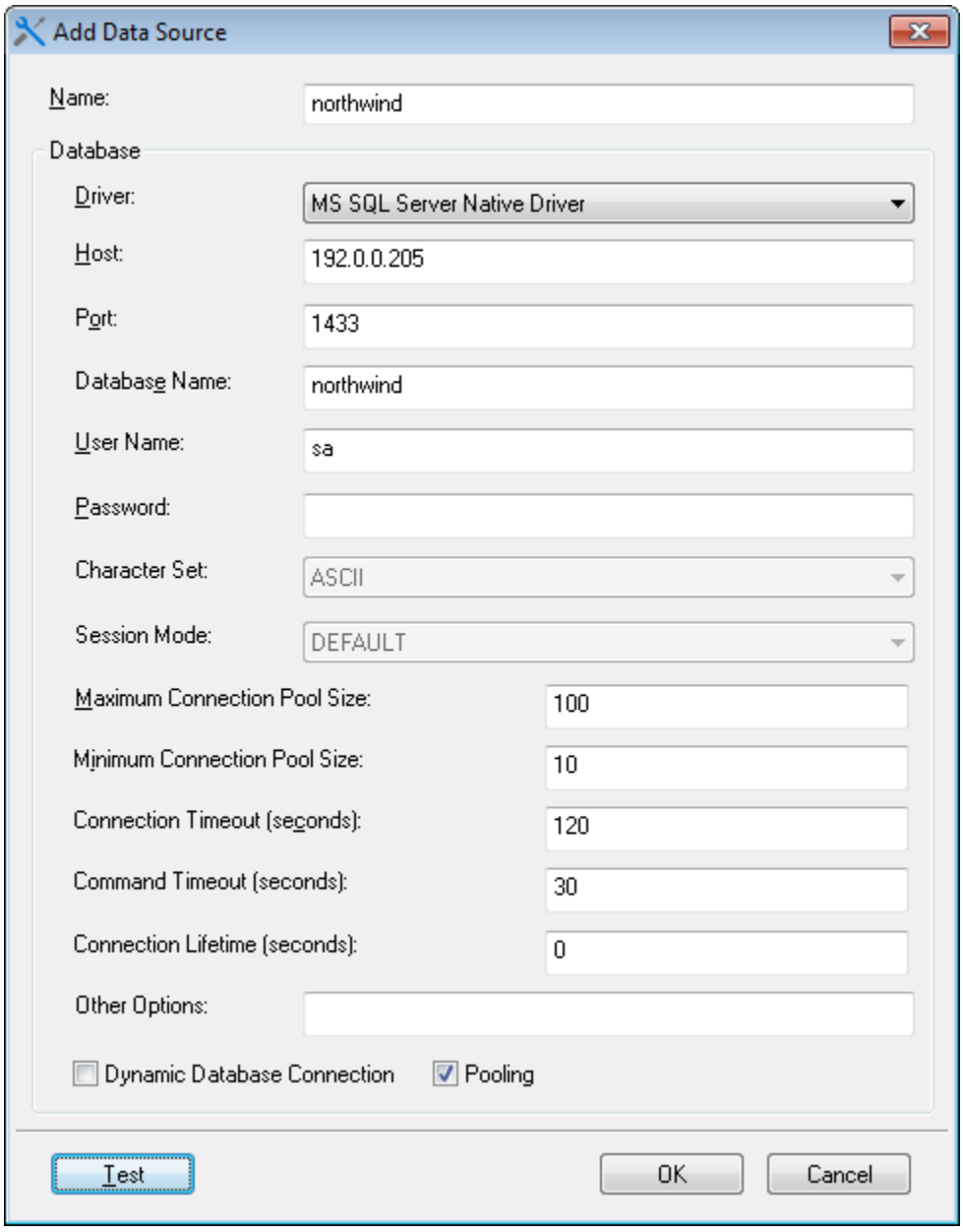
5. Input the database name, database login user name and password respectively in the **Database Name**, the **User Name** and the **Password** text boxes.

6. Keep the rest as default, and then click **Test**. Make sure the test is successful.

7. Click **OK**.

**Tip**: you should check with your SQL Server administrator to verify the port number, the user name, and the password are correct.

**Figure 3.5: MS SQL Server data source**




Now the data source you added will be listed in the **Data Source Profiles** tab. You can select it when you create the application profile for your application in Task 3: Configure and deploy the application.

## 3.3 Task 3: Configure and deploy the application

Step 1: On the PowerServer Toolkit click the first icon **Config Wizard** ( ).

Step 2: In the welcome screen click **Next**.

Step 3: Specify following settings and then click **Next**.

1. Specify the application profile name in the **Application Profile Name** text box. The same text will be displayed in the **Web Folder** text box and the **Mobile App Name** text box. You can modify these fields to use different text.

2. Select **Mobile** from the **Project Type** dropdown list box.

3. Select a device type from the **Device Type** dropdown list box.

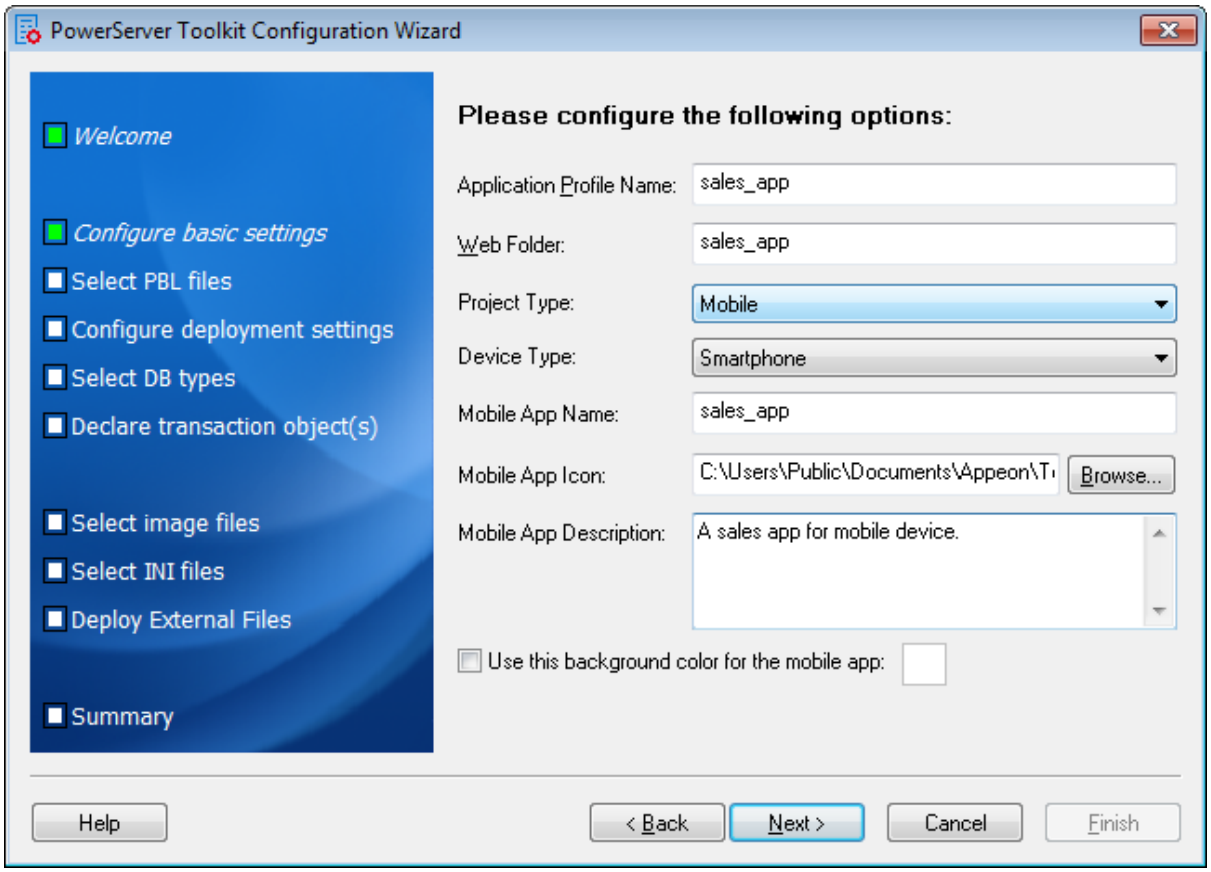
For applications that are designed for tablets, select **Tablet**; for applications that are designed for smartphones, select **Smartphone**; for applications that are designed for both devices, select **Both**.

4. In the **Mobile App Icon** text box, specify an icon for your application by clicking **Browse** to select the image file of the PNG or JPG format. The image that you choose will be displayed as the application icon in the Appeon Workspace. Recommended size is 86 x 86 pixels (or above) for low-resolution display and 172 x 172 pixels (or above) for high-resolution display (such as retina display). To ensure the best display quality in both high-resolution and low-resolution screens, you must supply an image of 172 x 172 or above. The image will be automatically shrunk to fit properly. If you leave this field empty, the default icon will be displayed in Appeon Workspace.

5. Enter a brief description in the **Mobile App Description** text box. This description will be displayed in the Appeon Workspace next to the application icon. If you leave this field empty, no app description will be displayed in the Appeon Workspace.

6. Select a background color for the app if necessary.

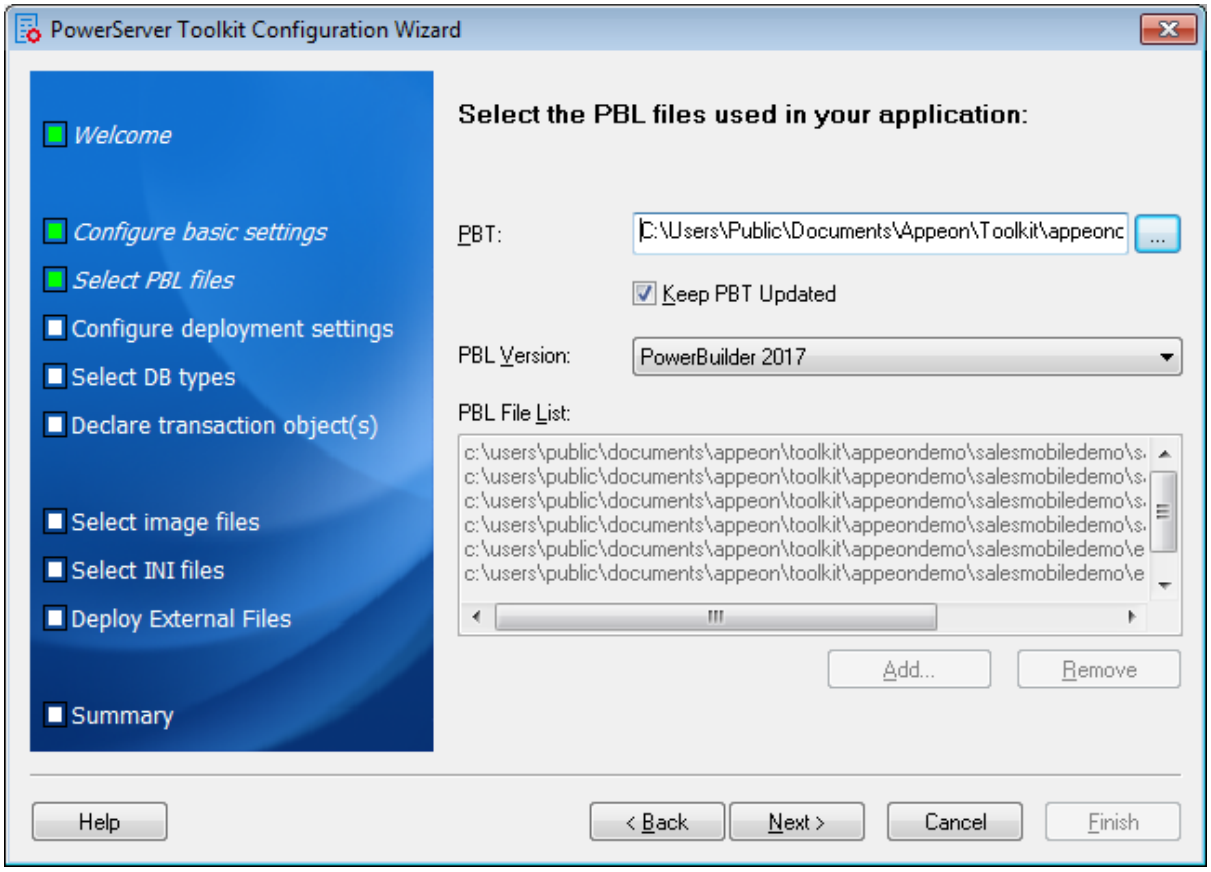
**Figure 3.6: PowerServer Toolkit Configuration Wizard**




Step 4: Add the PBT from the **PBT** dropdown list box (and all related PBLs will be automatically added for you) or add the PBL files individually that are used in your

application by clicking the **Add File** button in the **PBL File List** group box, and then select a PBL version from the **PBL Version** dropdown list box. Click **Next**.

**Figure 3.7: Add the PBT in ADT Wizard**

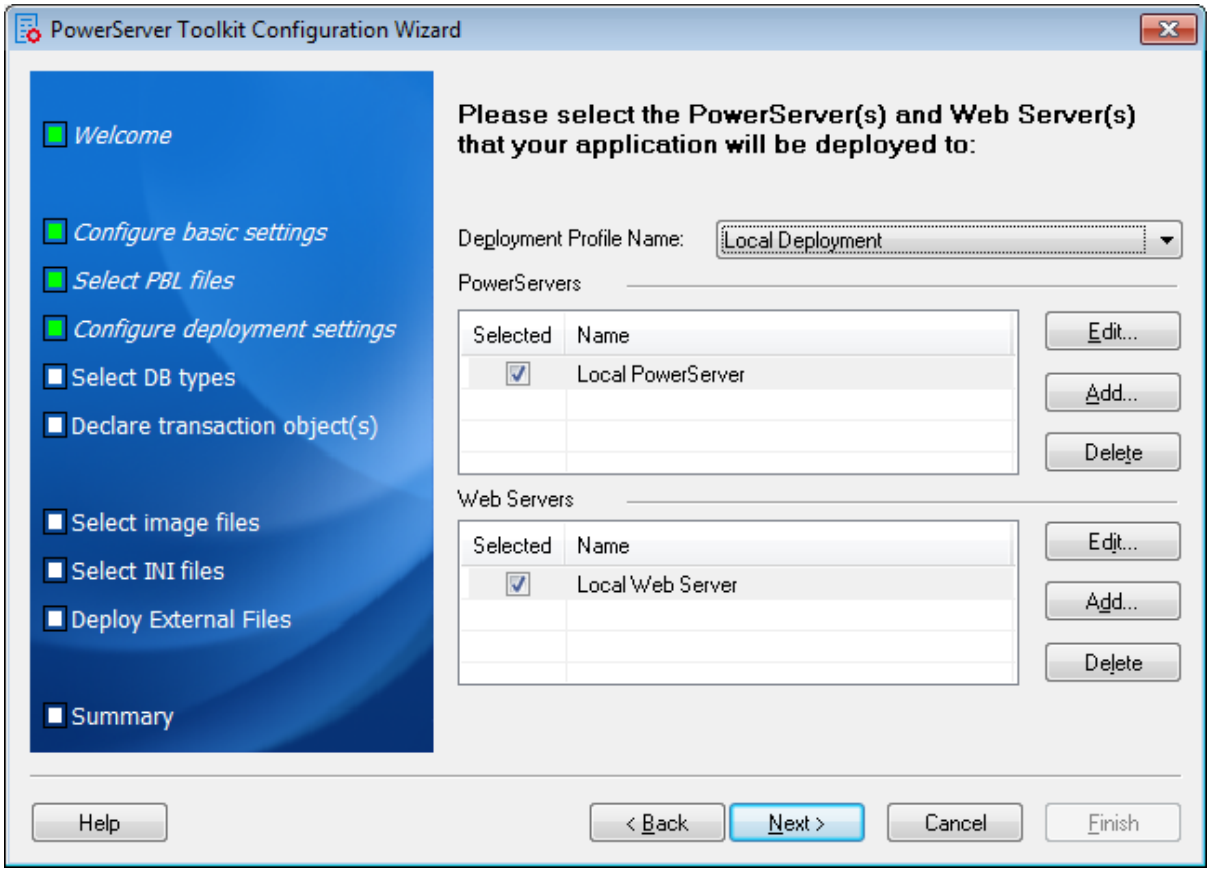



Step 5: Keep the default settings and click **Next**.

Since you have installed PowerServer Toolkit and PowerServer on the same machine, you can directly use the **Local PowerServer** profile, the **Local Web Server** profile, and the **Local Deployment** profile, all of which are configured automatically by the product setup program.

When you click **Next** the config wizard will automatically test the connection to the server. You will not be able to proceed if the connection test failed.

**Figure 3.8: Select the PowerServer(s) and Web Server(s)**

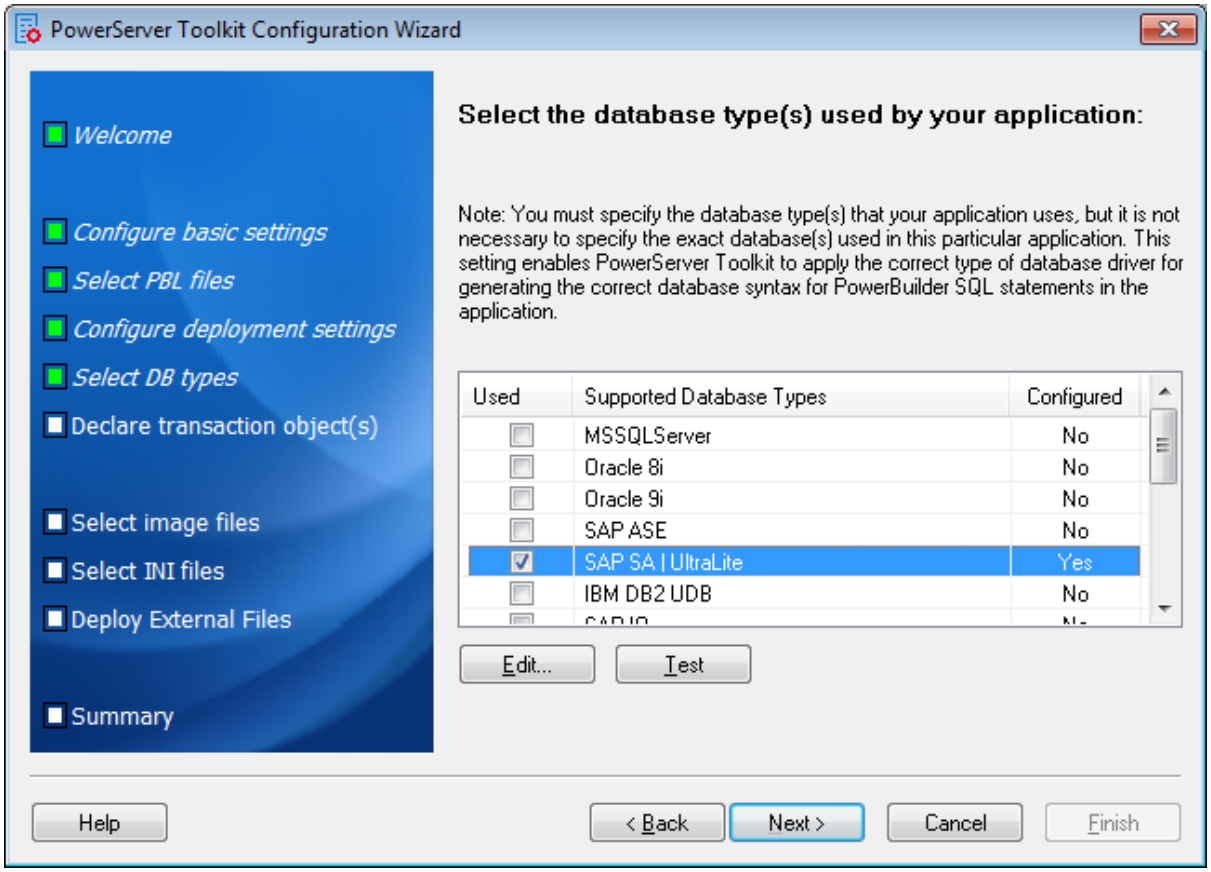



Step 6: Select the database type used by your application and click **Next**.

When you click **Next** the config wizard will automatically test the connection with the database engine. You will not be able to proceed if the connection test failed.

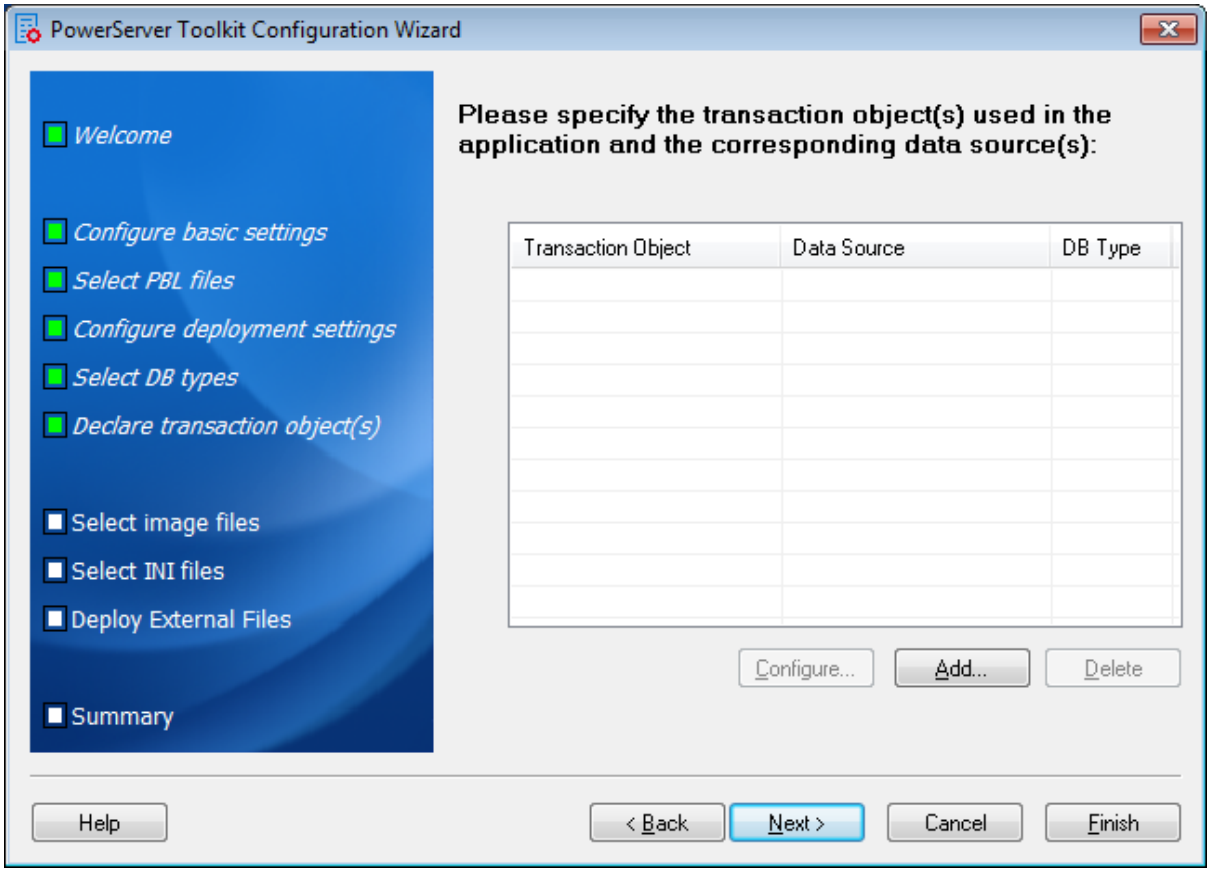
If the required database type has not been configured in Task 1: Configure the database type, you should highlight the database type, then click **Edit**, and then follow Step 4 in Task 1: Configure the database type to configure it.

**Figure 3.9: Select the database type(s)**




Step 7: Click **Add** to specify the transaction object(s) used in your application.

**Figure 3.10: Specify the transaction object**




In the **Add Transaction Object** window, specify the following settings:

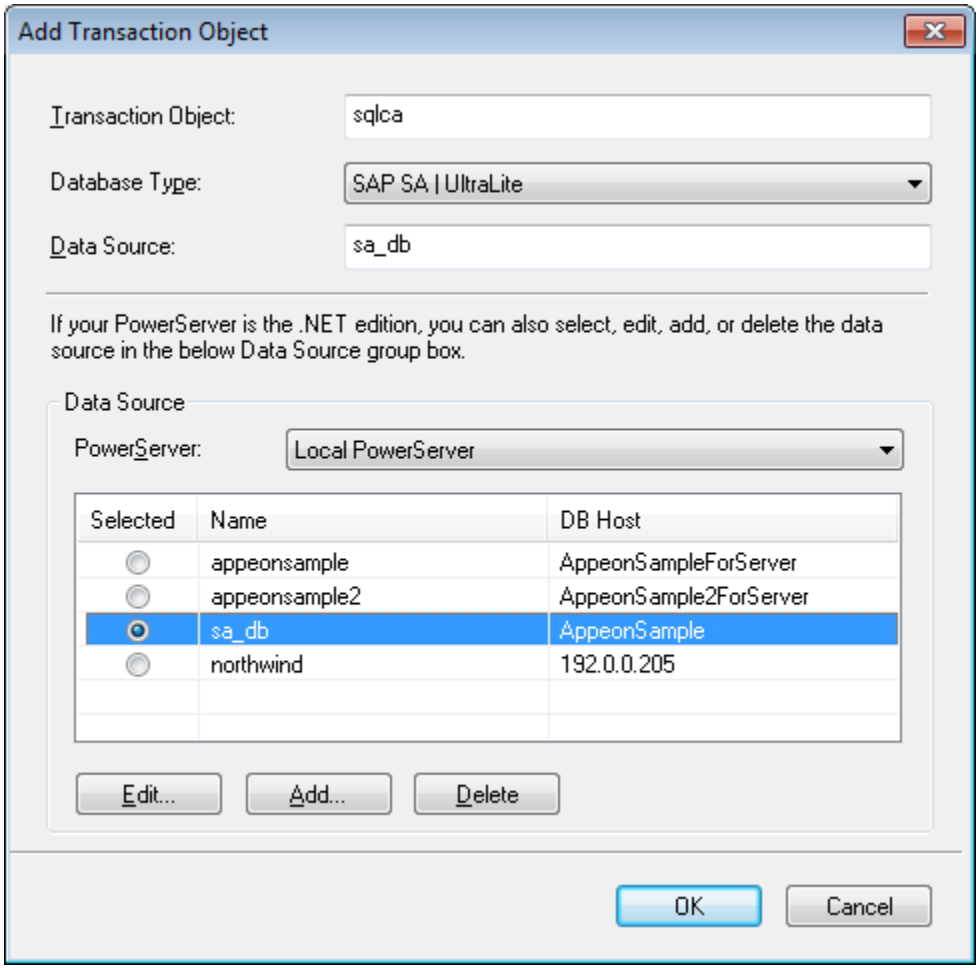
a. Input the transaction object name used by the application to the **Transaction Object** text box. Default transaction object for most PB applications is **SQLCA**.

b. Select the database type from the **Database Type** list box.

c. Select **Local PowerServer** from the **PowerServer** dropdown list box.

d. Select the data source from the table. The data source should connect to the same database that the PowerBuilder application connects to. If the required data source has not been configured in Task 2: Configure the database connection, you should click **Add**, and then follow Step 4 in Task 2: Configure the database connection to create it.

**Figure 3.11: Select the connection cache**




**Tip**: If you have more than one transaction object you can add additional transaction objects by repeating the above steps.

Step 8: Click **Browse** to select the image file or the folder that contains the image files, if any. Click **Next**.

Step 9: Click **Add File** or **Add Directory** to select the INI file, if any. Click **Next**.

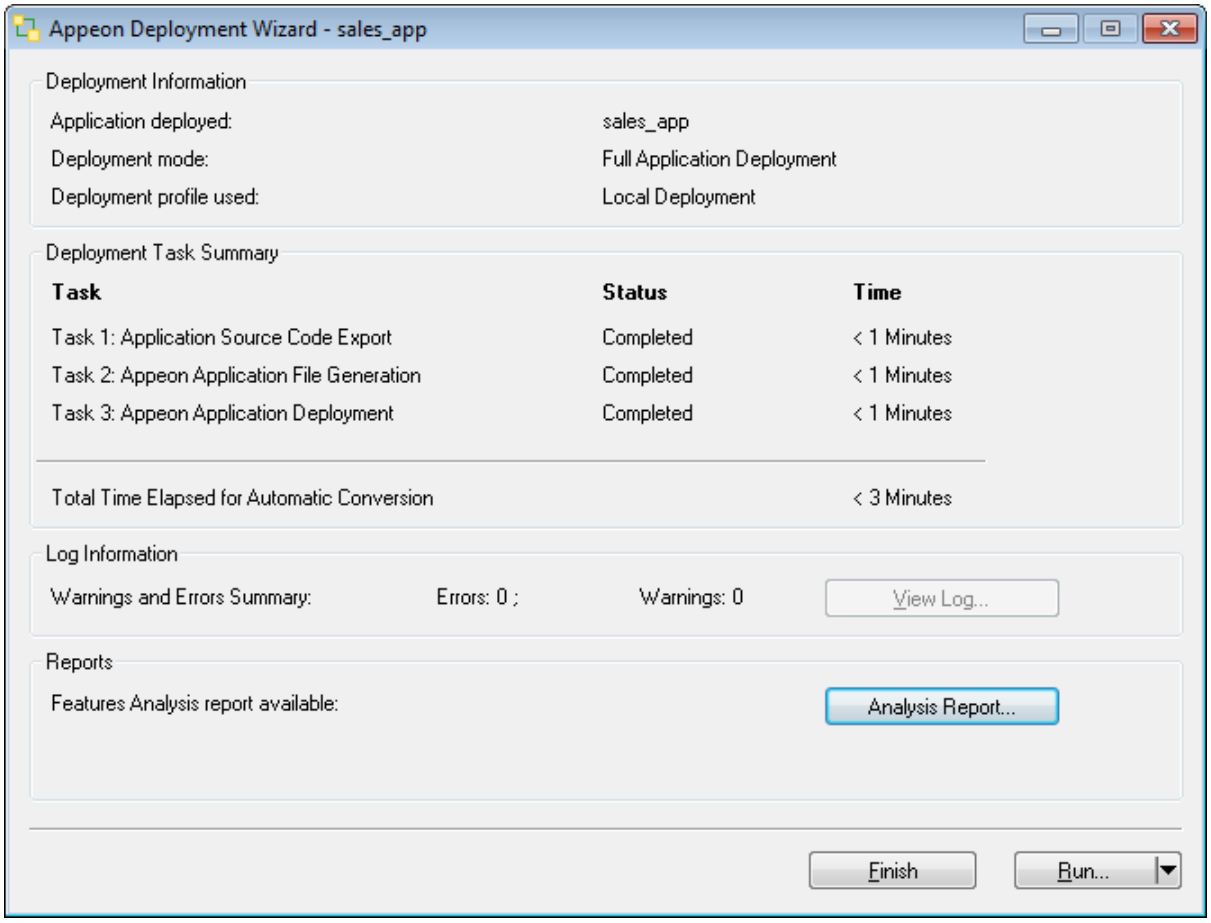
Step 10: Click **Add File** or **Add Directory** to select any other external files such as TXT file etc., if any. Click **Next**.

Step 11: In the configuration summary screen, keep the **Deploy the application now** option as selected. Click **Finish**.

Once you click **Finish**, the **Appeon Deployment Wizard** automatically starts deploying the application. To manually starts **Appeon Deployment Wizard**, you can click the **Deploy** icon (  ) on the PowerServer Toolkit.

Step 12: Click **Finish** when the deployment process is complete.

**Figure 3.12: Appeon Deployment Wizard_sales**




## 3.4 Task 4: Run the Web version of the application (in IE)

Before you run the application on the Android or iOS device, make sure your application can run successfully in the Internet Explorer Web browser (to aid debugging your application is automatically deployed as both a Web application using PowerServer Web and a mobile application using PowerServer Mobile):

Step 1: Open Internet Explorer and navigate to the trusted sites list (**Internet Option > Security > Trusted Sites > Sites**). Uncheck the **HTTPS** checkbox and add the IP address of your IIS as well as the localhost. For example, if your IP address is 192.168.1.117 then it should look as follows:

**Figure 3.13: Add the IP address of IIS**

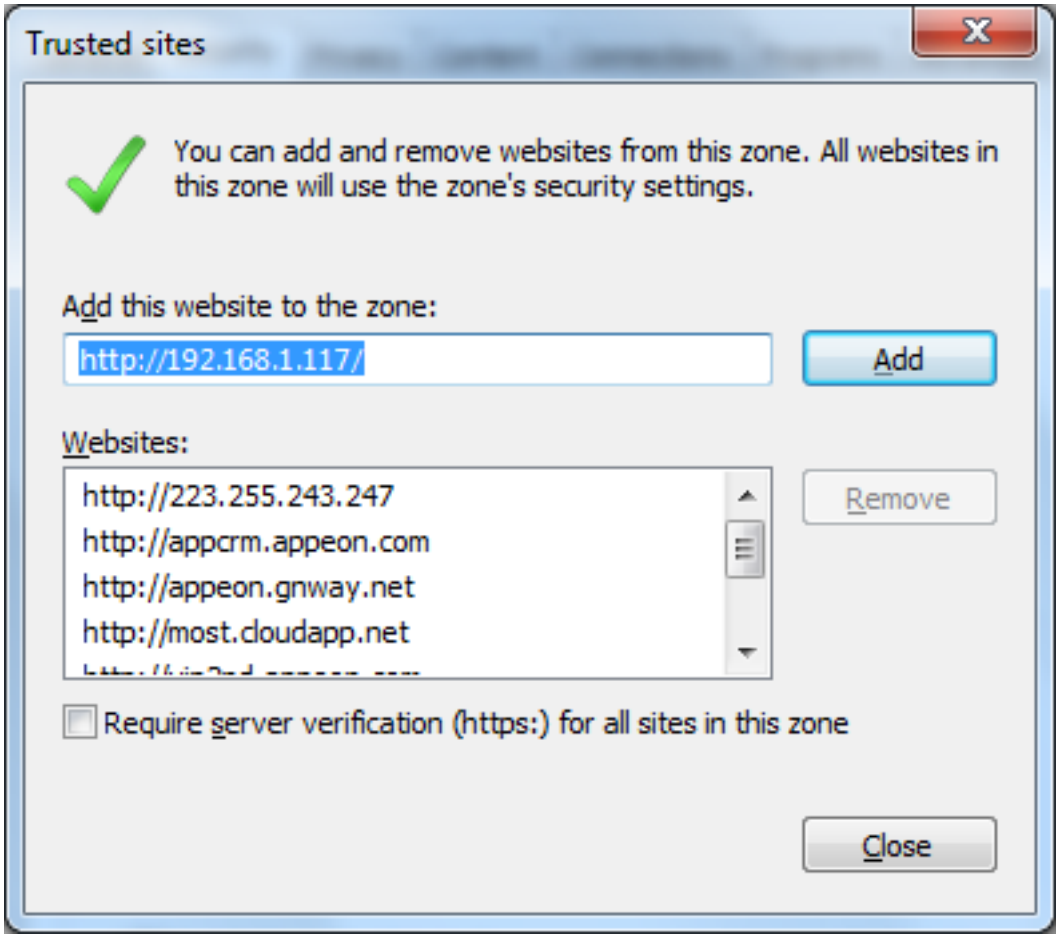
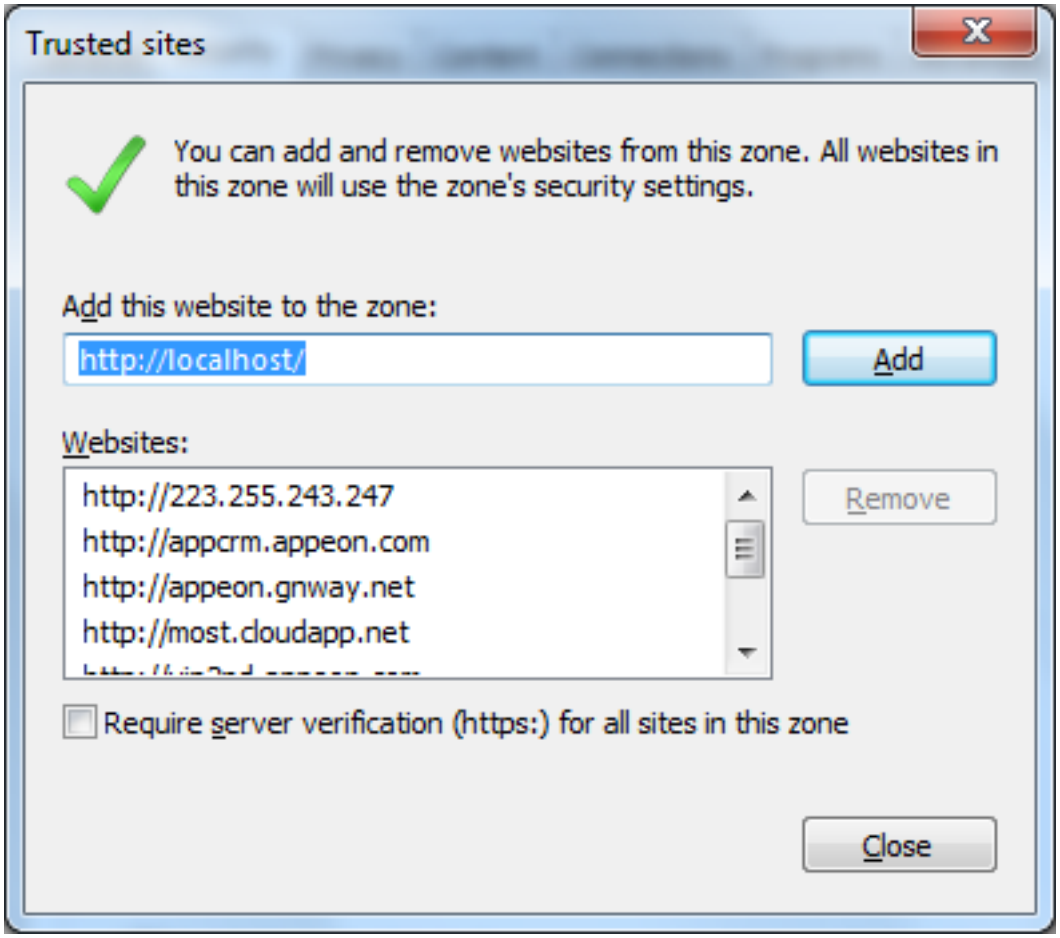


**Figure 3.14: Add "localhost"**




Step 2: Input the application URL in the Internet Explorer address bar and start the application. The application URL must contain index.htm, for example http://192.168.1.117/myApplication/index.htm.

**Note**: The application URL is specified in the **Web Folder** field in Step 1 of Task 3: Configure and deploy the application.

Step 3: Install the Web browser plug-in, when you are prompted. You will be prompted twice, please accept both times.

Step 4: When the application is loaded successfully in Internet Explorer, test it carefully.

If you notice any issues, please debug the application according to the Tutorial 3: Develop & Debug with PowerServer Mobile.

## 3.5 Task 5: Run the mobile version of the application (on the Android or iOS device)

Now you are ready to run your application on the Android or iOS device.

Step 1: Configure the network connection. See Task 4 in Tutorial 1: Set up the Environment.

Step 2: Install Appeon Workspace. See Task 3 in Tutorial 1: Set up the Environment.

Step 3: Tap the **AppeonMobile** icon on your Android or iOS device to launch Appeon Workspace.

Step 4: Tap the **New** icon (  ) to the left of the title bar.

Step 5: In the **App URL** text box, enter the application URL in this format: *http://server_ip:port/app_name*. For example, if your IIS IP address is 192.168.1.117 on port 80 and you specified *myApplication* in the PowerServer Toolkit configuration as the Web folder name then the URL would be http://192.168.1.117:80/myApplication/.

Step 6: Tap the **Test Connection** button to test the server connections. If successful please proceed to Step 7, otherwise please enter the correct URL.

Step 7: Tap the **Back** icon (  ) on the title bar to save the information and return to the main screen of the Appeon Workspace.

Once you return to the main screen of the Appeon Workspace, the downloading and installation process of the application occurs automatically.

Step 8: After the installation process has completed, tap the application icon on the main screen to run the mobile app that is installed.

# 4 Tutorial 3: Develop & Debug with PowerServer Mobile

Please keep in mind the following points when developing and debugging the Appeon mobile applications for the Android or iOS device. For detailed instructions on how to use the PowerServer Toolkit you may refer to PowerServer Toolkit User Guide.

## 4.1 Point 1: Mobile UI Considerations

Many aspects of the UI for mobile applications differ significantly from traditional PowerBuilder best practices. How you layout controls on the window, the size of the window, controls, and other visual objects, and how the user expects to interact with your application are fundamentally different for mobile devices.

Please carefully read Chapter 2, *Best Practices* in *Mobile UI Design & Development Guide*, including the **Window** subsection. We have compiled valuable tips and suggestions that will help you develop user-friendly mobile applications.

Once you have a good understanding of Chapter 2, *Best Practices* in *Mobile UI Design & Development Guide*, we strongly recommend using the formula in Section 2.5.1, “Unit conversion” in *Mobile UI Design & Development Guide* to properly size the application windows and controls to display appropriate on the Android or iOS device.

## 4.2 Point 2: Unsupported Features

PowerServer Mobile supports nearly all the same features as PowerServer Web. Those experienced with PowerServer Web know that it supports many powerful PB features. So you should be able to build equally robust applications with PowerServer Mobile also.

When developing applications for PowerServer Mobile, it is critical to ensure that your application does not contain unsupported features. Appeon provides several tools to assist you with this:

- Tool 1: The **UFA** tool (  ) in the PowerServer Toolkit will automatically scan your application and list major unsupported features found based on a keyword scan.
- Tool 2: The **Code Insight** tool (  ) in the PowerServer Toolkit will pop-up a window, when you are writing dot notation in the PB script editor, that prevents you from writing PowerScript that is unsupported.
- Tool 3: The Supported PB Features for PowerServer Mobile is a searchable online help that lists all unsupported features.

These tools have some limitations. Please keep in mind the following limitations when using these tools:

- The **UFA** tool uses a keyword scan. As such, not all unsupported features can be found, especially those that are dynamically occurring in runtime.

- The **Code Insight** tool is limited to unsupported PowerScript and essentially based on keywords (like the **UFA** tool). As such, it will not prevent all unsupported features from being added to the application, such as unsupported objects, unsupported expressions, etc.

We recommend you develop your mobile applications with PowerServer Toolkit as follows:

Step 1: Use the **Code Insight** tool (  ) from the PowerServer Toolkit to code PowerScript that is compatible with PowerServer Mobile. For details about how to configure and use the **Code Insight** tool, see Chapter 12, *Developing with Code Insight* in *PowerServer Toolkit User Guide*.

Step 2: Run the **UFA** tool (  ) from the PowerServer Toolkit to generate the unsupported features report. For details about how to run the UFA report, see Chapter 5, *Using UFA Tool* in *PowerServer Toolkit User Guide*.

Step 3: Remove or rewrite any unsupported features found by the **UFA** tool. Please make note of the unsupported features found such that in the future you do not reintroduce these same unsupported features.

Step 4: After you have deployed the application, attempt to run the Web version in the Internet Explorer Web browser. The Web version is based on **PowerServer Web** that has been on the market for over 10 years. As such, if you find something in your application is not working in the Web version chances are it is an unsupported feature rather than a product bug.

Step 5: Since the **UFA** tool and **Code Insight** cannot prevent 100% unsupported features, should you find something that does not work in the Web version please go back to the object or area of PowerScript in question and cross-reference that with the Supported PB Features for PowerServer Mobile to see if your application contains any unsupported features.

Step 6: If after cross-referencing the Supported PB Features for PowerServer Mobile you believe your application is free of unsupported features but it fails to run or operate properly please contact <support@appeon.com> for help.

## 4.3 Point 3: Previewing & Debugging

To preview the application UI and layout, run the application in Appeon Workspace on the mobile device. For instructions on installing Appeon Workspace, see Task 3 in Tutorial 1: Set up the Environment.

To debug the application business logic, first ensure that the UFA report does not contain any unsupported features, or at least no unsupported features in the area of the application you are trying to debug. Then, launch the **Appeon Debugger** (  ) from the PowerServer Toolkit, which will load the Web version of your application in the IE Web browser.

The **Appeon Debugger** requires the Microsoft Script debugger to be installed on your system and that your Internet Explorer is version 9 or later. There are also special configurations you need to perform in the PowerServer Toolkit and Internet Explorer. Please see Chapter 7, *Debugging Appeon Web Applications* in *PowerServer Toolkit User Guide* for more information about how to properly configure your environment for the Appeon Debugger.

**Please note that PowerServer Mobile does not contain a mobile-specific debugger, so we have included the Web debugger in lieu.** While not exactly the same, it can still help

to identify problems in the application business logic or areas that Appeon is having trouble converting.

# 5 Tutorial 4: Package & Distribute Native Mobile Apps

## 5.1 Overview

After you successfully deploy a PowerBuilder application to be a native mobile application using the PowerServer Toolkit, you can choose to run the mobile app immediately in Appeon Workspace (see Appeon Workspace User Guide for detailed instructions), or package the app and publish it to the online application store (such as Apple App Store, Google Play etc.) or distribute it over-the-air (such as by email, Web site etc.). In this section, you will go through all the required steps for packaging and distributing a PowerServer Mobile app as a standalone native app.

**Note**: these steps are also applicable for packaging and distributing Appeon Workspace, as intrinsically Appeon Workspace is also a standalone native mobile app.

## 5.2 Package & Distribute iOS Apps

You will need to go through the following tasks to package and distribute the PowerServer Mobile app as a standalone native iOS app.

1. Task 1: Prepare for the build environment.
2. Task 2: Generate the Xcode project.
3. Task 3: Create the app archive.
4. Task 4: Distribute the app archive

### 5.2.1 Task 1: Prepare for the build environment

#### 5.2.1.1 Task 1.1: Register for an Apple ID

Follow the onscreen instructions on My Apple ID to create an Apple ID if you do not have one.

#### 5.2.1.2 Task 1.2: Prepare the Mac machine

Buy a Mac machine and install the latest Xcode on the Mac machine.

You will need to have a Mac machine with the Xcode tool installed already to create and install the distribution certificate and the provisioning profile; run the Xcode project of the application and create the app archive; distribute the app archive; etc.

##### 5.2.1.2.1 Install Xcode

Click https://developer.apple.com/xcode/ to download the latest Xcode, which is 6.1 by the time of writing.

Sign in with your Apple ID, follow the onscreen instructions to register your Apple ID as an Apple developer, and download and install the latest Xcode tool from the above mentioned website.

For more information on Xcode, see the documentation provided on the Apple website.

#### 5.2.1.3 Task 1.3: Enroll in an iOS Developer Program

Enroll in an iOS Developer program at https://developer.apple.com/programs/.

Carefully compare the programs and choose one that suits you best, because different distribution types (App Store, Ad Hoc, In-House, etc.) will be available in different programs, for example, **App Store Distribution** is available in **iOS Developer** program only, while **In-House Distribution** is available in **iOS Developer Enterprise** program only.

Below is a brief introduction to the three distribution types that are commonly used (detailed instructions for these distribution types are provided with in Section 5.2.4, "Task 4: Distribute the app archive"):

- **App Store Distribution**: allows you to submit the app for publication to the Apple App Store.

  You would need to enroll in the **iOS Developer** program.

- **In-House Distribution**: allows you to install the app to an unlimited number of devices inside your company.

  You would need to enroll in the **iOS Developer Enterprise** program.

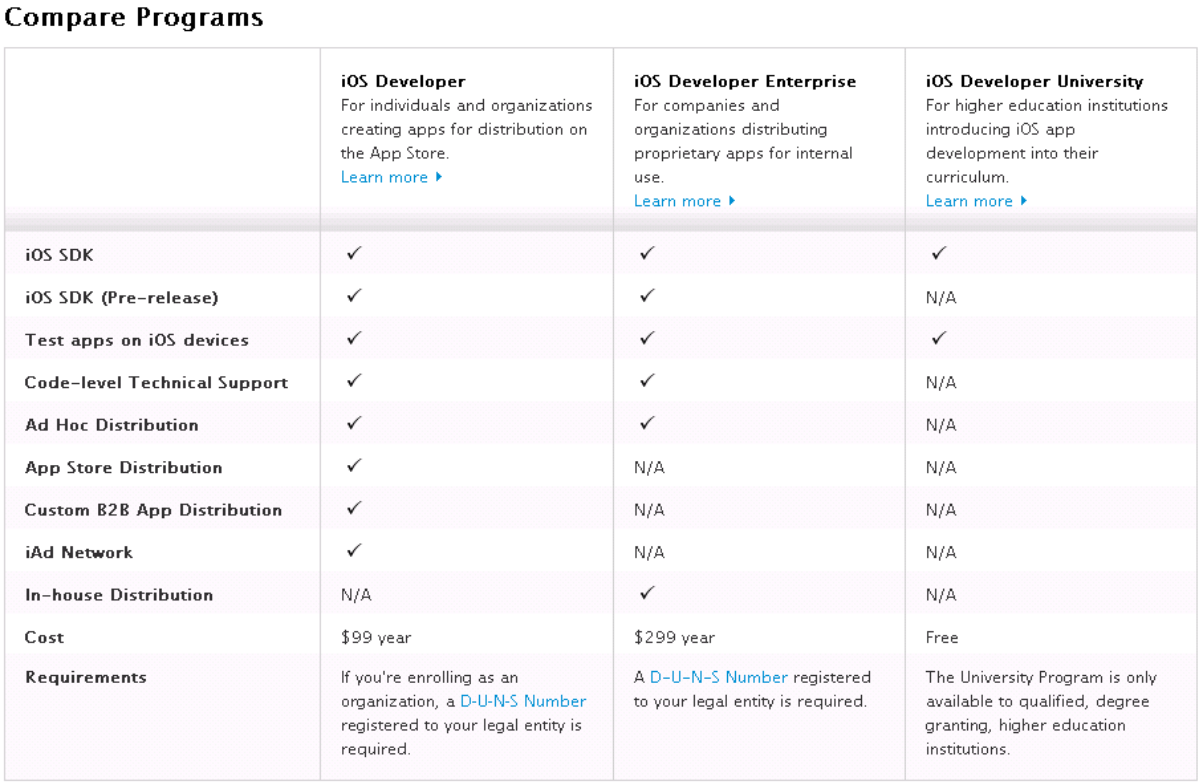
- **Ad Hoc Distribution**: allows you to install the app on a limited number (up to 100) of registered devices.

  You would need to enroll either in the **iOS Developer** program or the **iOS Developer Enterprise** program. And you would need to register the devices by their unique device ID (UDID) and add them to the provisioning profile. For details, refer to Registering Devices Using Member Center in the Apple document *App Distribution Guide*.

**IMPORTANT NOTE:**

If you enroll in the **iOS Enterprise Program**, you are responsible for managing the Enterprise Program certificate and provisioning profile. According to the Apple policy, the **iOS Enterprise Program** certificates expire after three years and provisioning profiles expire after one year, which means, before the distribution certificate expires, you will need to request an additional distribution certificate and replace the expired one, and before the provisioning profile expires, you will need to renew the provisioning profile using a valid certificate. Otherwise the mobile app distributed using the enterprise provisioning profile will not launch after expiration. For more details, refer to Managing Expiring Certificates and Provisioning Profiles in the Apple document *App Distribution Guide*.

**Figure 5.1: Compare programs**



Compare Programs

| | iOS Developer<br>For individuals and organizations creating apps for distribution on the App Store.<br>Learn more | iOS Developer Enterprise<br>For companies and organizations distributing proprietary apps for internal use.<br>Learn more | iOS Developer University<br>For higher education institutions introducing iOS app development into their curriculum.<br>Learn more |
|---|---|---|---|
| iOS SDK | ✓ | ✓ | ✓ |
| iOS SDK (Pre-release) | ✓ | ✓ | N/A |
| Test apps on iOS devices | ✓ | ✓ | ✓ |
| Code-level Technical Support | ✓ | ✓ | N/A |
| Ad Hoc Distribution | ✓ | ✓ | N/A |
| App Store Distribution | ✓ | N/A | N/A |
| Custom B2B App Distribution | ✓ | N/A | N/A |
| iAd Network | ✓ | N/A | N/A |
| In-house Distribution | N/A | ✓ | N/A |
| Cost | $99 year | $299 year | Free |
| Requirements | If you're enrolling as an organization, a D-U-N-S Number registered to your legal entity is required. | A D-U-N-S Number registered to your legal entity is required. | The University Program is only available to qualified, degree granting, higher education institutions. |

#### 5.2.1.4 Task 1.4: Create an App ID

An App ID is required when you create the provisioning profile and when you compile the Xcode project to be an IPA file later.

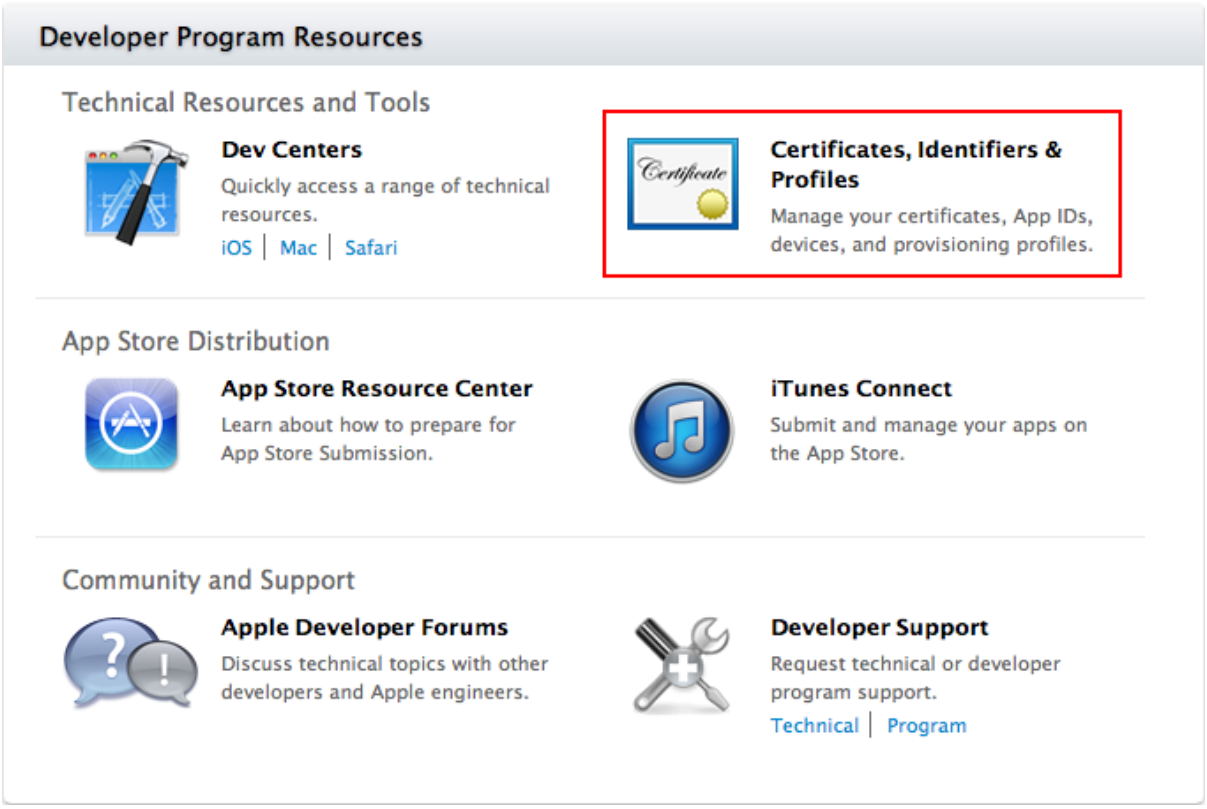
An App ID is a unique identifier for the app. It is composed of two parts: the App ID Prefix and the App ID Suffix (also called Bundle Identifier). The App ID Prefix is a 10-character hexadecimal string generated by the iOS Certificates, Identifiers & Profiles page. It is unique to you and your developer account. The App ID Suffix is a name you enter called the Bundle Identifier. The Bundle Identifier can be explicit or a wildcard. Wildcard Bundle IDs are great for quick development – you do not have to create a new ID for each and every app you test. We will use wildcard Bundle IDs in this tutorial.

To create an App ID:

Step 1: On your Mac machine, log in to the Apple Developer Member Center with your Apple ID and password.

Step 2: Click the icon or text next to **Certificates, Identifiers & Profiles**.

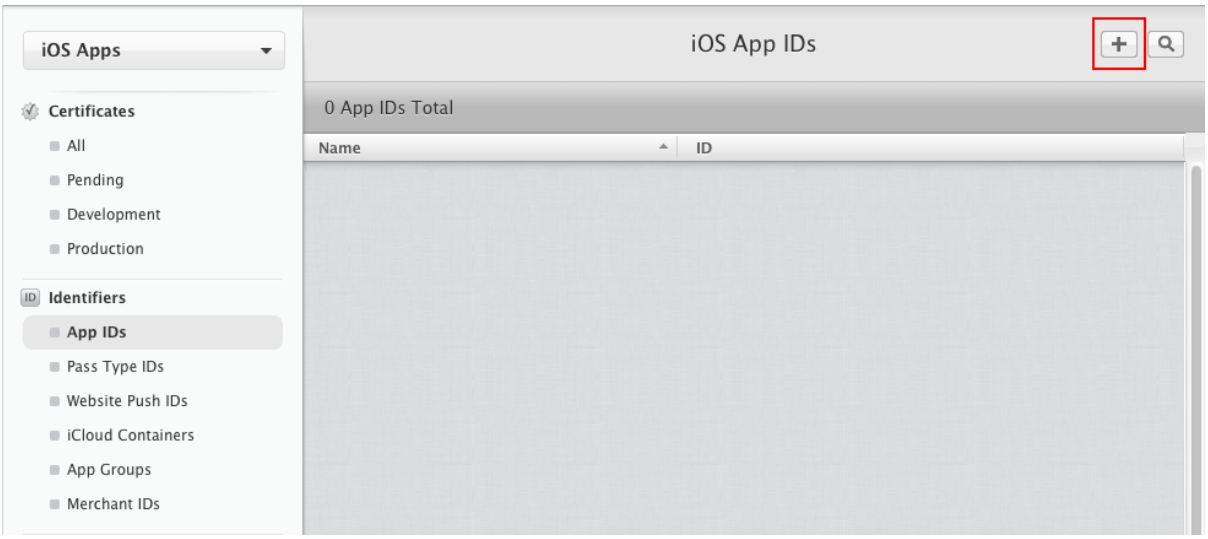
**Figure 5.2: Certificates, Identifiers & Profiles**




Step 3: Click **Identifiers**, then click the plus sign (+) button near the top-right corner.

**Figure 5.3: Identifiers**


**Figure 5.4: Add an App ID**

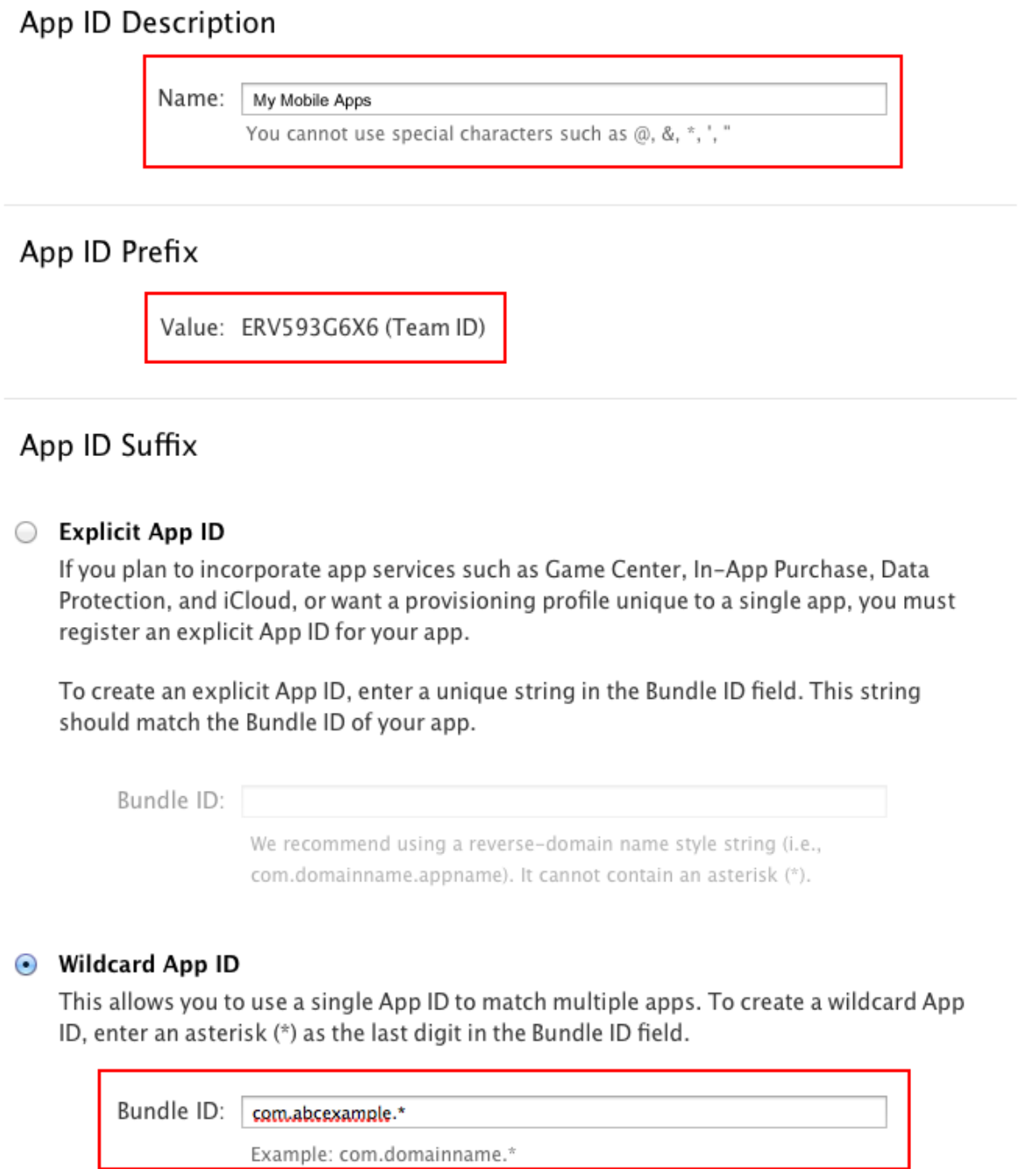



Step 4: Enter a description, for example, "My Mobile Apps", which cannot include special characters (including most punctuation).

Step 5: For App ID Prefix, understand that it is automatically generated by the iOS Certificates, Identifiers & Profiles page.

Step 6: For App ID Suffix, choose **Wildcard App ID** to use a single ID to match multiple applications. Enter an asterisk (*) as the last digit of the Bundle ID. For example, "com.abcexample.*".

**Figure 5.5: Information for the App ID**




Step 7: Click **Continue**.

Step 8: Confirm the App ID information and click **Submit**.

Remember the App ID, especially the App ID Suffix (Bundle ID), as it will be used in the Xcode project settings and the iTunes Connect app record.

#### 5.2.1.5 Task 1.5: Create & install a distribution certificate

A distribution certificate is required to sign the code and create an app for testing and submitting to the Apple Store etc.; it is different from a development certificate which is used for development purpose in Xcode. You must have a valid distribution certificate linked to a distribution provisioning profile.

Note that if you use an **iOS Enterprise Program** distribution certificate, the certificate will expire after three years, you will need to request an additional certificate to replace the old one before expiration. For more, refer to Requesting Additional Enterprise Distribution Certificates in the Apple document *App Distribution Guide*.

##### 5.2.1.5.1 Create a CSR file

To generate a distribution certificate, you will first need to create a Certificate Signing Request (CSR) from your Mac. Follow the instructions below to create a CSR file using Keychain Access.

To create a CSR file:

Step 1: Open **Finder** on your Mac, select the **Application** folder, then open the **Utilities** folder, and launch **Keychain Access**.

**Figure 5.6: Utilities folder**

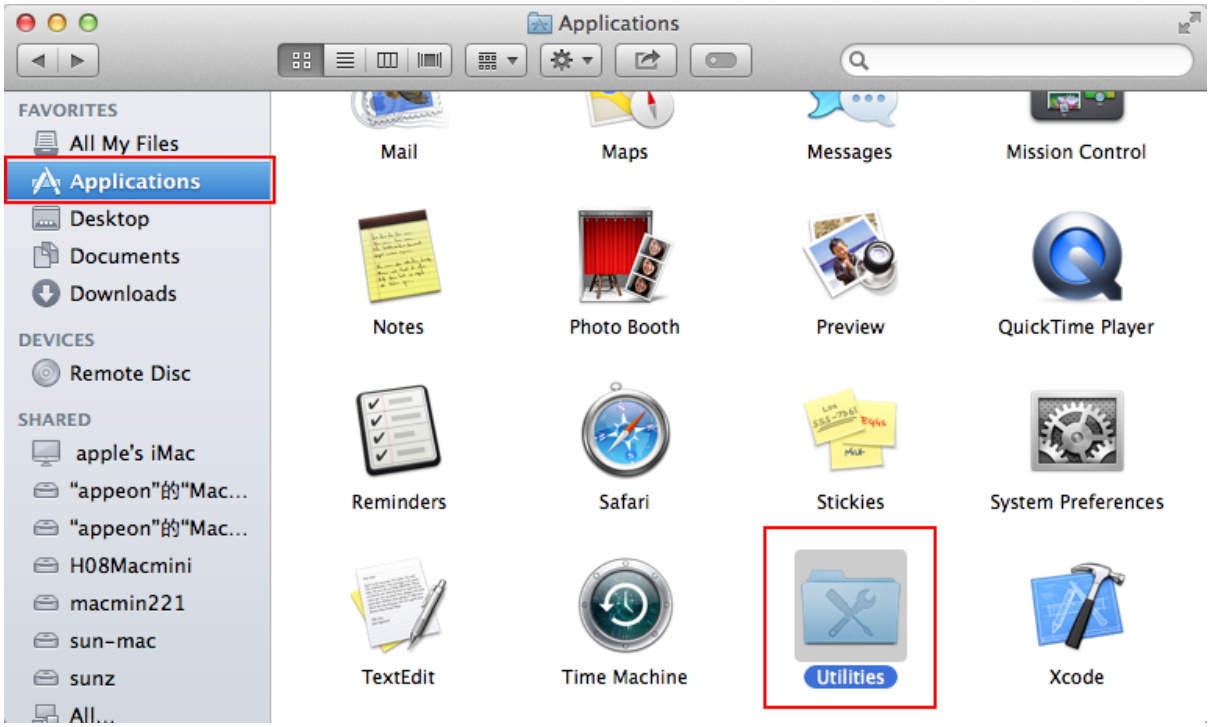
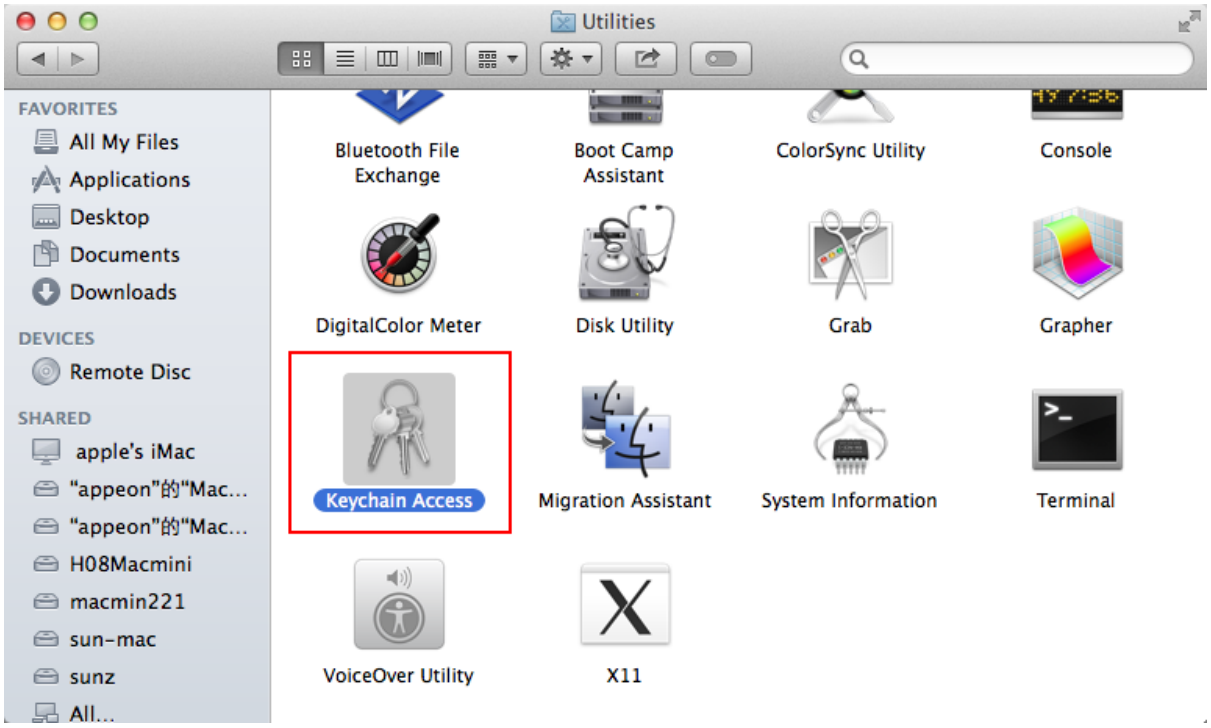
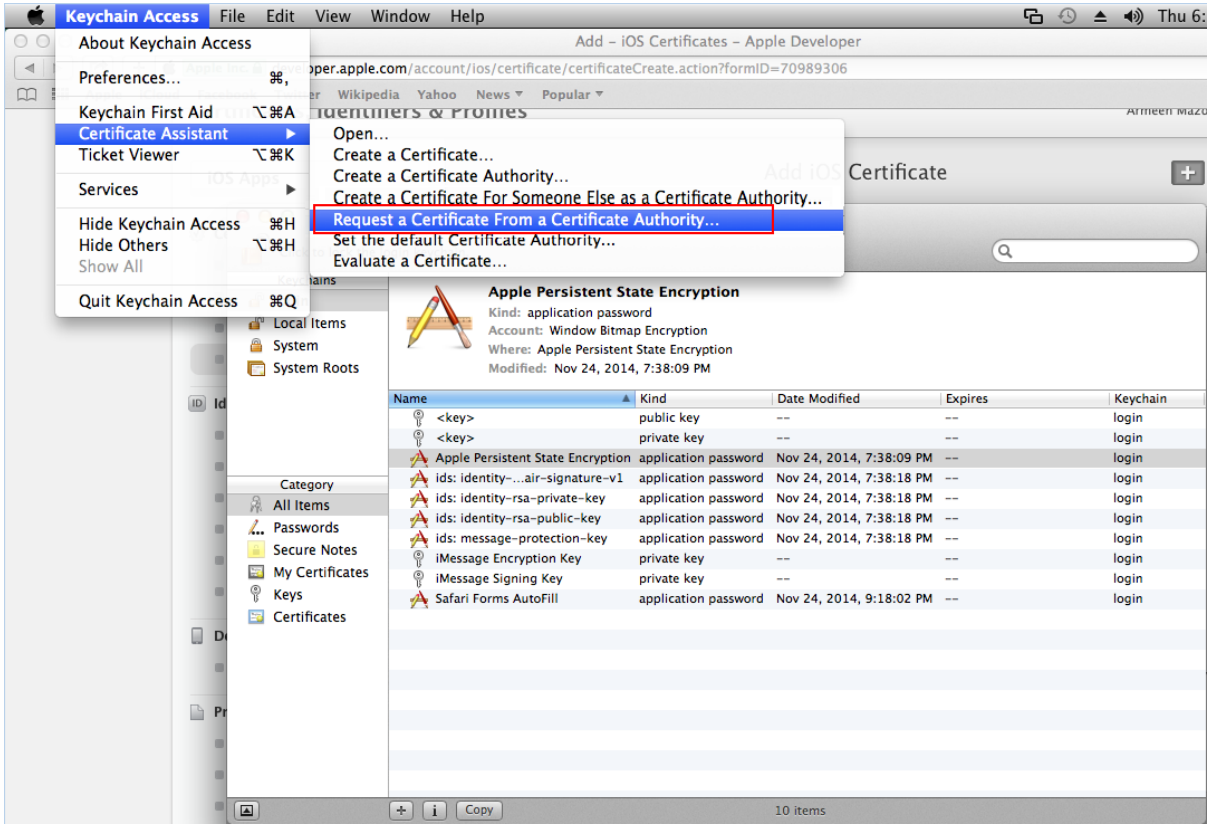


**Figure 5.7: Keychain Access application**




Step 2: Within the **Keychain Access** drop down menu, select Keychain Access > Certificate Assistant > Request a Certificate from a Certificate Authority.

**Figure 5.8: Request a certificate from a certificate authority**

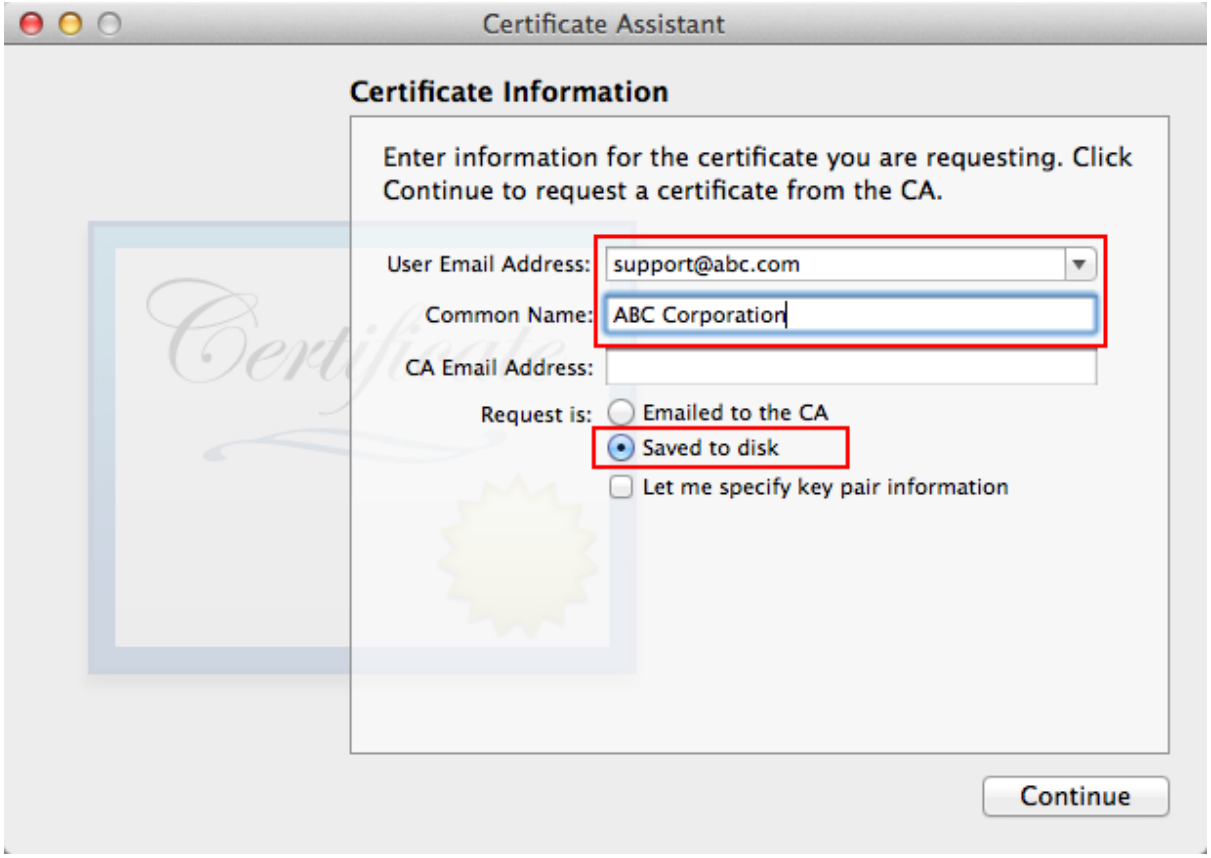


Step 3: In the **Keychain Information** window, enter the following information:

- In the **User Email Address** field, enter your email address. Use the same email address you used to register in the iOS Developer Program.
- In the **Common Name** field, create a name for your private key. For example, ABC Distribution Key.
- The **CA Email Address** field should be left empty.
- In the "**Request is**" group, select the "**Save to disk**" option.

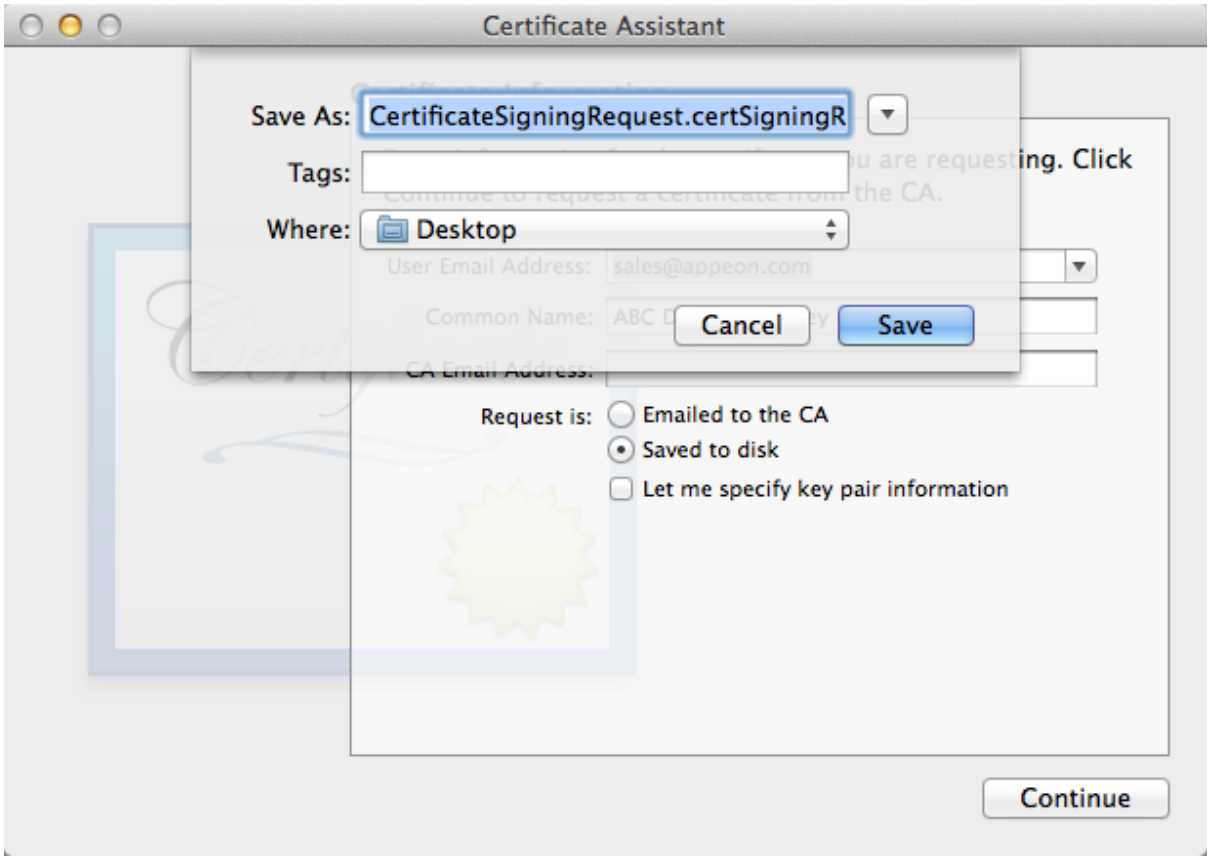
Step 4: Click **Continue**.

**Figure 5.9: Certificate information**




Step 5: Click **Save** to save the CSR file to your Mac desktop. You will need to upload this CSR file when you create the distribution certificate in the next section.

**Figure 5.10: Save CSR to Desktop**




#### 5.2.1.5.2 Create & install a distribution certificate

Now you should go back to the **Apple Developer Member Center** to create a distribution certificate.

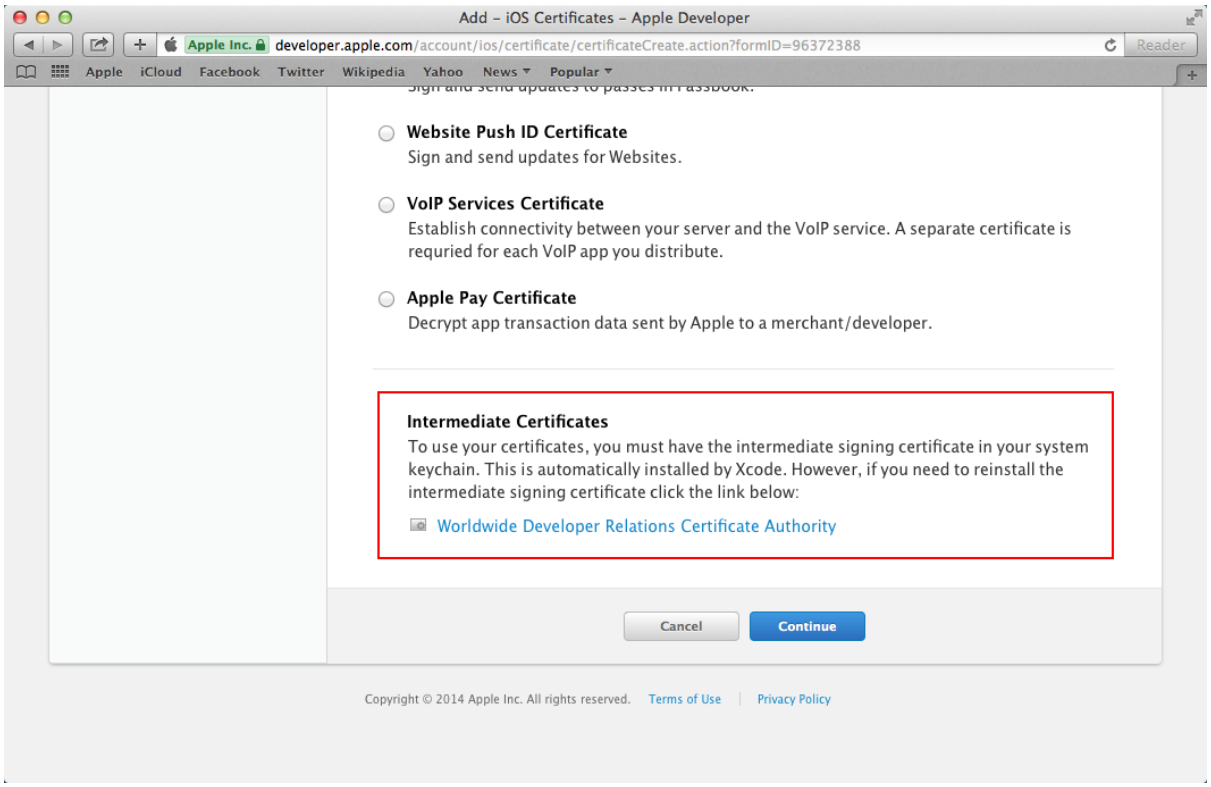
Step 1: In the **Certificates, Identifiers & Profiles** page of the Member Center, choose **Certificates** > **Production**, then click the plus sign (+) button near the top-right corner.

**Figure 5.11: Add a certificate**


Step 2: First drag down the page and click **Worldwide Developer Relations Certificate Authority** to download AppleWWDRCA.cer to your Mac. Then click AppleWWDRCA.cer

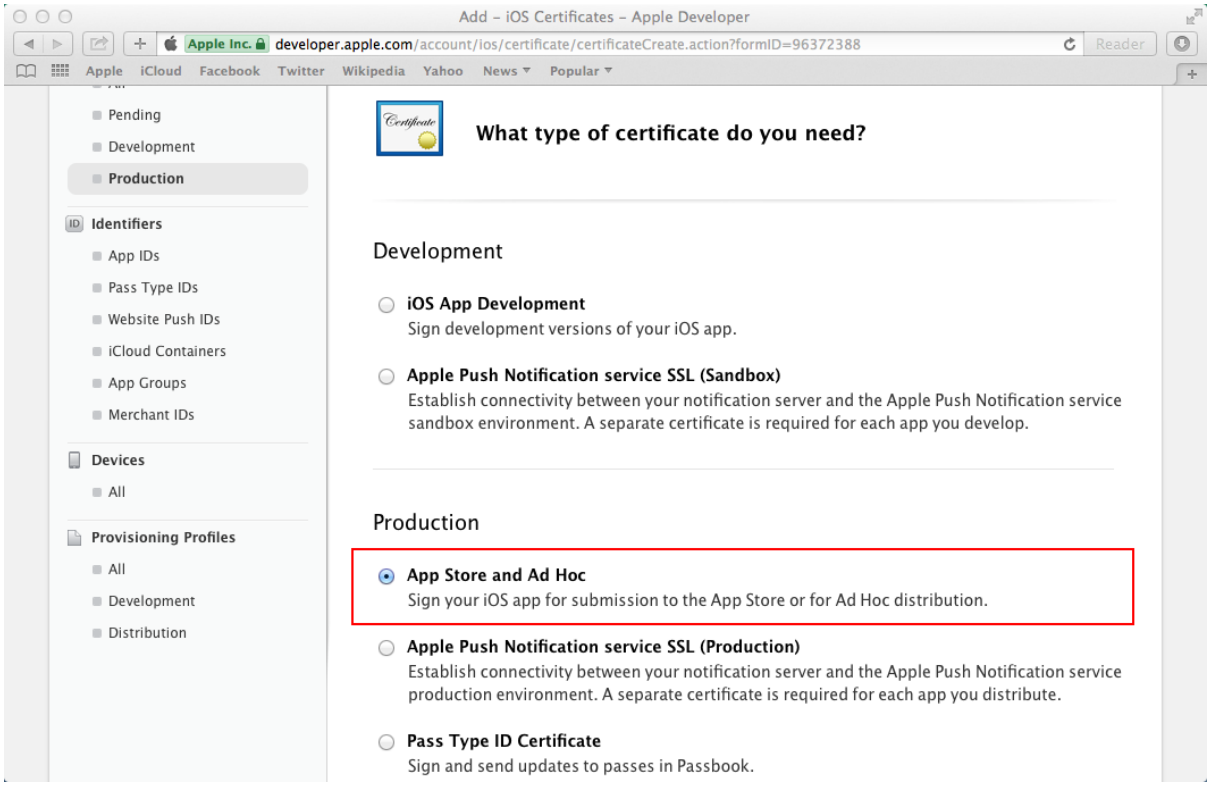
in the **Downloads** folder on your Mac to add the certificate to Keychain Access. You should be able to see the "Apple Worldwide Developer Relations Certificate Authority" certificate listed in the Keychain Access Certificates section.

**Figure 5.12: Download the AppleWWDRCA.cer**




Step 3: Return to the page for adding the iOS certificate, select a certificate type (**App Store and Ad Hoc**), then click **Continue**.

**Figure 5.13: App Store and Ad Hoc distribution**




Step 4: If you have already created a Certificate Signing Request (CSR) file in the previous section, click **Continue** to upload the CSR file. If not, you could follow the directions on this page or in the previous section Create a CSR file to create one.

Step 5: Click **Choose File** to select your CSR and click **Generate**.

**Figure 5.14: Upload CSR file**

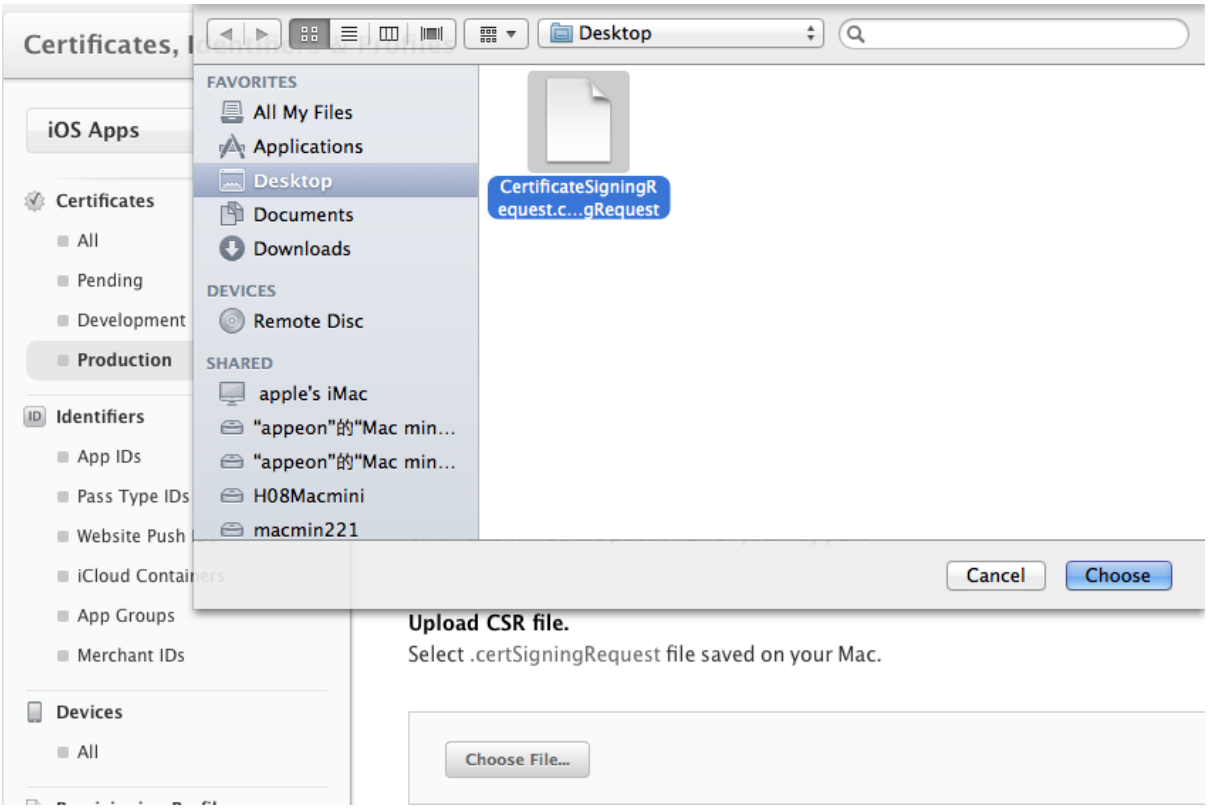
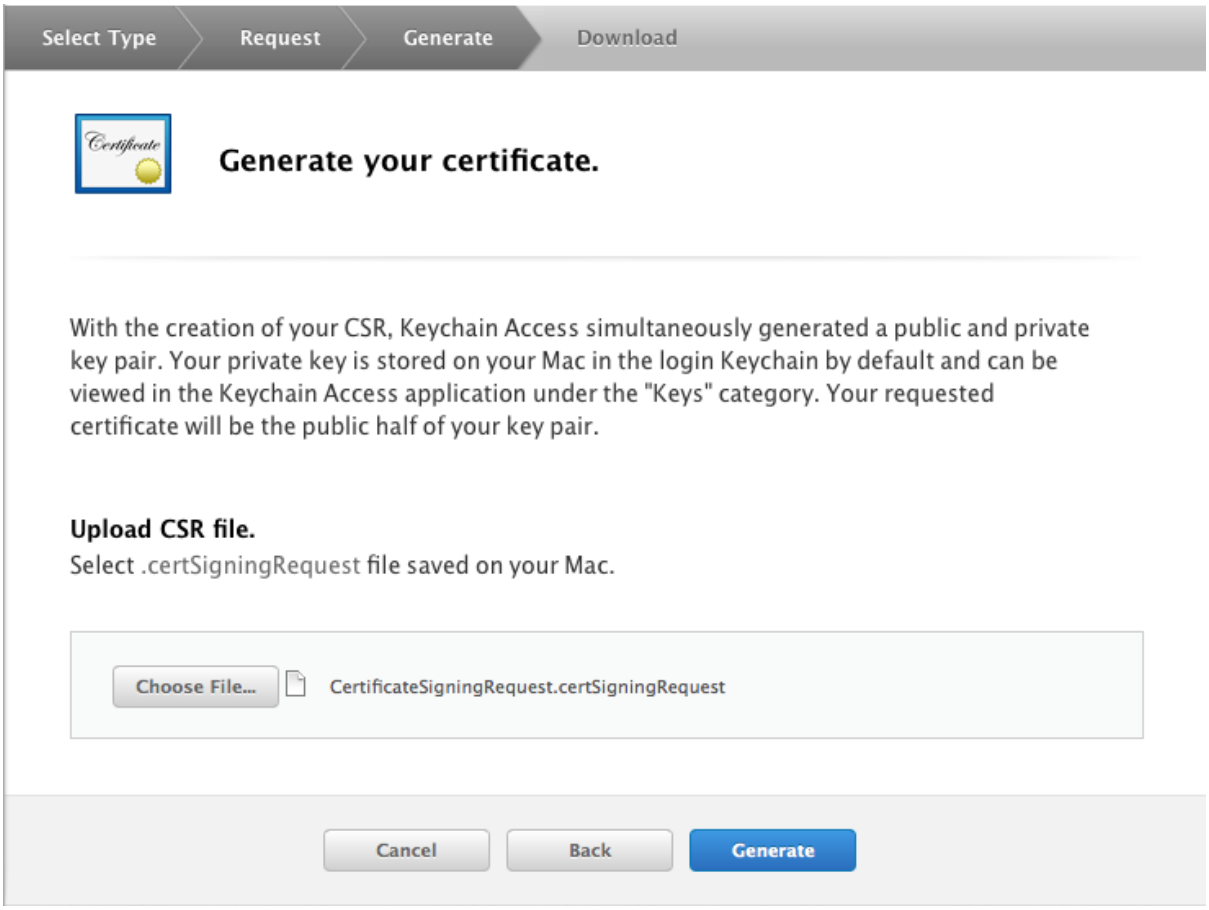


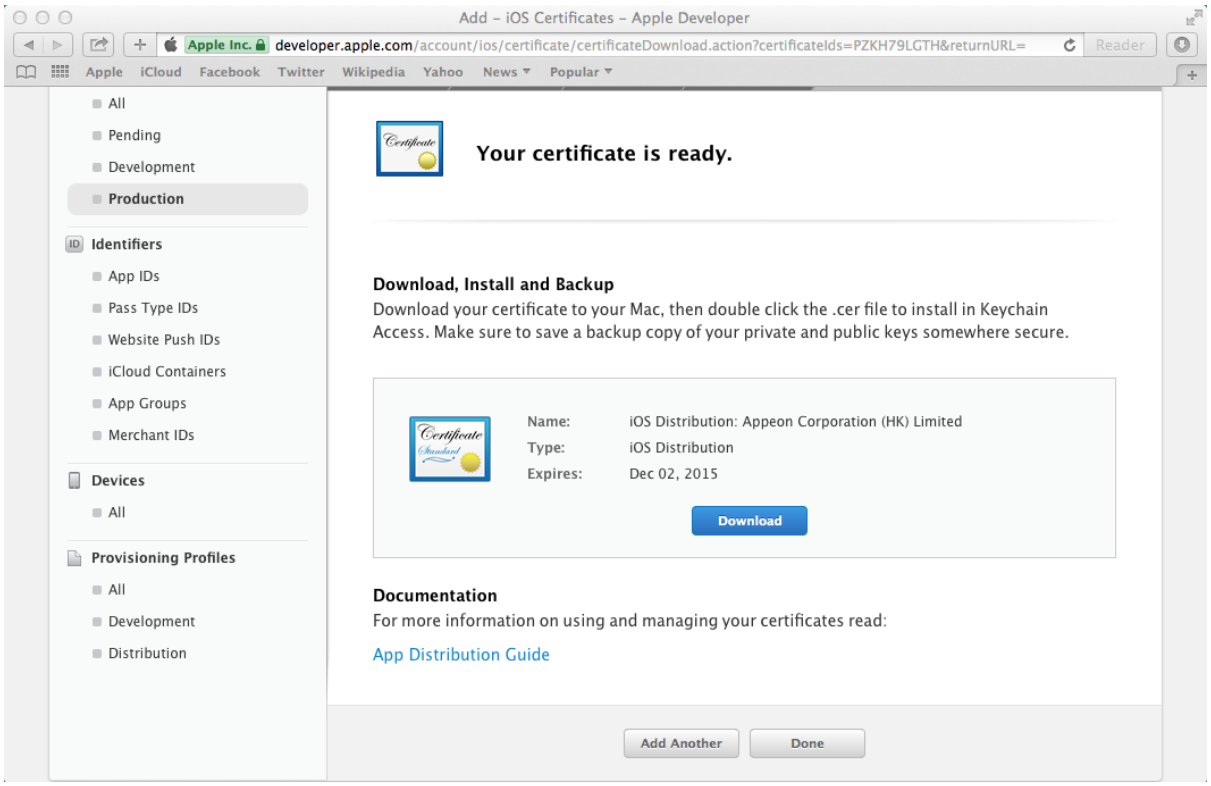

**Figure 5.15: Generate a certificate**



Step 6: Click **Download** to download the distribution certificate to your Mac.

The distribution certificate (ios_distribution.cer) appears in your **Downloads** folder on your Mac.

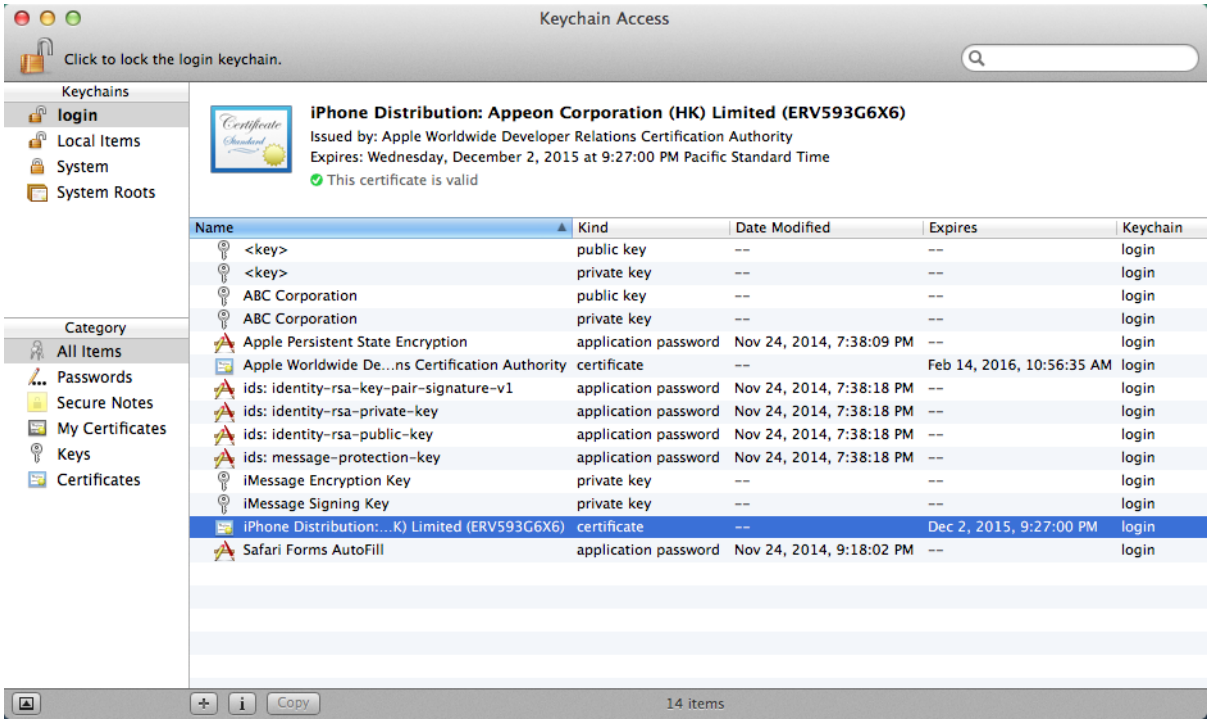
**Figure 5.16: Download Distribution Certificate**




Step 7: Click the distribution certificate (ios_distribution.cer) in the **Downloads** folder on your Mac to add it to **Keychain Access**. The certificate is then listed in the Keychain Access Certificates section, as shown in the following figure.

Remember the name of the certificate, as you will need to use it later when creating the iOS application archive (IPA) file.

**Figure 5.17: Installed Certificate**




#### 5.2.1.6 Task 1.6: Create & install a distribution provisioning profile

Depending on how you will distribute the app (App Store, In-House, or Ad Hoc), you need to create different distribution provisioning profiles.

The steps for creating the different distribution provisioning profiles are similar. To create a **Store** provisioning profile or an **In-House** provisioning profile, you select 1) an App ID and 2) a single distribution certificate. To create an **Ad Hoc** provisioning profile, you select 1) an App ID, 2) a single distribution certificate, and 3) multiple test devices. For details, refer to the below steps, or refer to Creating Store Provisioning Profiles, and Creating Ad Hoc Provisioning Profiles in the Apple document *App Distribution Guide*.

Step 1: Go back to **Certificates, Identifiers & Profiles**, then choose **Provisioning Profiles** > **Distribution**, and then click the add icon ( ).

Step 2: Select a distribution type and click **Continue** in the **Select Type** tag.

You will be provided with different distribution types according to the programs you enrolled in. If you enrolled in the **iOS Developer Program**, you would choose between **App Store** and **Ad Hoc**; if you enrolled in the **iOS Developer Enterprise Program**, you would choose between **Ad Hoc** and **In-House**.

Note that the **In-House** or **Ad Hoc** provisioning profile that is created using the **iOS Enterprise Program** distribution certificate will expire after one year. You will need to renew the provisioning profile before expiration, otherwise the mobile app distributed using these provisioning profiles will not launch after expiration. For more, refer to Renewing Expired Provisioning Profiles in the Apple document *App Distribution Guide*.

Step 3: Select your App ID and click **Continue**.

Step 4: Select the distribution certificate you have created in Task 1.4: Create & install distribution certificates and click **Continue**.

Step 5: Select the devices you want to use for testing, and click **Continue**.

**Note**: This step is only required for creating the **Ad Hoc** provisioning profile.

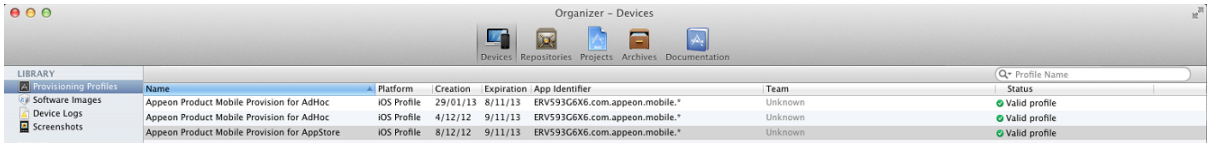
Step 6: Enter a profile name and click **Generate**.

Step 7: After the profile is generated, click **Download** to download and use it.

Step 8: Double-click the downloaded file to install the provisioning profile in Xcode.

Step 9: In **Xcode**, choose **Window** > **Organizer**, and then click **Device**. The installed provisioning file is listed.

**Figure 5.18: Installed provisioning profile**




Now that you have finished creating and installing the distribution certificate and the provisioning profile containing code signing, you are ready to create the app archive.

### 5.2.2 Task 2: Generate the Xcode project

Step 1: On the PowerServer Toolkit machine, use the **Package** tool in the PowerServer Toolkit to package the app files into an Xcode project.

You can select **Package a Stand-alone Mobile Project** (for packaging an Appeon mobile app as a mobile project) or **Customize and Package Appeon Workspace** (for packaging the Appeon Workspace as a mobile project) to package the app files into an Xcode project, and the steps (in this task as well as the subsequent tasks) are the same for both projects.

During the package process, you will be able to specify the settings, such as the app name, icons, etc., that are required by the Xcode project. See Section 10.3, “Packaging a stand-alone mobile project” in *PowerServer Toolkit User Guide* or Section 10.4, “Customizing and packaging Appeon Workspace” in *PowerServer Toolkit User Guide* for detailed instructions.

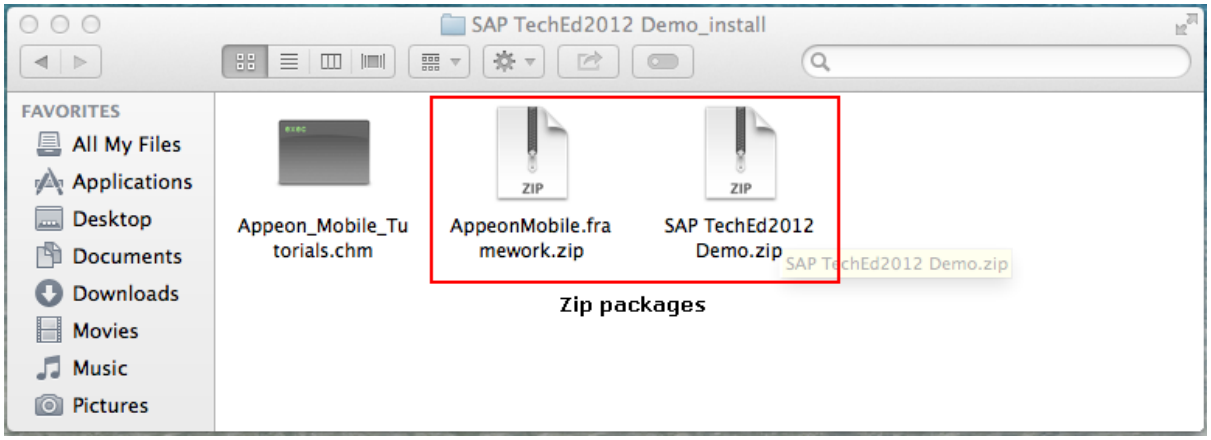
After the package process is complete, you will find the following two zip packages generated under the specified destination folder.

- The zip package of the application.
- AppeonMobile.framework.zip: this zip package is necessary for all apps (including the customized Appeon Workspace) to be distributed on iOS.

These two zip packages are compressed according to the standard of Mac, therefore, they have to be decompressed on the Mac machine.

Step 2: Copy the above two zip packages to the Mac machine.

**Figure 5.19: Zip packages on the Mac machine**




Step 3: On the Mac machine, extract the above two zip packages respectively.

**Figure 5.20: Extracting the packages on the Mac machine**

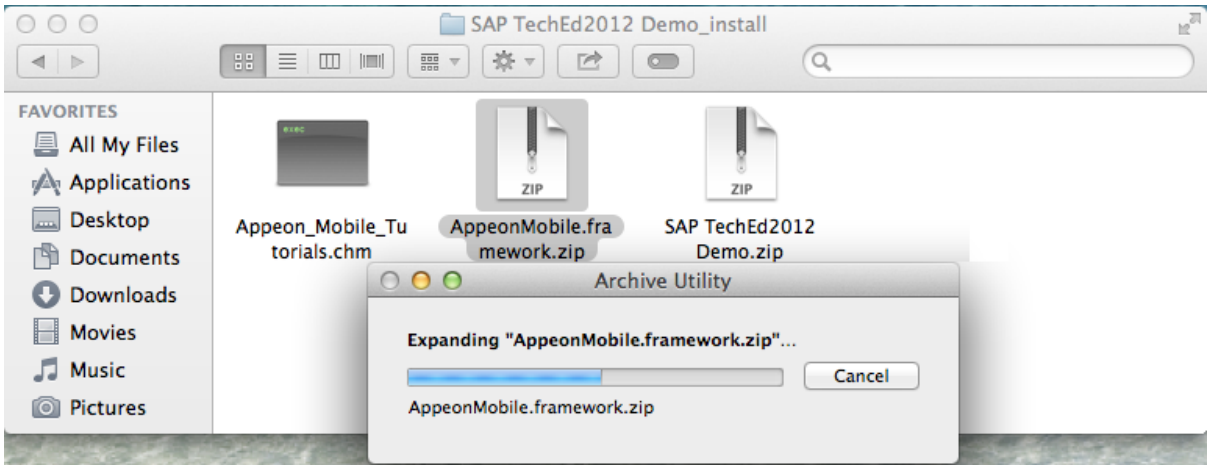
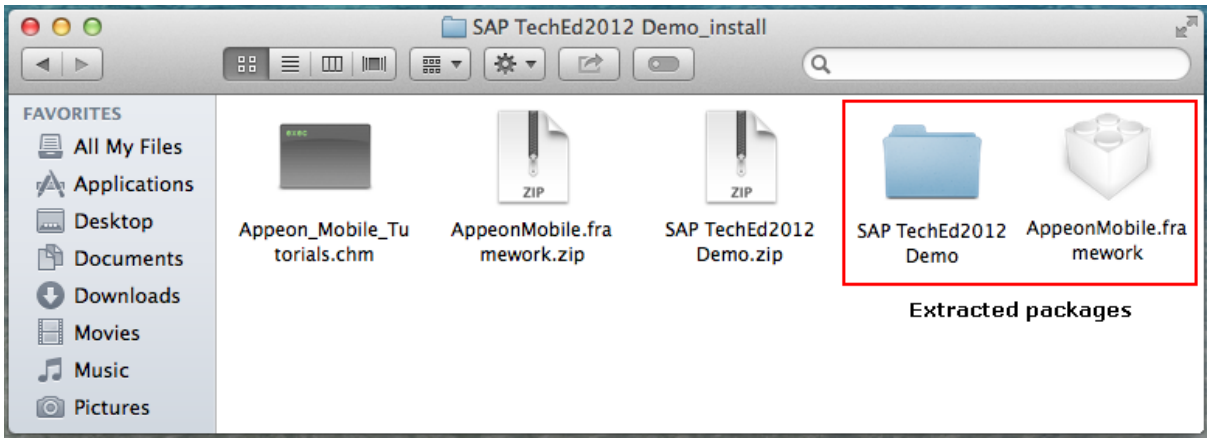


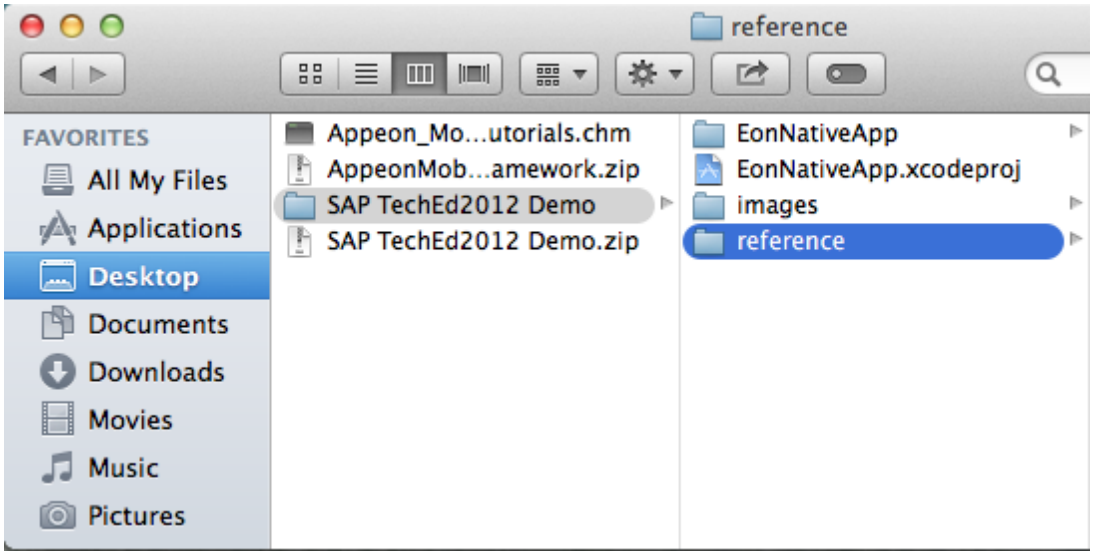

**Figure 5.21: Extracted packages on the Mac machine**




Step 4: Copy all of the files under the extracted **AppeonMobile.framework** folder into the **reference** folder under the extracted app package.

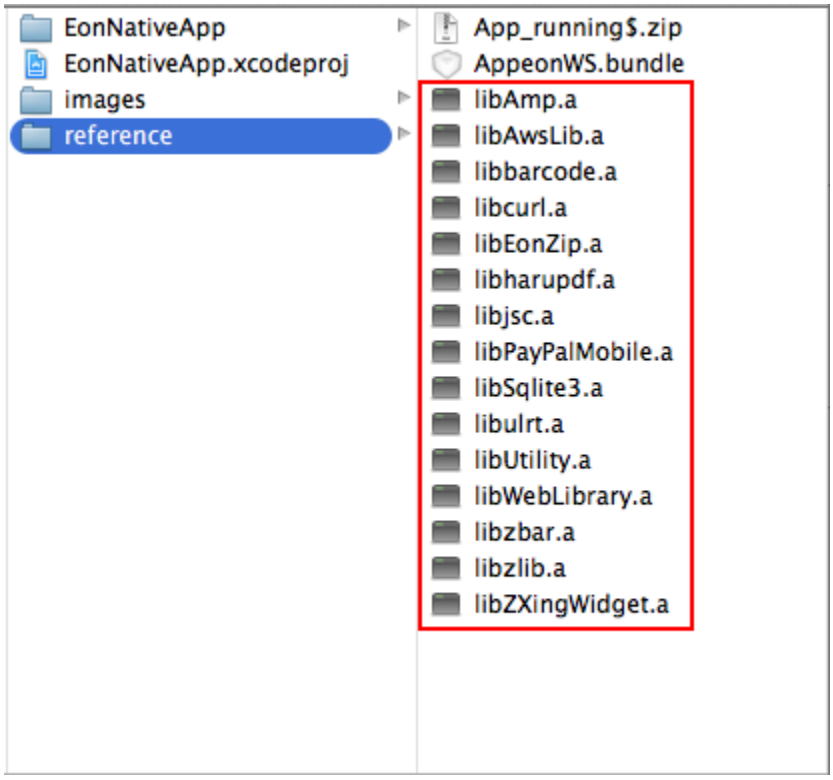
The following screenshot shows the location of the **reference** folder under the extracted app package.

**Figure 5.22: Location of the reference folder**




The following screenshot shows the **reference** folder that contains all of the files that were copied from the **AppeonMobile.framework** folder.

**Figure 5.23: Files contained in the reference folder**




Make sure you do Step 3 & 4 on the Mac machine, not on the Windows machine.

## 5.2.3 Task 3: Create the app archive

### 5.2.3.1 Task 3.1: Set App ID and Bundle Identifier

In order for the apps running on the same mobile device to be recognized as running on one device not on multiple devices by the Appeon License file, you will need to always input the same App ID for all apps and specify the same bundle identifier for the same app. For details, refer to Section 5.3.4, “Product Activation” in *PowerServer Configuration Guide for .NET* or in PowerServer Configuration Guide for J2EE.

You would need to modify the App ID in the keychainaccessgroups.plist file and the Bundle Identifier in the Xcode project settings to match with what you have specified when creating the App ID (in Task 1.4: Create an App ID).

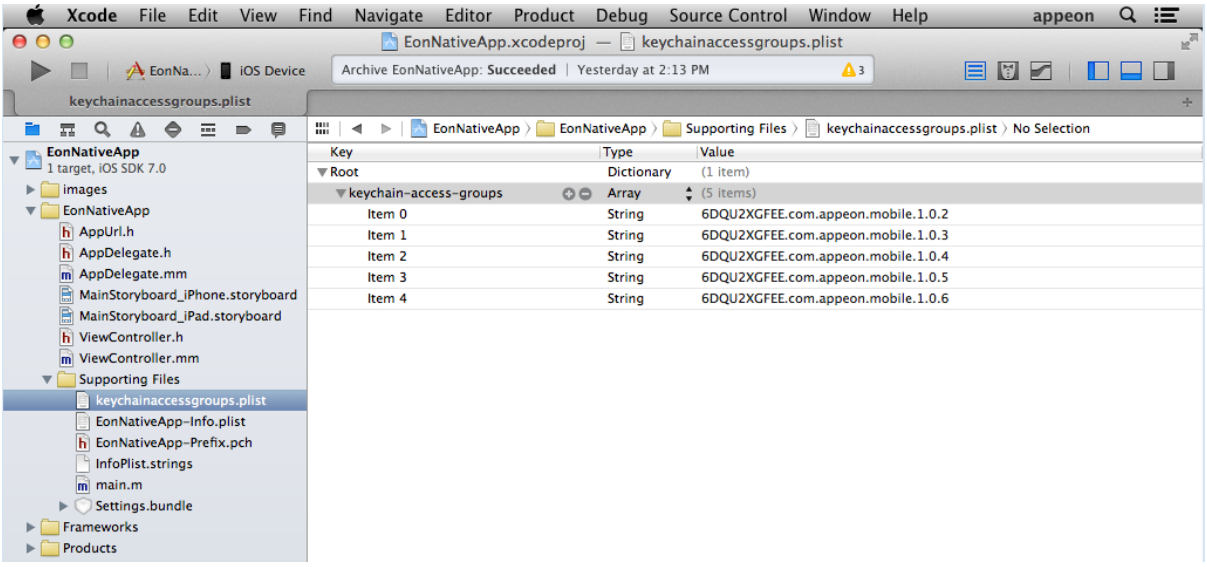
Follow steps below to specify the consistent App ID in the keychainaccessgroups.plist file and the Bundle Identifier in the project settings:

Step 1: Find the **keychainaccessgroups.plist** file by expanding **EonNativeApp** > **Supporting Files**, as shown in the following figure. The **keychainaccessgroups.plist** file is in the same directory as the **EonNativeApp.xcodeproj** file.

The key for the app in the **keychainaccessgroups.plist** file is made up of two parts: App ID + Bundle Identifier, in this format "AppID.BundleIdentifier", take 6DQU2XGFEE.com.appeon.mobile.1.0.2 for example, 6DQU2XGFEE is the App ID, com.appeon.mobile.1.0.2 is the Bundle Identifier.

If you specify a wildcard bundle identifier when creating the App ID, you will need to replace "*" with an explicit string, for example, com.abcexample.app1 or com.abcexample.1 etc.

**Figure 5.24: Keychain access group**




Step 2: Check that the App ID in the **keychainaccessgroups.plist** file is correct and is the same as the App ID in the provisioning profile.

If the App ID in the **keychainaccessgroups.plist** file is incorrect, you can input the correct App ID, then click the menu **Product** -> **Clean**, and then click **Product** -> **Archive** to create the IPA file again.

Step 3: Copy the Bundle Identifier from the **keychainaccessgroups.plist** file.

Step 4: Click the target project for the app, select the **General** tab, and then paste to the **Bundle Identifier** field. By default, the **Bundle Identifier** on this tab is different from the one in the **keychainaccessgroups.plist** file.

Also make sure the same **Bundle Identifier** is input when uploading the app to the App Store.

If you want to make change to the **Bundle Identifier**, please make sure the same **Bundle Identifier** is used consistently in all three areas: the **keychainaccessgroups.plist** file, the **General** tab, and the App Store upload process.

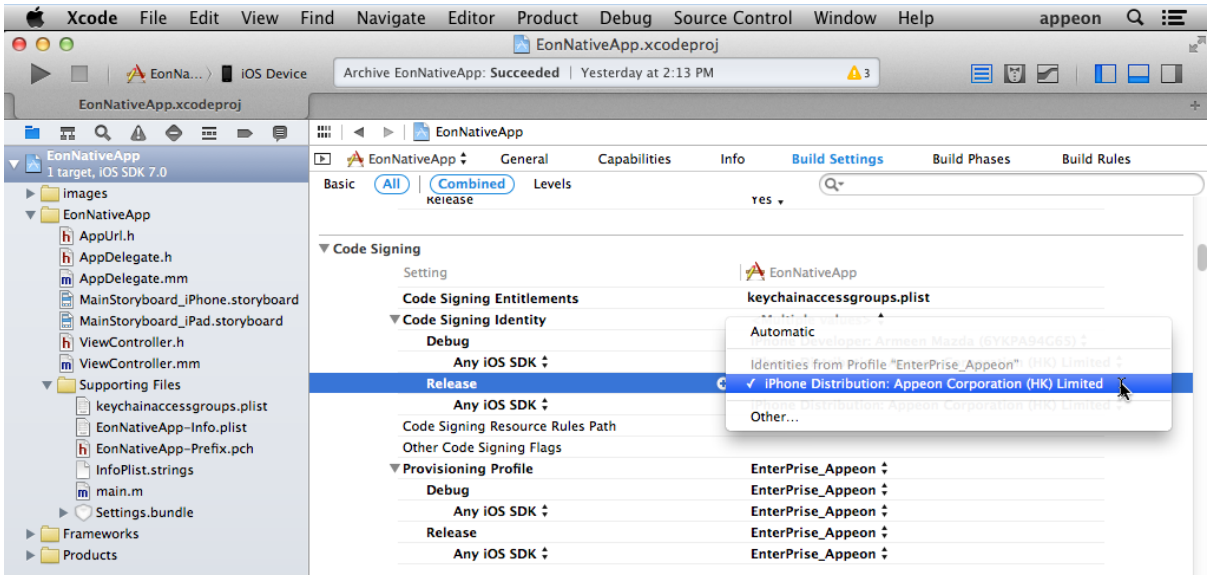
#### 5.2.3.2 Task 3.2: Code sign the app

You now code sign the app with the signing certificate contained in the corresponding provisioning profile.

To set the code signing identity to the certificate:

In the Xcode project navigator, select the project, select **Building Settings**, drag down the screen to find the **Code Signing** group, make sure **Code Signing Entitlements** is set to "keychainaccessgroups.plist", then click **Release** under **Code Signing Identity** and select the corresponding distribution certificate, and then click **Release** under **Provisioning Profile** and select the corresponding provisioning profile.

**Figure 5.25: Code signing**



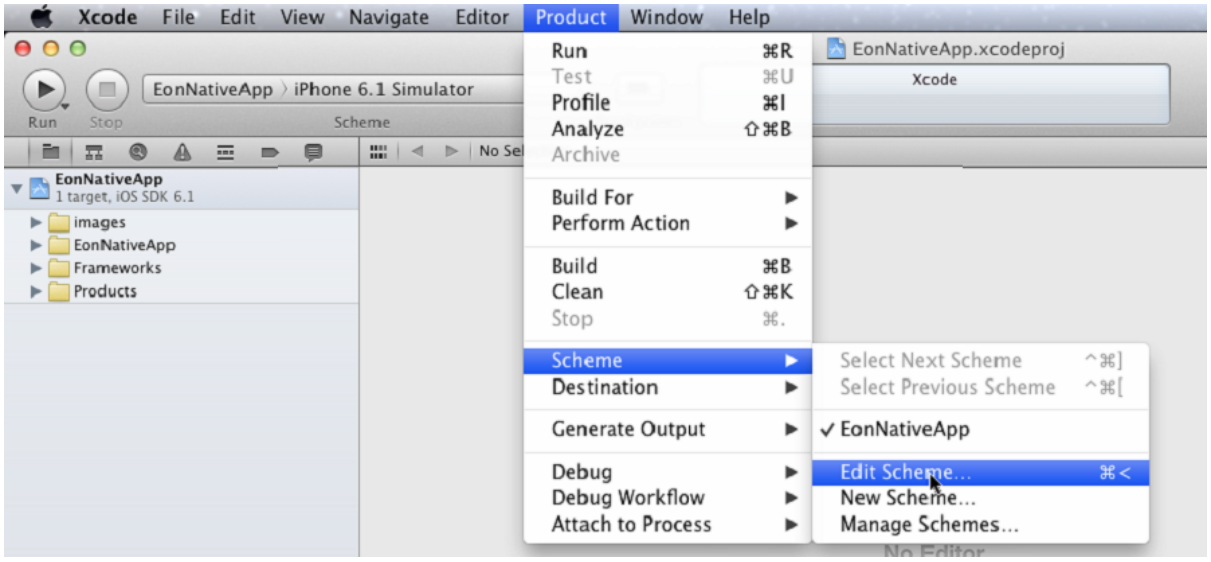

#### 5.2.3.3 Task 3.3: Review the archive scheme settings

You would need to double check the scheme settings to ensure you do not archive a debug version of the app.

To review the archive scheme:

Step 1: From the Xcode dropdown menu, choose **Product** > **Scheme** > **Edit Scheme** to open the scheme editor, as shown in the following figure.

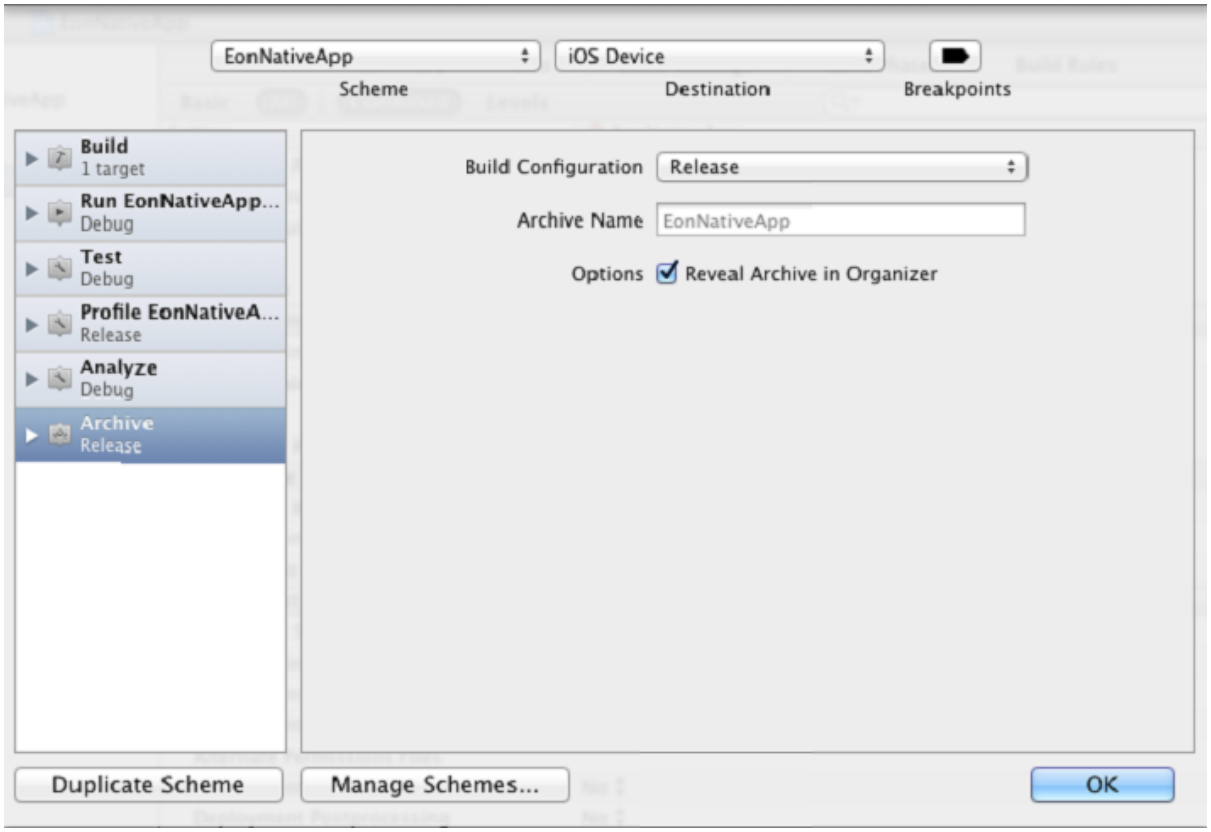
**Figure 5.26: Edit Scheme**




Step 2: In the scheme editor, do the following:

1. In the left column, select **Archive**;
2. Select the project from **Scheme**;
3. Select **iOS Device** from **Destination**;
4. Make sure **Release** (not **Debug**) is selected from **Build Configuration**;
5. Modify the archive name or use the default name in the **Archive Name** field.
6. Click **OK** to save the settings.

**Figure 5.27: Archive Release**




Step 3: (Optional) Click the project for the app, select the **General** tab, and then configure the Xcode project for distribution. All the information specified in the **PowerServer Toolkit Package** tool can be reconfigured here and you can also configure those that are not listed in the **PowerServer Toolkit Package** tool (only the required information are listed for configuration in the **PowerServer Toolkit Package** tool).

For details about how to configure the settings in Xcode, read Configuring Your Xcode Project for Distribution in the Apple document *App Distribution Guide*.

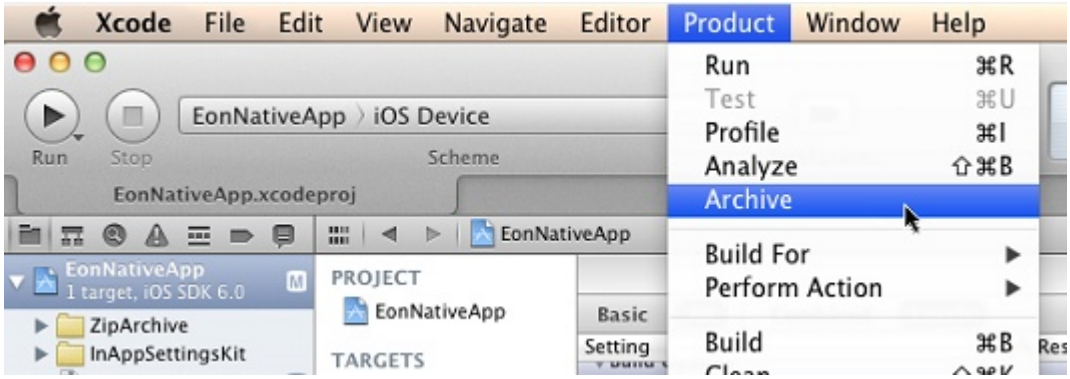
#### 5.2.3.4 Task 3.4: Create the archive

Now you are ready to create an app archive.

Step 1: In the Xcode project editor, select the project.

Step 2: From the Xcode dropdown menu, choose **Product** > **Archive**.

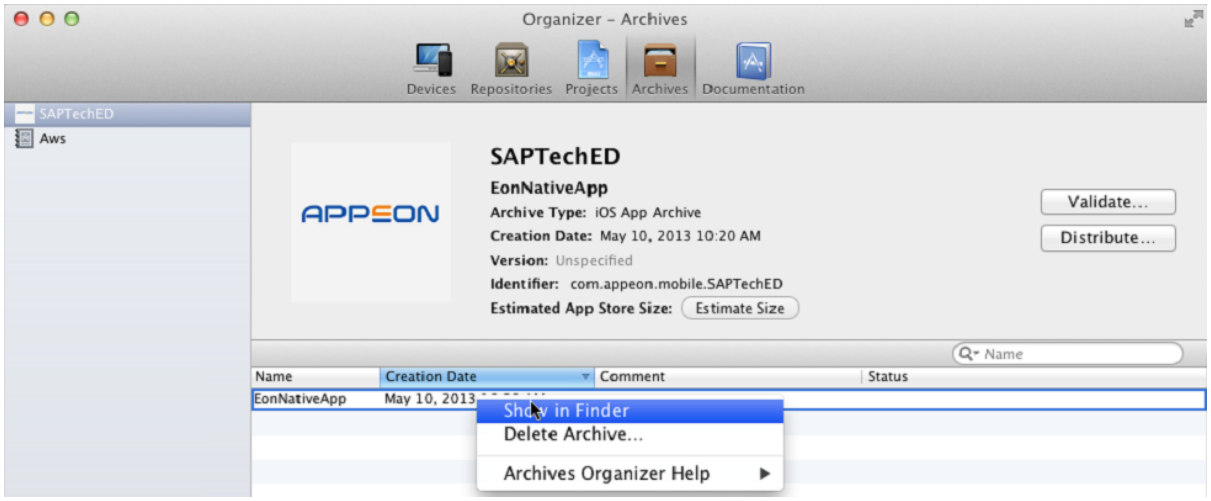
**Figure 5.28: Arhive**



If a prompt appears asking whether code sign can sign the app using a key in your keychain, click **Allow** or **Always Allow**.

After the project is built successfully, it will appear in the **Archives Organizer**, as shown below.

**Figure 5.29: Archive list**



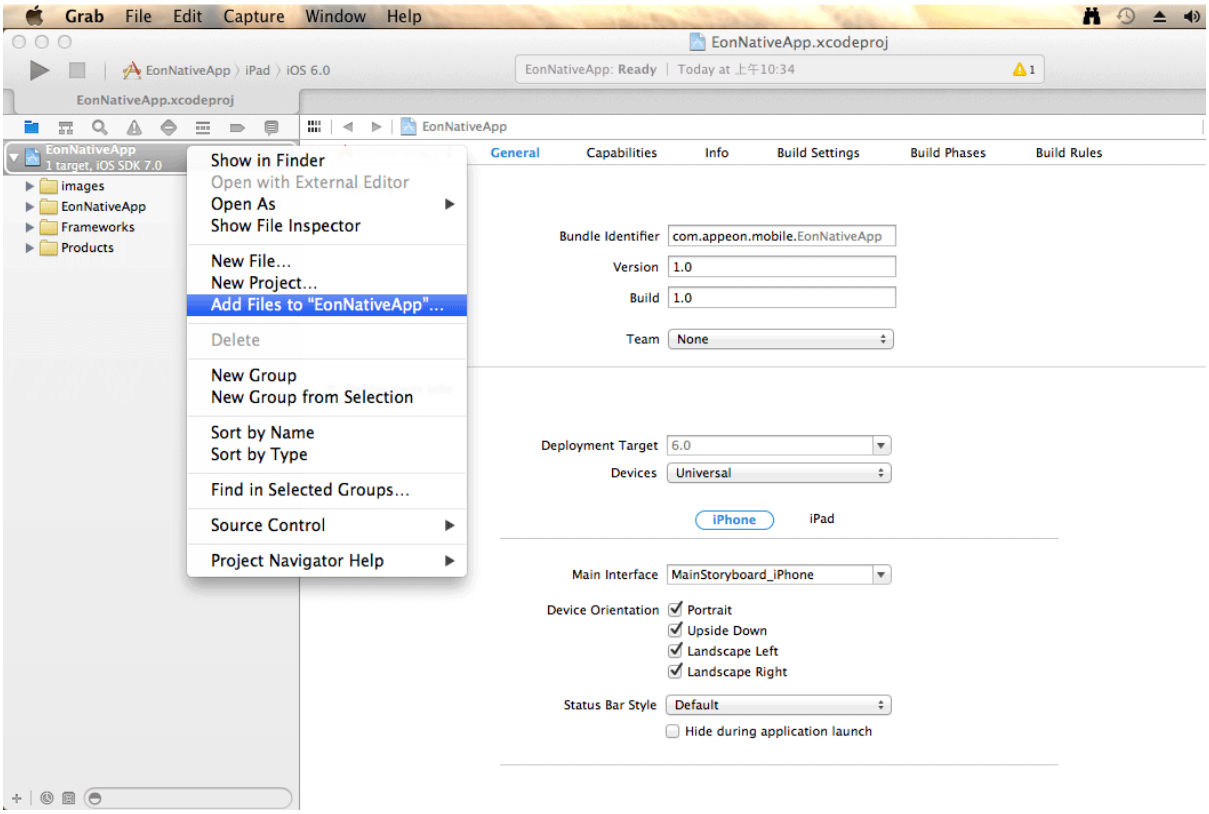

Now the app archive is ready. You can submit the archive to the Apple App Store or save the file for enterprise or Ad-Hoc distribution.

#### 5.2.3.5 (Optional) Import the certificate

MobiLink supports the encrypted transmission of the synchronous data stream by using HTTPS. If you want to use HTTPS for the encrypted transmission of the synchronous data stream, you should add the certificate file into the project. The following steps take **EonNativeApp** as an example.

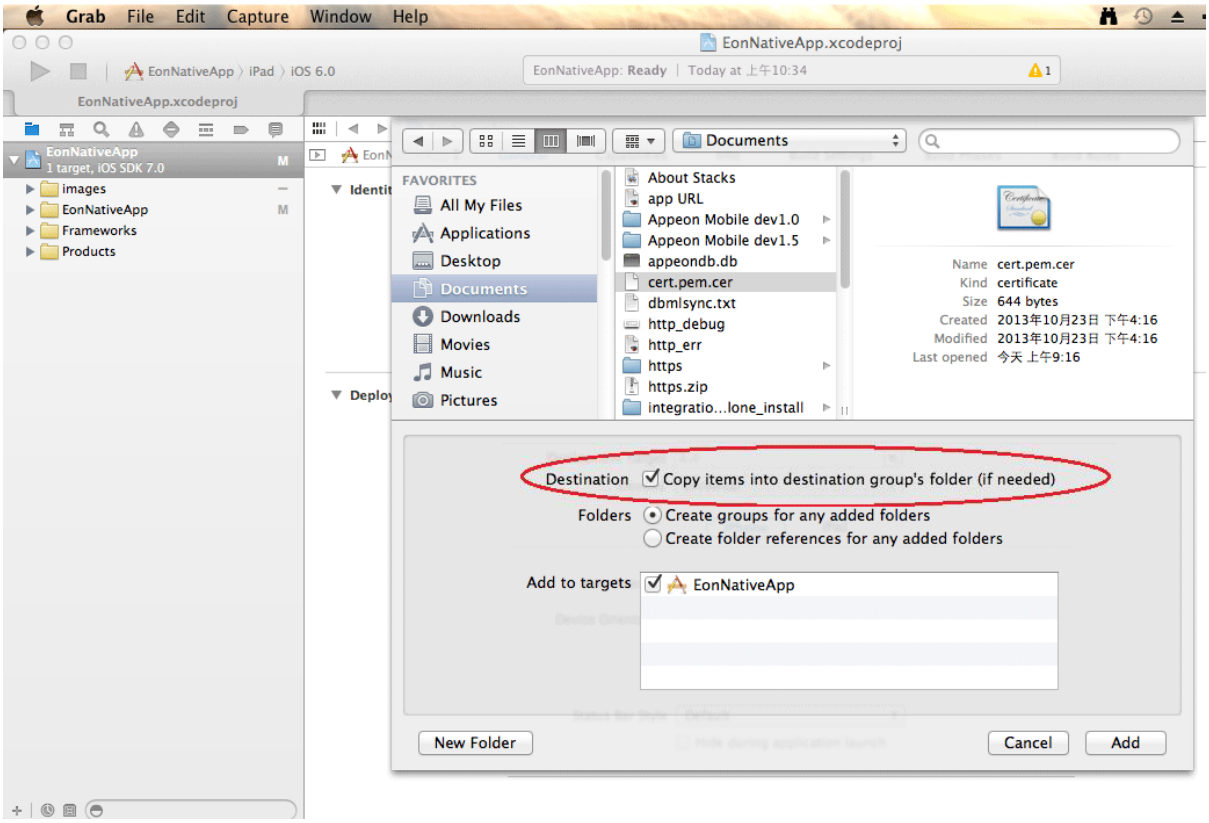
1. Right click **EonNativeApp** and select **Add Files to "EonNativeApp..."** to choose the certificate file, as shown in the following figure.

**Figure 5.30: Add files to "EonNativeApp"**



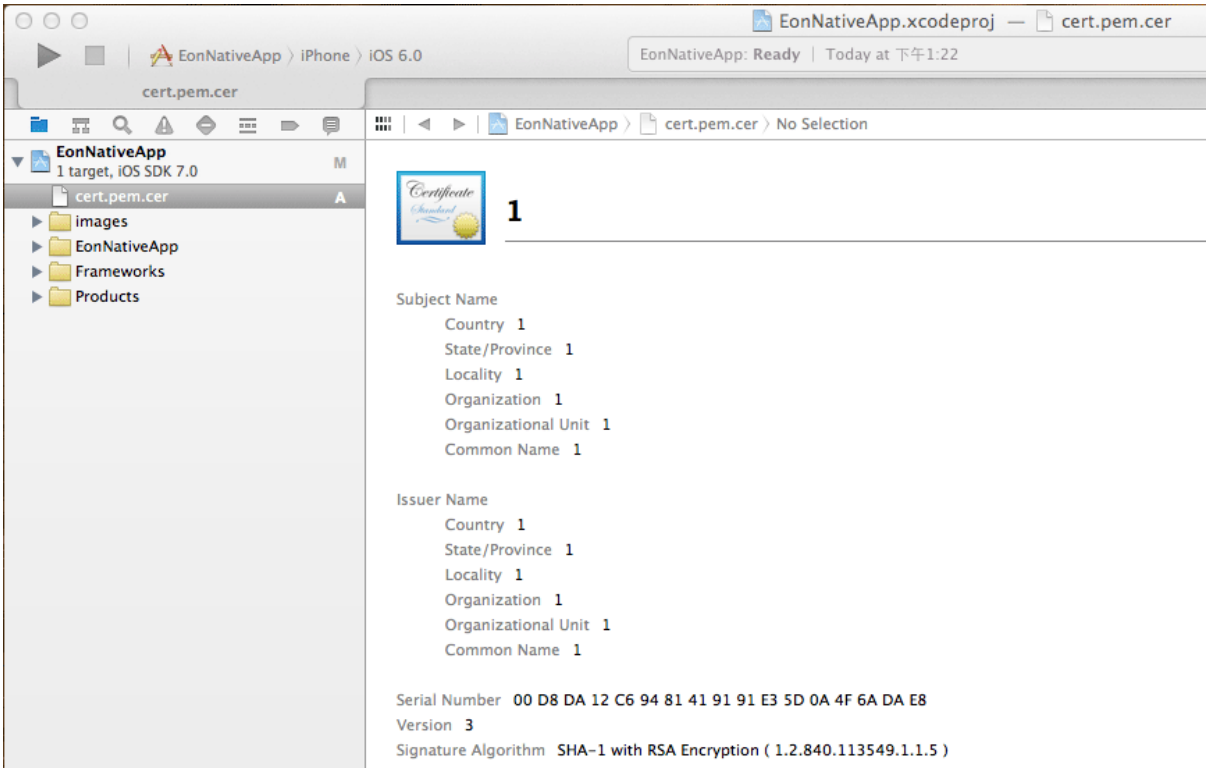

2. In this example, choose the cer.pem.cer file. Make sure to select the **Copy items into destination group's folder (if needed)** checkbox, as shown in the following figure. Click **Add**.

**Figure 5.31: Select the destination checkbox**




3. The certificate file is added successfully and is placed in the position parallel to other folders under the **EonNativeApp** directory, as shown in the following figure.

**Figure 5.32: The certificate file is added successfully.**

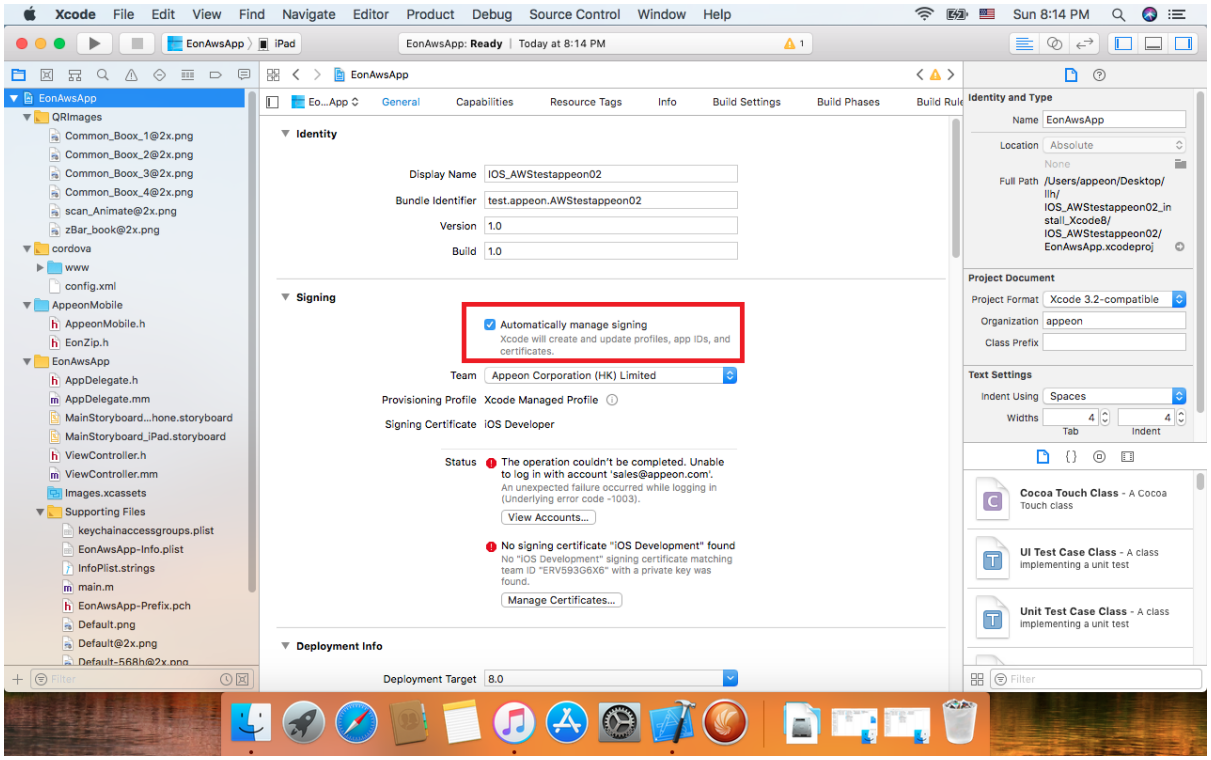


#### 5.2.3.6 Additional settings for Xcode 9

If you use Xcode 9 to create the app archive, make sure that you also configure the following Xcode 9 settings:

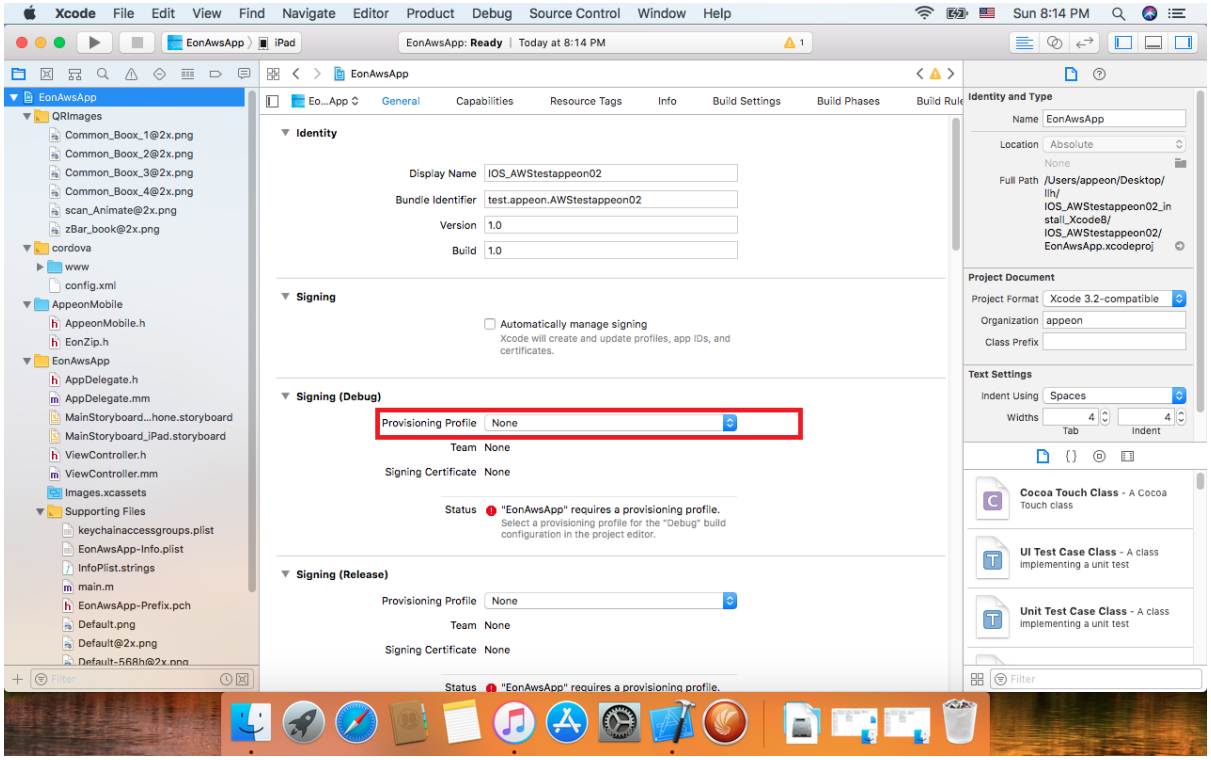
Setting 1: Clear the selection of the **Automatically manage signing** check box.

**Figure 5.33: Automatically manage signing**



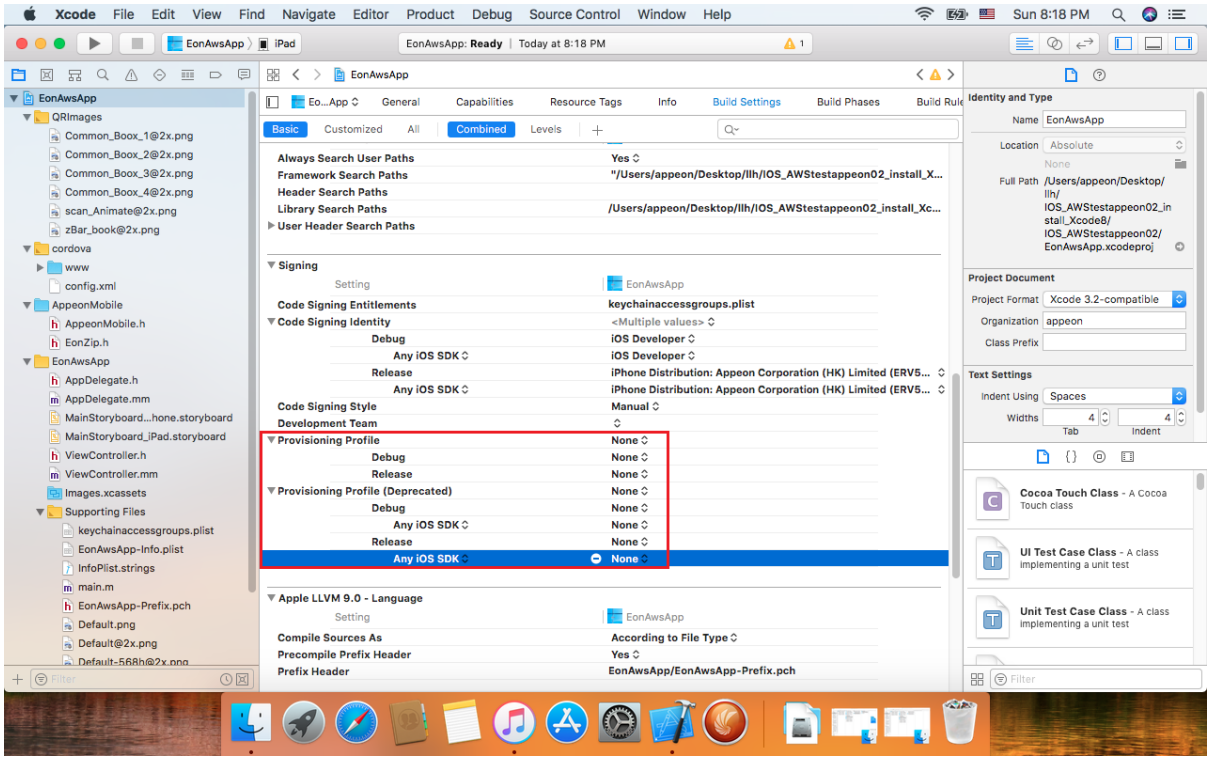

Setting 2: Select the corresponding provisioning profile from **Signing (Debug)** or **Signing (Release)** section.

**Figure 5.34: Signing (Debug)**



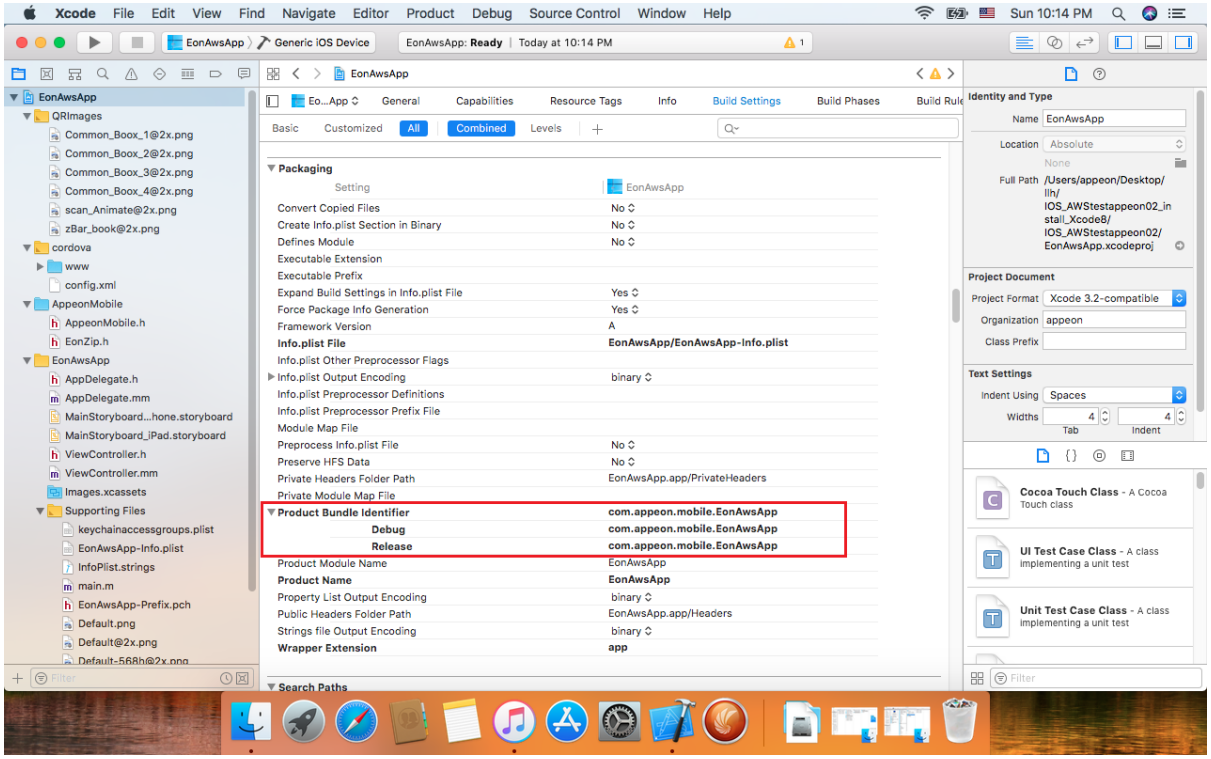

Setting 3: Select the corresponding provisioning profile under the **Provisioning Profile (Deprecated)** section as well.

**Figure 5.35: Provisioning Profile (Deprecated)**




Setting 4: If you are archiving Appeon Workspace, modify the bundle identifier under the **Product Bundle Identifier** section as well.

**Figure 5.36: Product Bundle Identifier**




## 5.2.4 Task 4: Distribute the app archive

Xcode allows you to distribute the app archive in the following methods:

- App Store distribution: submits the app for publication to the Apple App Store. Refer to Section 5.2.4.1, “Method 1: App Store distribution” for detailed instructions.
- In-House distribution: installs the app to an unlimited number of devices inside your company. Refer to Section 5.2.4.2, “Method 2: In-house or Ad-Hoc distribution” for detailed instructions.
- Ad Hoc distribution: installs the app on a limited number (up to 100) of registered devices. Refer to Section 5.2.4.2, “Method 2: In-house or Ad-Hoc distribution” for detailed instructions.

  The detailed instructions for Ad Hoc distribution and In-House distribution are the same, though their provisioning profiles are different.

### 5.2.4.1 Method 1: App Store distribution

To submit the app archive to the iOS App Store:

Step 1: In the Archives organizer, select the application archive you want to submit to iOS App Store, and click **Submit**.

Step 2: In the dialog that appears, choose a team from the pop-up menu and click **Choose**.

Step 3: Click **Submit** to transfer the IPA file to the iOS App Store.

For more instructions, refer to Submitting Your App to the Store in the Apple document *App Distribution Guide*.

#### 5.2.4.2 Method 2: In-house or Ad-Hoc distribution

To export the app archive and distribute it outside the iOS App Store:

Step 1: In the Archives organizer, select the application archive you want to export, and click **Export**.

Optionally, you can click **Validate** before clicking **Export**, to validate the archive and fix any problems before exporting it.

Step 2: In the dialog that appears, select the type of export and click **Next**.

- To distribute the app archive using ad hoc provisioning (where the app runs on registered test devices only), select **Save for Ad Hoc Deployment**.
- To distribute the iOS Developer Enterprise application, select **Save for Enterprise Deployment**.

Step 3: In the dialog that appears, choose a team from the pop-up menu and click **Choose**.

Step 4: Review the app, its entitlements, and provisioning profile.

Step 5: Click **Export**.

Step 6: In the dialog that appears, choose a location for the iOS App file (the .ipa file), and click **Export**.

To learn more about ad hoc deployment, refer to Distributing Your App Using Ad Hoc Provisioning in the Apple document *App Distribution Guide*.

To learn more about the iOS Enterprise application, refer to Distributing iOS Developer Enterprise Program Applications in the Apple document *App Distribution Guide*.

Now you can distribute the app to your users in the following four ways:

- Distribute the .ipa file and .plist file to your users, and users perform the installation using **iTunes**. See the section called “Install apps using iTunes” for detailed instructions.
- Distribute the .ipa file and .plist file to your users, and users perform the installation using **iPhone Configuration Utility** or **Apple Configurator** (Using **Apple Configurator** is left out in this tutorial). See the section called “Install apps using iPhone Configuration Utility” for detailed instructions.
- Post the .ipa file and .plist file on a secure Web server, and users perform the installation wirelessly. See the section called “Install apps wirelessly” for detailed instructions.
- Use your MDM server to instruct managed devices to install the app, if your MDM server supports it. This is left out in this tutorial.

##### 5.2.4.2.1 Install apps using iTunes

Step 1: Copy the .ipa file for the app to a machine (either Windows or Mac) with iTunes already installed.

Step 2: In **iTunes**, choose **File** > **Add to Library**, and then select the .ipa file.

Step 3: Connect a device to the computer, and then select it in the **Device** list in **iTunes**.

Step 4: Click the **Apps** tab, and then select the **Sync Apps** checkbox and the app in the list.

Step 5: Click **Apply** (or **Sync**).

If your user's computers are managed, you can deploy the files to their computers and ask them to sync their device. iTunes automatically installs the files found in iTunes Mobile Application and Provisioning Profiles folders.

#### 5.2.4.2.2 Install apps using iPhone Configuration Utility

Step 1: Copy the .ipa file for the app to a machine (either Windows or Mac) with **iPhone Configuration Utility** already installed.

Step 2: In **iPhone Configuration Utility**, choose **File** > **Add to Library**, and then select the .ipa file.

Step 3: Connect a device to the computer, and then select it in the **Device** list.

Step 4: Click the **Applications** tab, and then select the .ipa file.

Step 5: Click **Install**.

#### 5.2.4.2.3 Install apps wirelessly

Before installing the app, you will need to first post the app to a Web server.

Step 1: Construct and configure the server.

1. Install IIS on the server.
2. Copy the app (.ipa) file, the manifest (.plist) file, and the icons to a location on the IIS website that is accessible to the users.
3. Create a page that links to the manifest file.

   Here is a sample link:

```
<a href="itms-services://?action=download-manifest&url=http://example.com/?
manifest.plist">Install App</a>
```

Step 2: Set the MIME type of the server, so the server can correctly transfer the manifest file and the application.

- For Mac OS X Server, use the Server Admin to add the following MIME types to the Web service's **MIME Types** settings:

  application/octet-stream ipa

  text/xml plist

- For IIS, use the IIS Manager to add the following MIME types on the server's **Properties** page:

  .ipa application/octet-stream

  .plist text/xml

Now your app is ready for installation.

Step 3: Distribute the URL for downloading the manifest file by SMS or email to the users.

Step 4: The user clicks the URL to download the manifest file from the website to the iOS device, and the manifest file instructs the device to download and install the apps referenced in the manifest file.

## 5.3 Package & Distribute Android Apps

Compared to iOS apps, the steps for packaging and distributing the Android apps are a lot simpler. Once you package the app files using the PowerServer Toolkit, you will get an Android application package (APK) that can readily be distributed using any approach, including publishing in the Android app marketplace such as Google Play, or serving the app from a Web site or emailing the app directly to users.

Therefore, you need to go through the following two tasks mainly.

1. Task 1: Generate the Android application package (APK).
2. Task 2: Distribute the Android APK.

### 5.3.1 Task 1: Generate the Android application package (APK)

On the PowerServer Toolkit machine, use the **Package** tool in the PowerServer Toolkit to package the app files into an Android APK file.

You can select **Package a Stand-alone Mobile Project** or **Customize and Package Appeon Workspace** to package the app files into an Android APK file. During the package process, you can press F1 to get help on the fields. See Section 10.3, “Packaging a stand-alone mobile project” in *PowerServer Toolkit User Guide* or Section 10.4, “Customizing and packaging Appeon Workspace” in *PowerServer Toolkit User Guide* for detailed instructions.

After the package process is complete, you will find an APK file generated under the specified destination folder. Get the APK file and distribute the app using any approach that meets your needs.

### 5.3.2 Task 2: Distribute the Android APK

This section only highlights three distribution approaches. For more details about these three approaches, refer to http://developer.android.com/distribute/open.html.

- Distributing through an App Marketplace.

  For detailed instructions on how to publish on Google Play -- the premier marketplace for Android apps, refer to http://developer.android.com/distribute/googleplay/publish/index.html.

- Distributing your application through email
- Distributing through a web site

# 6 Tutorial 5: Configure PowerServer Cluster

(**Note**: This chapter is applicable for both PowerServer Web and PowerServer Mobile, although it's listed in the PowerServer Mobile Tutorials document.)

Appeon provides its own tools and plug-ins to configure a PowerServer cluster and implement the load balancing and failover functionalities. A PowerServer cluster is essentially a group of application servers, each with PowerServer installed.

Following are high-level steps for configuring a PowerServer cluster. For detailed instructions, you may need to go through the documents and sections as specified below.

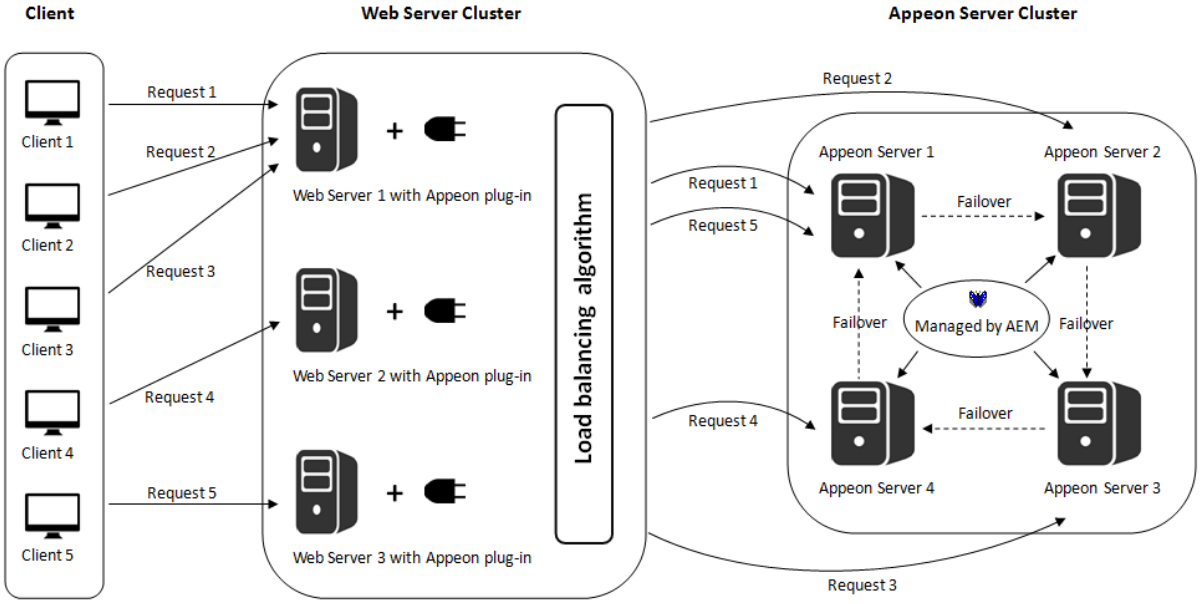
1. Task 1: Get to understand the architecture of a PowerServer cluster.
2. Task 2: Install PowerServer to multiple application servers.
3. Task 3: Create the PowerServer cluster in AEM.
4. Task 4: Configure the Web server for the PowerServer cluster.
5. Task 5: Install an Appeon application to the PowerServer cluster and Web server(s).

## 6.1 Task 1: Understand the cluster architecture

The following graphic illustrates the architecture of the client, the Web server cluster, and the PowerServer cluster.

- The Web server can be a cluster of Web servers or a single Web server, depending on the demands. Configuring the Web server cluster is not an Appeon task, therefore, it is not mentioned in any Appeon documentation; to explore this topic, you could refer to the user documents provided by the Web server vendor. Generally, a single Web server is sufficient to support the PowerServer cluster.
- Appeon cluster plug-in is installed to the Web server (either a single Web server or a cluster of Web servers), for distributing the user requests to the PowerServer according to the specified load balancing settings and the failover settings.
- If there is only one Web server, it can be on an independent machine separated from the PowerServer, or it can be on the same machine where PowerServer resides. If it is on the same machine with PowerServer, make sure the PowerServer and the Web server are set up on different Web sites.
- There are two load balancing algorithms: **random** and **sequence**. The following graphic uses **sequence** which distributes the user requests to the PowerServer in sequential order.

**Figure 6.1: Cluster architecture**




## 6.2 Task 2: Install PowerServer to multiple application servers

Step 1: Set up a group of application servers.

The application server must be of the same type and of the same version. Appeon supports the following application server types: .NET IIS, JBoss, WebLogic, WebSphere, JEUS, and NetWeaver. For more details, refer to the **PowerServer requirements** section in Installation Guide for .NET.

Step 2: Install PowerServer to each application server by following the installation instructions in the **PowerServer installation** section in Installation Guide for .NET. In the **Select Components** page, make sure you select to install the appropriate edition of PowerServer, for example, select **PowerServer for .NET** if the application server is .NET/IIS.

## 6.3 Task 3: Create a PowerServer cluster in AEM

Once you have installed PowerServer to several application servers, you can use the AEM **Cluster** tool to create a PowerServer cluster and then configure the load balancing and failover functionalities for the cluster. The following are the detailed instructions.

Step 1: Log into AEM. Each PowerServer has a AEM. You only need to log in to one of them for the cluster, make configurations there, and then synchronize the settings to the other AEMs in the cluster.

Step 2: In the **Server | Resource | Cluster** tool, add the PowerServer one by one. It is **important** that the first PowerServer you add is the one hosting the AEM you are currently logged into.

1. Click the **Add PowerServer** button under the **Cluster Server List** table. The **Add New PowerServer** page opens.

2. Verify that the PowerServer to be configured is running and provide the required information (IP address, port, AEM user name, and password). For example:

- IP address: 161.0.0.1

  Use the IP address or machine name of the PowerServer when adding a PowerServer. Do not use "localhost" or "127.0.0.1". The IP address or machine name and port number must match the settings of HTTP listeners in the application server.

- Port: 80
- AEM User Name: admin
- AEM Password: admin

3. Click the **Save and Add** button to add the other PowerServer. The program will automatically test the connection and add the PowerServer if the test is successful.

   Adding a PowerServer will succeed only if:

   - The PowerServer is new to AEM.
   - The information provided is correct.
   - The PowerServer is running.
   - The first PowerServer that you add is the one hosting AEM you are currently logged into.

The PowerServer listed in the **Cluster Server List** group will work as a cluster in supporting the requests from their associated Web server.

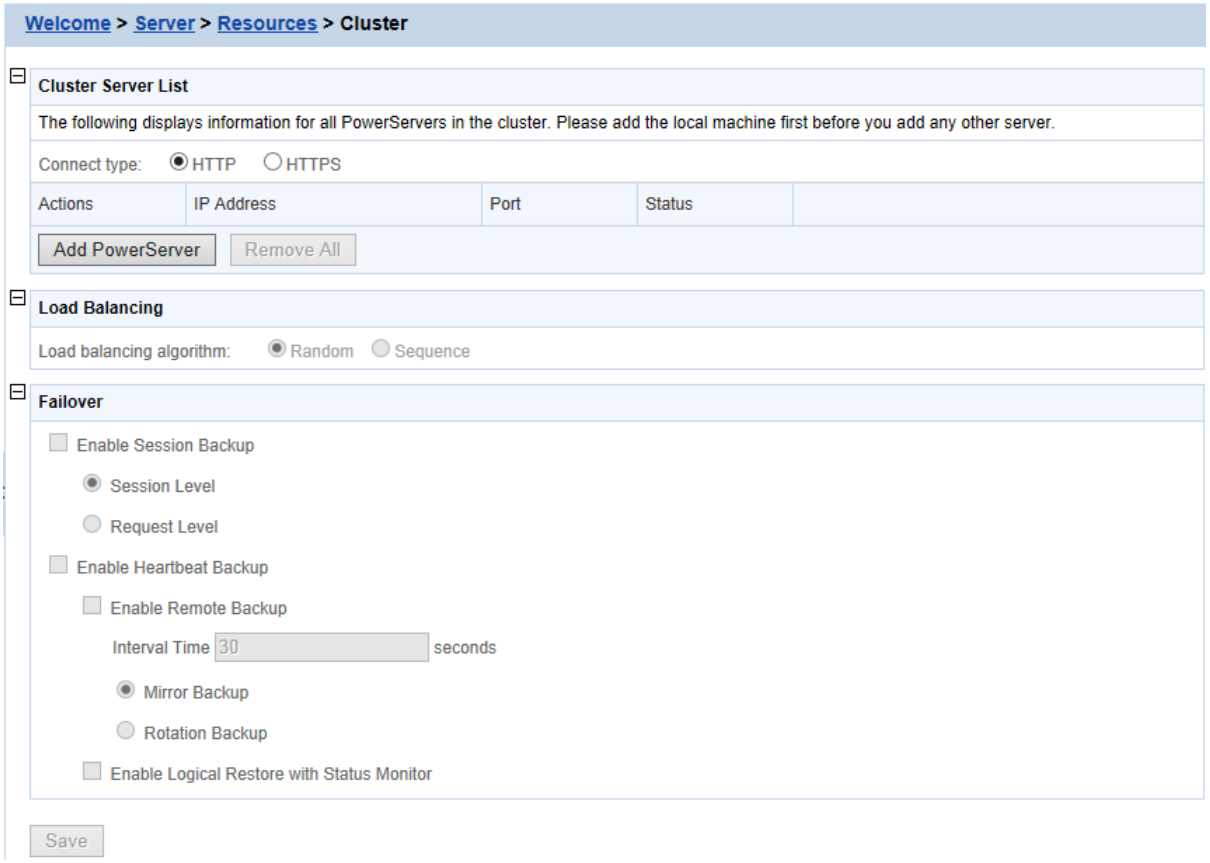
Step 3: Configure the load balancing and the fail-over settings by following instructions in Section 5.3.3.1, “Cluster” in *PowerServer Configuration Guide for .NET* or PowerServer Configuration Guide for J2EE.

Step 4: Check the status of PowerServer and verify that all servers are running.

Step 5: Click **Save** to synchronize AEM settings to the other servers in the cluster.

Once a PowerServer is added to the cluster server list, you can use the AEM which maintains the list to manage all servers in the cluster. Whenever you change the settings of this AEM and click **Save**, the settings will be saved to the other servers in the cluster. Therefore, you do not need to repeat the configuration in each AEM. However, not all of the AEM settings will be synchronized, because some settings are not necessary to be the same for all servers.

**Figure 6.2: Cluster**

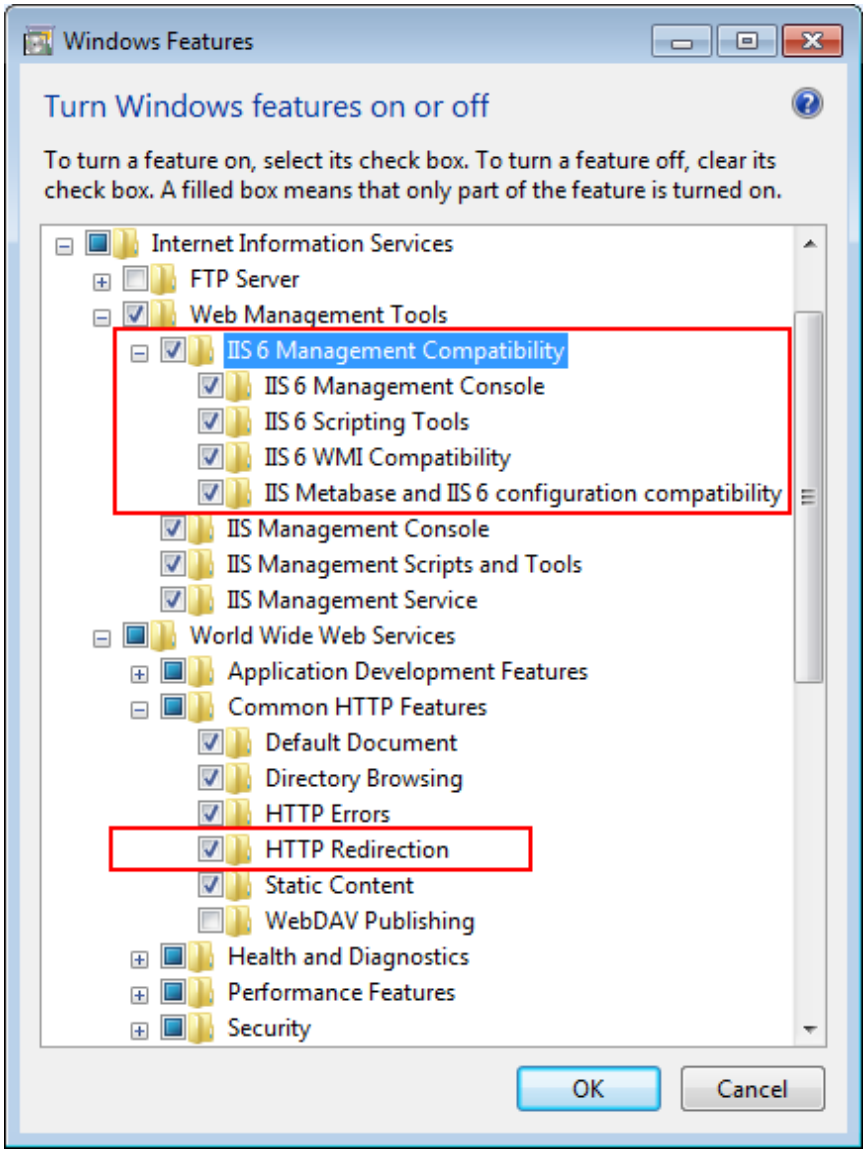



## 6.4 Task 4: Configure the Web server for the PowerServer cluster

### 6.4.1 Task 4.1: Important notes for installing IIS

If the Web server is the IIS server, you should make sure the **IIS 6 Management Compatibility** feature (including its sub-features) and the **HTTP Redirection** feature are selected and installed, otherwise, the application URL and AEM URL cannot be redirected successfully.

**Figure 6.3: Required IIS features**




### 6.4.2 Task 4.2: Install the PowerServer Web Component on Web server

Step 1: Set up one or more separate Web site(s) as Web server(s).

If the Web server is on the same machine with PowerServer, make sure the PowerServer and the Web server are set up on different Web sites that use different port number and root directory.

For detailed system requirements of the Web server, refer to the **(Optional) Web server requirements** section in Installation Guide for .NET.

Step 2: Install **PowerServer Web Component** to the Web root (or document root) of each Web server. For example, the default Web root for IIS is C:\inetpub\wwwroot, the default document root for Apache 2.0 is C:\Program Files\Apache Group\Apache2\htdocs.

For detailed instructions, please follow the installation instructions in **PowerServer Web Component installation** section in Installation Guide for .NET.

## 6.4.3 Task 4.3: Configure the Web server with the Appeon cluster plug-in

### 6.4.3.1 Type 1: IIS Web server

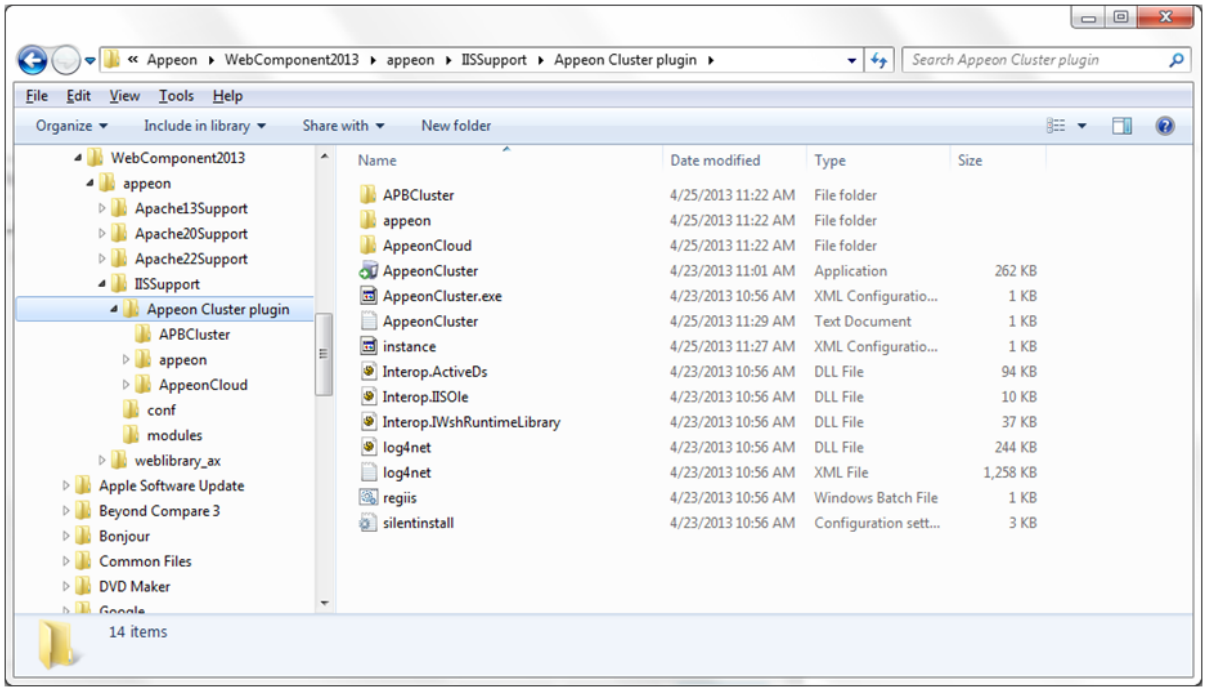
#### 6.4.3.1.1 Method 1: Automatic configuration

**Installing and configuring Appeon cluster plug-in**

Step 1: Get the Appeon cluster plug-in installation package.

After you install **PowerServer Web Component**, you can find the **Appeon Cluster plugin** folder under the **WebComponent2017\appeon\IISSupport** directory, and this folder is the Appeon cluster plug-in installation package, as shown in the following figure. You can copy this installation package to any Web server associated with the PowerServer cluster.

**Figure 6.4: Appeon Cluster plugin**



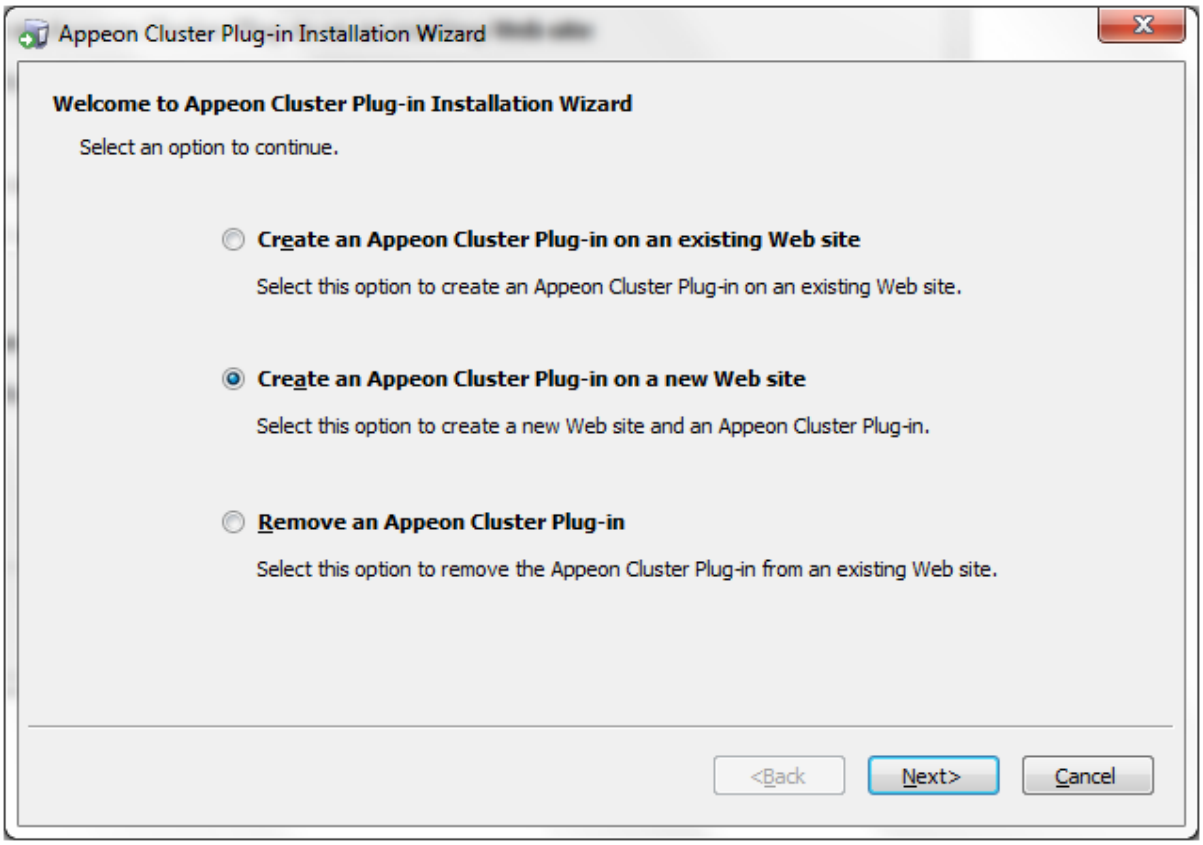

Step 2: Double-click **AppeonCluster.exe** under the **Appeon Cluster plugin** folder to install the Appeon cluster plug-in to the Web server.

The **Appeon Cluster Plug-in Installation Wizard** is displayed.

Step 3: Select **Create an Appeon Cluster Plug-in on a new Web site**, and click **Next**.

If you want to install to an existing Web site, select **Create an Appeon Cluster Plug-in on an existing Web site**, and click **Next**.

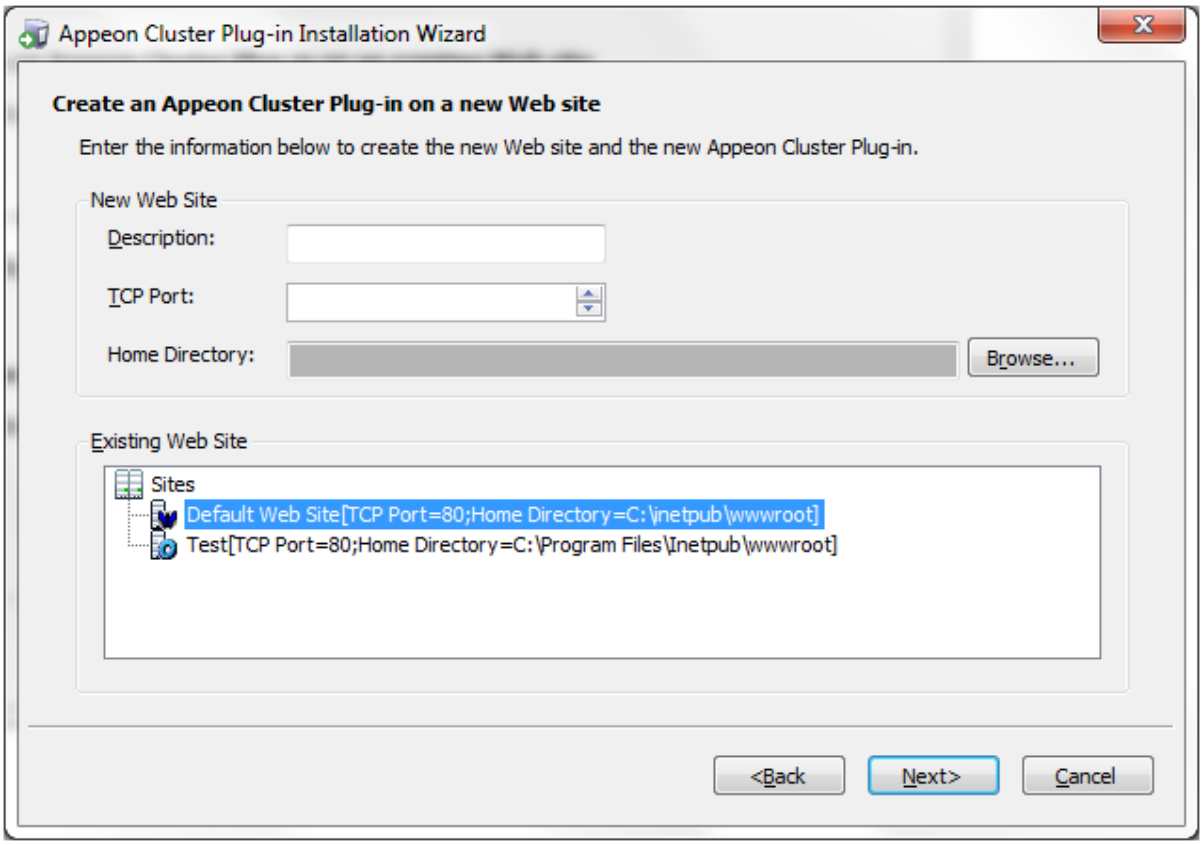
**Figure 6.5: Create plugin on a Web site**




Step 4: To create the Appeon cluster plug-in on a new Web site, configure **Description**, **Port** and **Home Directory**, then click **Next**.

The new Web site will be created.

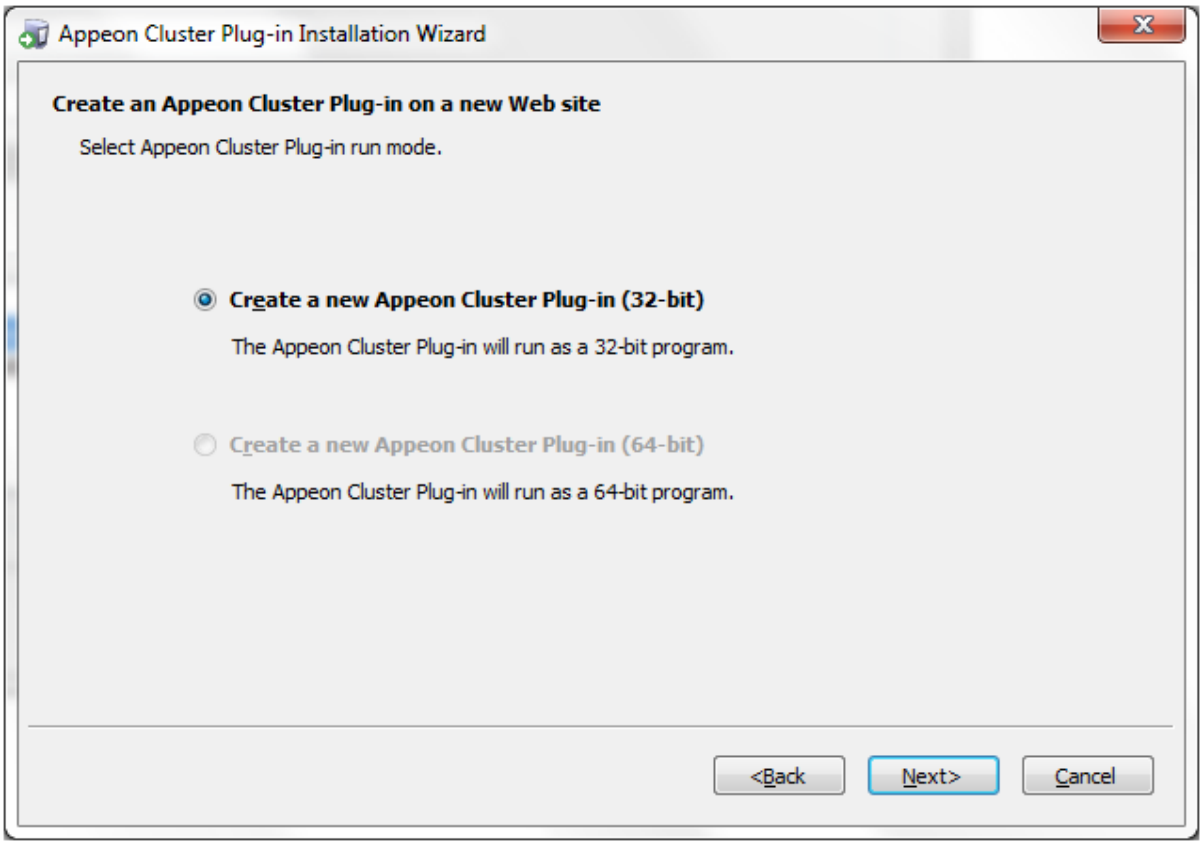
**Figure 6.6: Configure the Web site**




Step 5: Select to install a 32-bit or 64-bit Appeon cluster plug-in, and click **Next**. The program will detect if the server is on 32-bit OS or 64-bit OS and disable the inappropriate option.

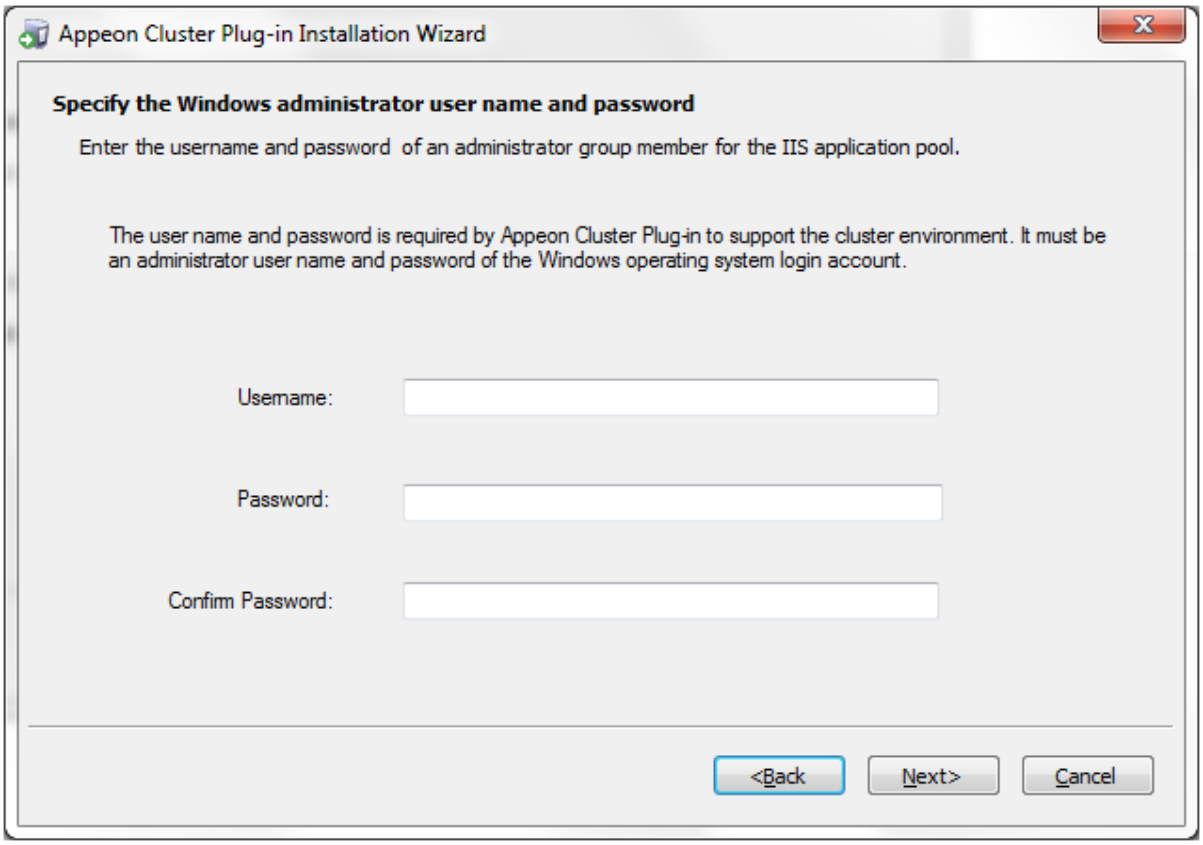
32-bit plug-in works with 32-bit server and 64-bit server, and 64-bit plug-in works with 64-bit server only.

**Figure 6.7: Select run mode**



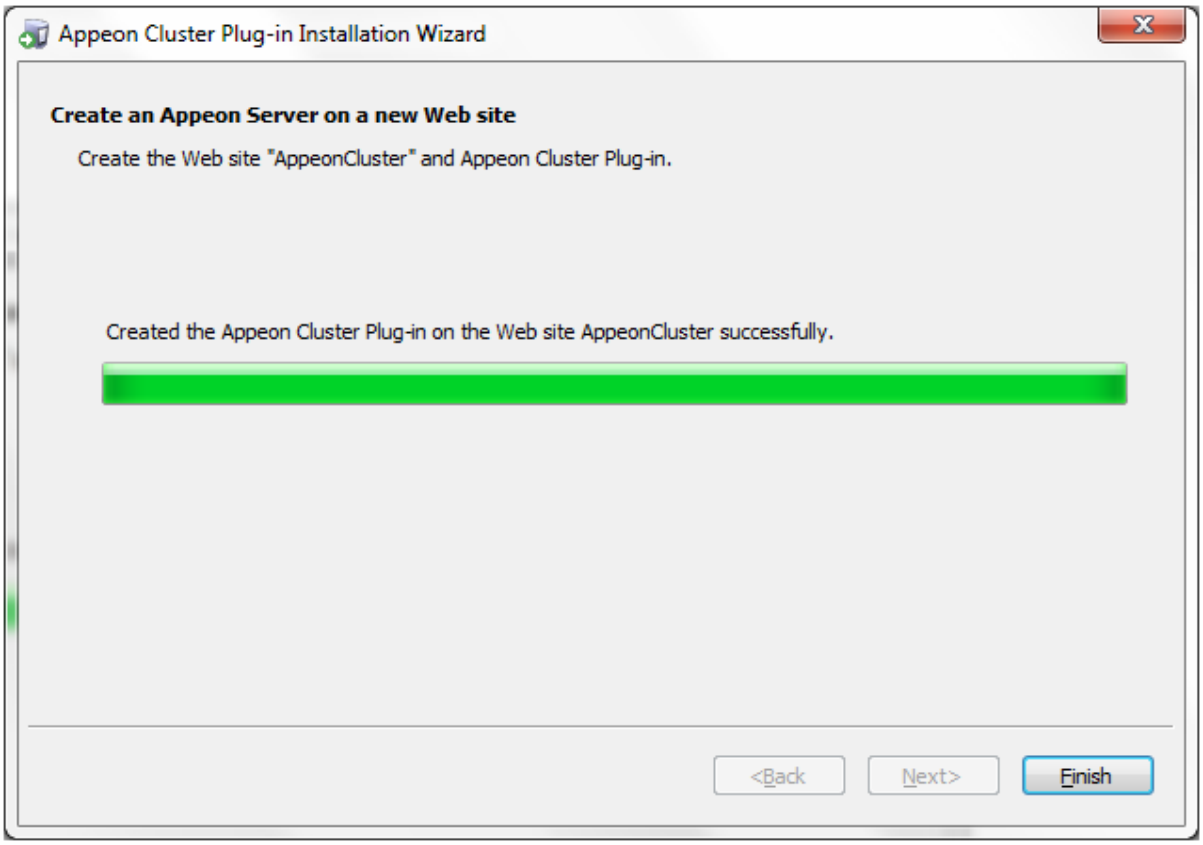

Step 6: Specify the Windows administrator user name and password, and click **Next**. Make sure to input the correct user name and password, otherwise you may not be able to access the **Appeon Cluster Manager** in the Web browser.

**Figure 6.8: Windows administrator username and password**




Step 7: When the plug-in is created successfully, click **Finish** to exit the **Appeon Cluster Plug-in Installation Wizard**.

**Figure 6.9: Create the plug-in**




Step 8: Restart IIS to make the new settings effective.

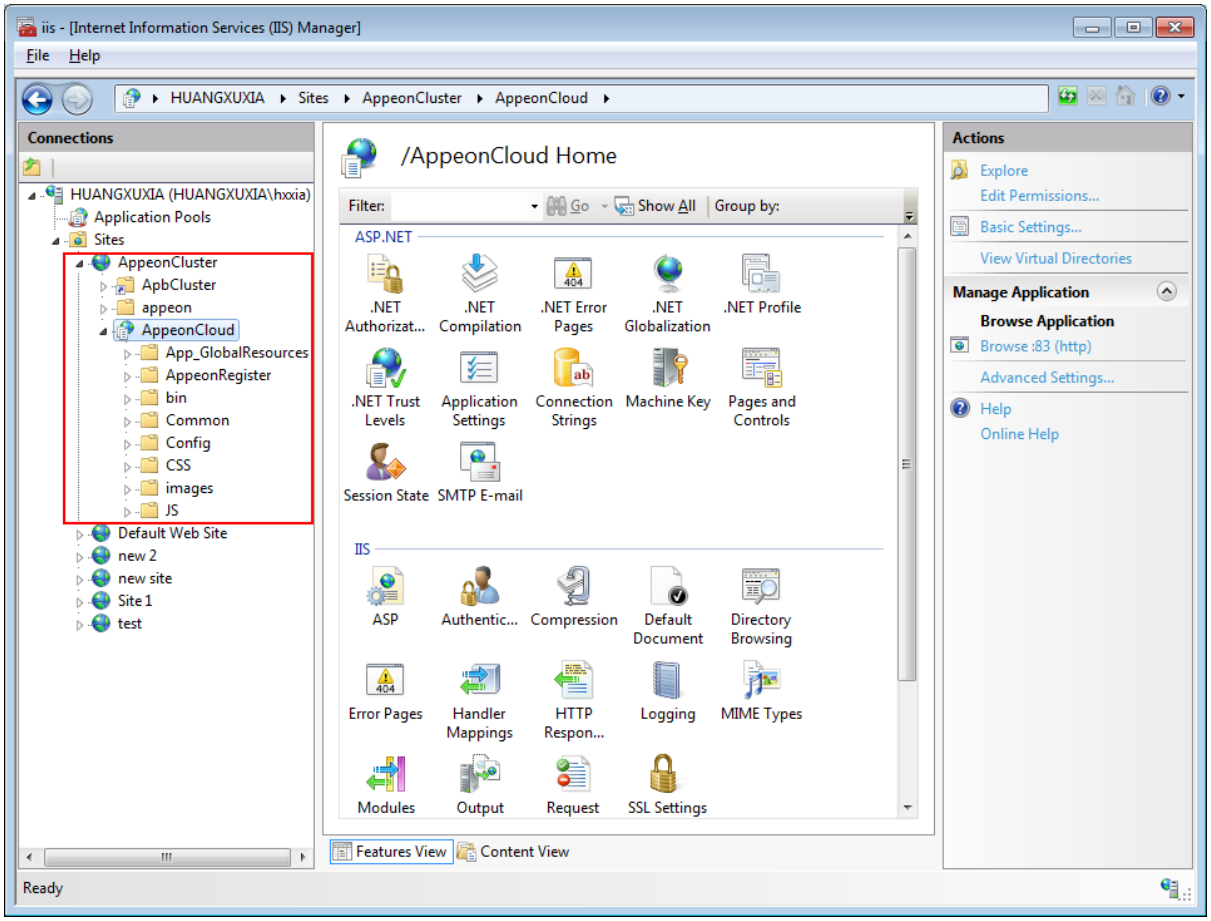
#### Configuring redirections to PowerServer

After successfully installing and configuring the Appeon cluster plug-in, you can find **AppeonCloud** Web console (though it is named "Cloud", it also applies to non-cloud platform) under the specified IIS Web site, as shown in the figure below. You can configure here to which PowerServer(s) and in what order the user requests should be redirected. Detailed instructions are provided below.

Another important advantage that the cluster provides is **failover**. To enable the **failover** functionality, you will need to go to the AEM, add PowerServer to the cluster and configure the failover settings there. For details, refer to Section 5.3.3.1, “Cluster” in *PowerServer Configuration Guide for .NET* or PowerServer Configuration Guide for J2EE.

The Appeon cluster currently supports three kinds of platforms including **Local** (non-cloud platform), **Amazon Web Services** and **Windows Azure**. The **AppeonCloud** console will display different settings according to the specific platform.

**Figure 6.10: IIS Manager**




Step 1: Run the **AppeonCloud** console (browse the console in the right **Actions** pane in the IIS manager). The **Appeon Cluster Manager** displays.

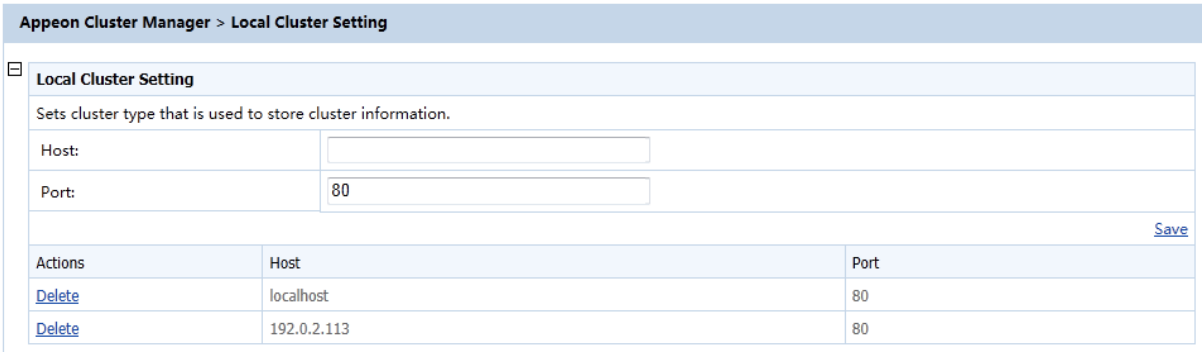
Step 2: Input the user name and password (both "admin") to log into the **Appeon Cluster Manager**.

Step 3: Click **Cluster Settings**. The following settings will be display for the non-cloud platform.

Add the PowerServer to the list one by one. The Web server will redirect the user requests to the PowerServer in the list.

- Specify the host name (or IP address) and port number of PowerServer.
- Click **Save**. PowerServer will be added to the cluster.

**Figure 6.11: Local cluster settings**




Step 4: Configure the other settings of the cluster if necessary, such as load balancing algorithm, timeout, and interval.

**Figure 6.12: Cluster settings**

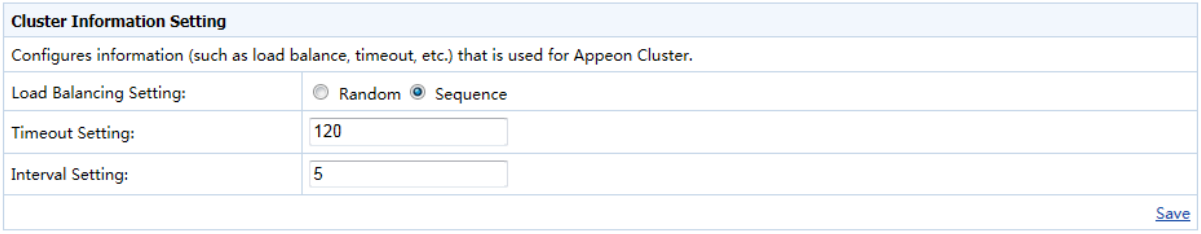



**Table 6.1: More cluster settings**

| Settings | Description |
|---|---|
| Load Balancing | Specify the load balancing algorithm which determined how requests will be distributed among the servers in the cluster. **Random** indicates that the plug-in distributes requests across PowerServer in random order, regardless of the status of PowerServer; **Sequence** indicates that the plug-in distributes requests to PowerServer in an allocated order. |
| Timeout | Specify the timeout for distributing a request, and the default value is 30 seconds. |
| Interval | Specify the interval (in seconds) for the plug-in to refresh the PowerServer list. |

#### 6.4.3.1.2 Method 2: Manual configuration

##### Configuring IIS 7.x with .NET server

Choose an existing Web site, or create a new site. The **Default Web Site** will be used as examples in the following guide.

1. Installing PowerServer Web component to the Web root of the IIS server.

   Make sure you have installed PowerServer Web Component to the Web root of the IIS server. To verify it, go to the IIS Web root (by default C:\inetpub\wwwroot) and check

if there is an **appeon** folder which at least contains the **IISSupport** and **weblibrary_ax** sub-folders. If any folder is missing, go to the default installation directory of PowerServer Web Component (C:\Program Files\Appeon\WebComponent2017) and manually copy the entire **appeon** folder to the IIS Web root.

2. Creating a virtual directory.

   Step 1: Open **Control Panel** | **Administrative Tools** | **Internet Information Service (IIS) Manager**.

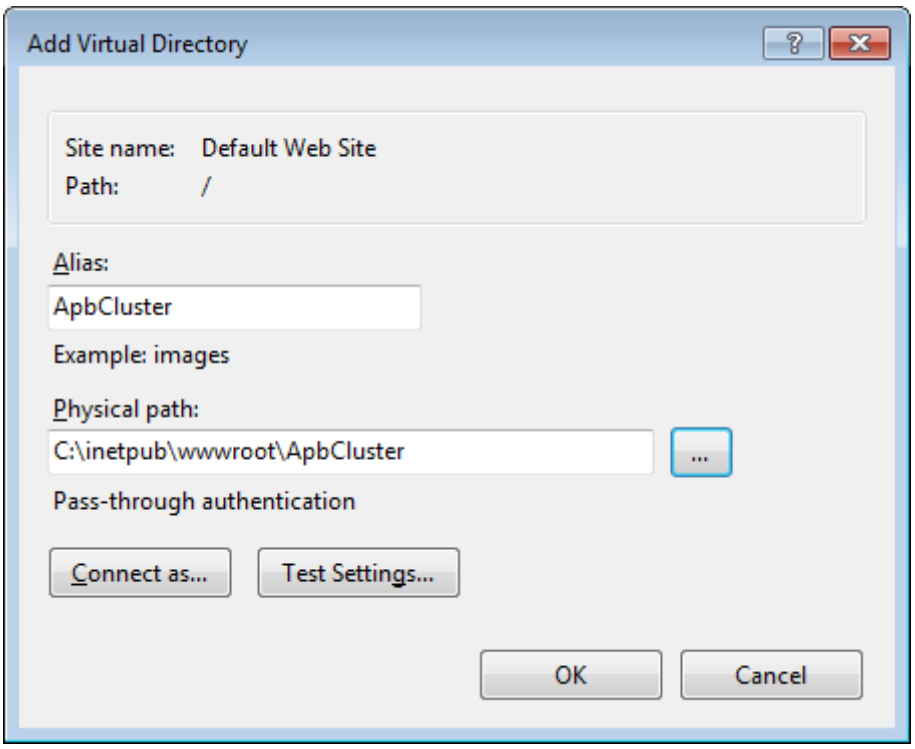
   Step 2: Right click the **Default Web Site** and select **Add Virtual Directory**.

   Step 3: Input the alias name, for example, "**ApbCluster**".

   Step 4: Choose a local path in the **Physical path** box; for example, "**C:\inetpub\wwwroot\ApbCluster**".

   Step 5: Click **OK**.

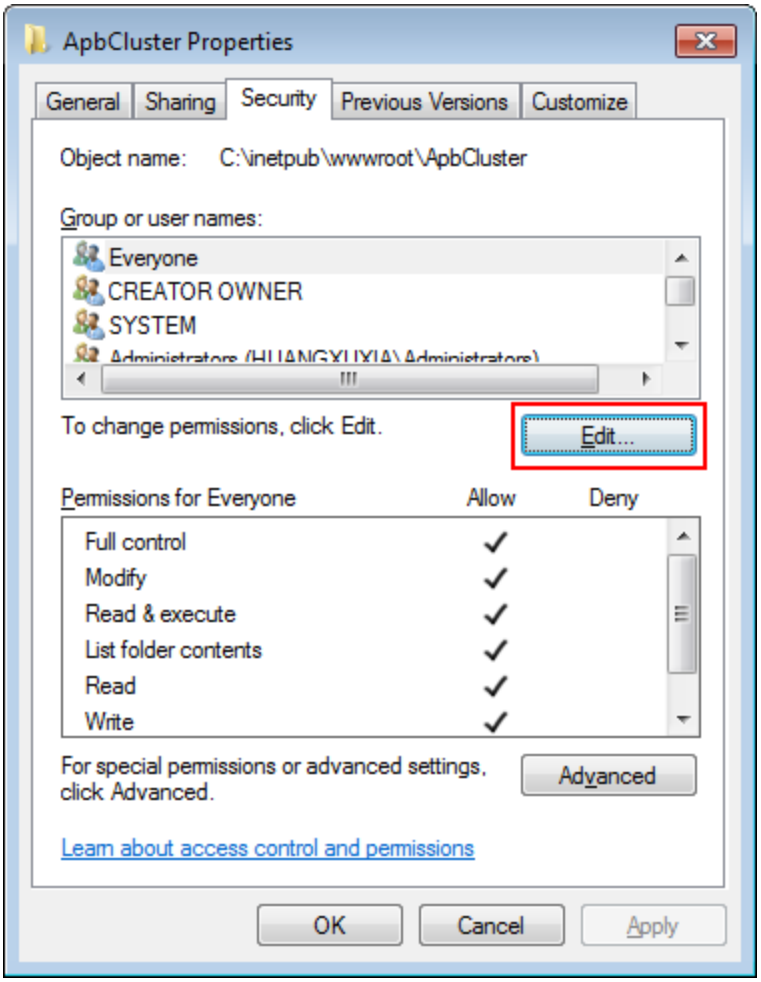
**Figure 6.13: Add virtual directory**




Step 6: Right click the **ApbCluster** virtual directory and select **Edit Permissions**.

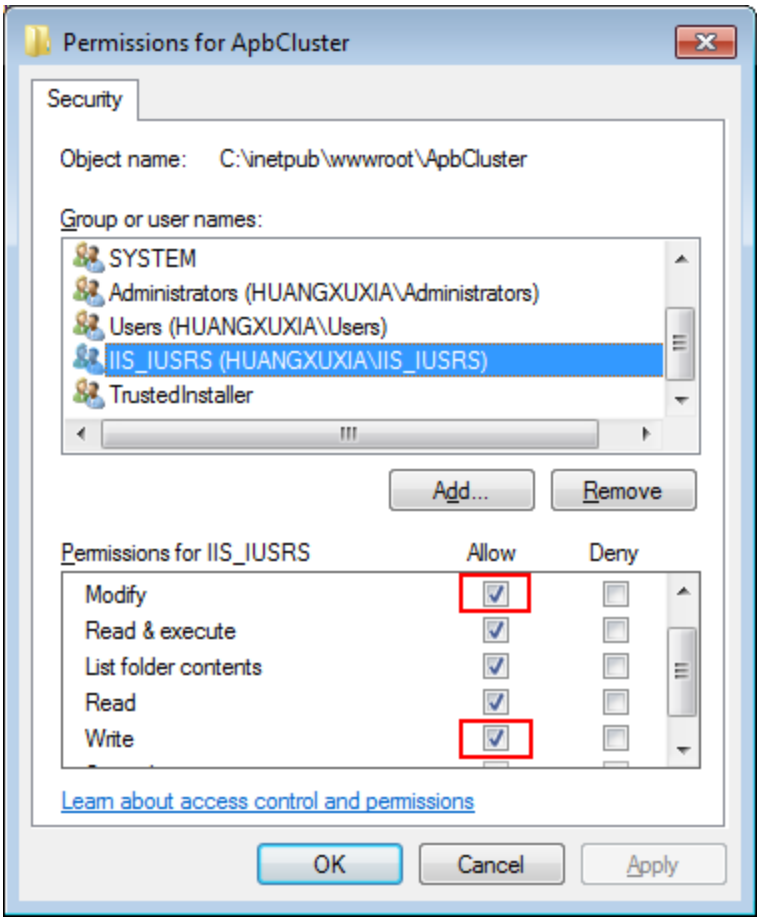
Step 7: Choose the **Security** tab, and then click the **Edit** button.

**Figure 6.14: ApbCluster properties**




Step 8: Select **IIS_IUSRS**, for example, IIS_IUSRS (HUANGXUXIA\IIS_IUSRS) from the **Group or user names** list, and then select the **Allow** check box for **Modify** and **Write** in the **Permissions for IIS_IUSRS** list.

**Figure 6.15: ApbCluster permissions**




Step 9: Click **OK** to go back to the IIS window.

3. Copying the Appeon plug-in (**ApbCluster.dll** or **ApbCluster64.dll**).

   Copy the **ApbCluster.dll** (for 32-bit OS) or **ApbCluster64.dll** (for 64-bit OS) plug-in from **appeon\IISSupport\modules** to the newly created virtual directory (**C:\inetpub \wwwroot\ApbCluster** in this guide).

   In the 64-bit OS, you can either use **ApbCluster.dll** or **ApbCluster64.dll**. If you use **ApbCluster.dll**, you will need to set the **Enable 32-Bit Applications** to **True** for the application pool used by the current Web site. If you use **ApbCluster64.dll**, then set this option to **False**.

4. Creating the Appeon cluster configuration file (**cluster-config.xml**).

   Create the **cluster-config.xml** file in the newly created virtual directory (**C:\inetpub \wwwroot\ApbCluster** in this guide). The **cluster-config.xml** file acts as the redirector configuration file which helps the Web server redirect requests to the PowerServer according to the IP addresses/ports of PowerServer, and the load balancing algorithm that it stores.

```
<?xml version="1.0" encoding="UTF-8" ?>
<cluster-config local="true" log="debug" timeout="30" polling="5"
 arithmetic="sequence">
    <extention>/ApbCluster/ApbCluster.dll</extention>
```

```
    <filters>
        <filter>/AEM</filter>
        <filter>/servlet</filter>
        <filter>/reportfile</filter>
        <filter>/imagefile</filter>
        <filter>/dwfile</filter>
        <filter>/ajsdown</filter>
    </filters>
    <app-servers>
        <app-server host="192.0.0.168" port="88" />
        <app-server host="192.0.3.131" port="80" />
        <app-server host="192.0.3.145" port="80" />
    </app-servers>
</cluster-config>
```

Notes:

- **local** specifies whether to read the configurations on the local server or the cloud server. For the cloud server, set **Local** to **False**.
- **log** specifies the level of information to be recorded in the log file. It can be set to the following level: **debug**, **info**, **warning**, **error**, or **off**.
- **timeout** specifies the number of seconds the Web server waits between sending an HTTP request to PowerServer and receiving information from it. The HTTP connection timeout value is 2 times of this value.
- **polling** indicates the number of seconds when the configurations will be read again.
- **arithmetic**="random" indicates that the random algorithm is used to pick a PowerServer; **arithmetic**="sequence" indicates that the round-robin algorithm is used.
- **extention** specifies the location of the virtual directory which contains ApbCluster.dll (or ApbCluster64.dll), that is /*virtual-directory-alias-name*/ApbCluster.dll (/**ApbCluster/ApbCluster.dll** in this guide).
- **filter** specifies the type of pages to redirect.
- **host** indicates the IP address (recommended) or machine name of the PowerServer in the cluster.
- **port** indicates the port number of the PowerServer in the cluster.

5. Installing the Appeon plug-in as ISAPI Filters.

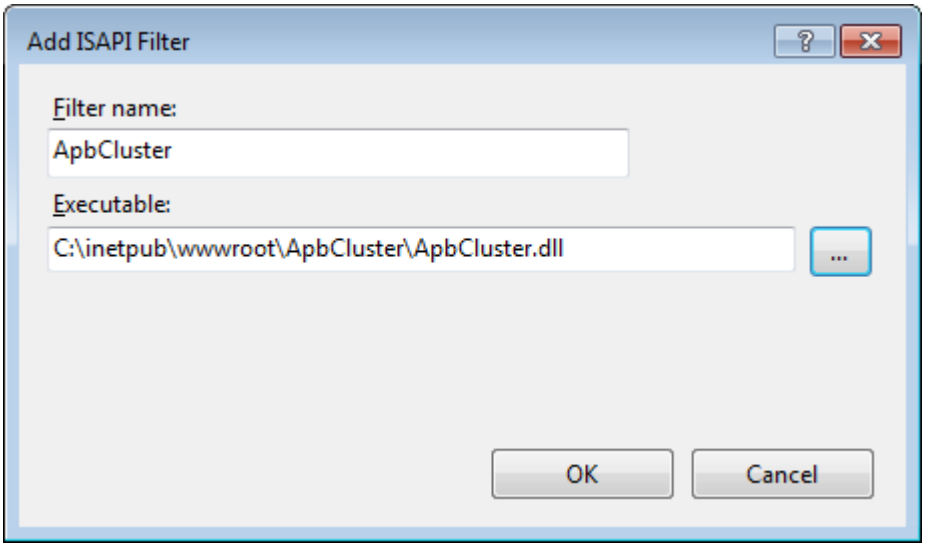
   Step 1: Open **Control Panel** | **Administrative Tools** | **Internet Information Services (IIS) Manager**.

   Step 2: Select **Default Web Site** and then double-click **ISAPI Filters** on the right side.

   Step 3: Right-click the blank area on the **ISAPI Filters** page, and click **Add**.

   Step 4: Input **ApbCluster** in the **Filter name** box and specify **ApbCluster.dll** (or **ApbCluster64.dll**) as the ISAPI filter in the **Executable** box (**C:\inetpub\wwwroot\ApbCluster\ApbCluster.dll** in this guide). Click **OK**.

**Figure 6.16: Add ISAPI filter**




6. Copying the redirector configuration file (**ApbCluster.cfg**).

   Copy the **ApbCluster.cfg** redirector configuration file from **appeon\IISSupport\conf** to the virtual directory (**C:\inetpub\wwwroot\ApbCluster** in this guide) and edit the file when necessary.

```
Extension_URI=/ApbCluster/ApbCluster.dll
MatchExpression=/AEM
MatchExpression=/servlet
MatchExpression=/reportfile
MatchExpression=/imagefile
MatchExpression=/dwfile
MatchExpression=/ajsdown
Log=On
```

   Notes:

   - The **Extension_URI** command points to the virtual directory where **ApbCluster.dll** (or **ApbCluster64.dll**) resides, that is /*virtual-directory-alias-name*/ApbCluster.dll (/**ApbCluster/ApbCluster.dll** in this guide).
   - The **MatchExpression** command specifies the pages to be redirected.
   - The **Log** command specifies whether logging is enabled. On indicates logging is enabled; Off indicates logging is disabled.
   - The commands and parameters are case insensitive.

7. Setting ISAPI and CGI Restrictions.

   Step 1: Open **Control Panel** | **Administrative Tools** | **Internet Information Services (IIS) Manager**.

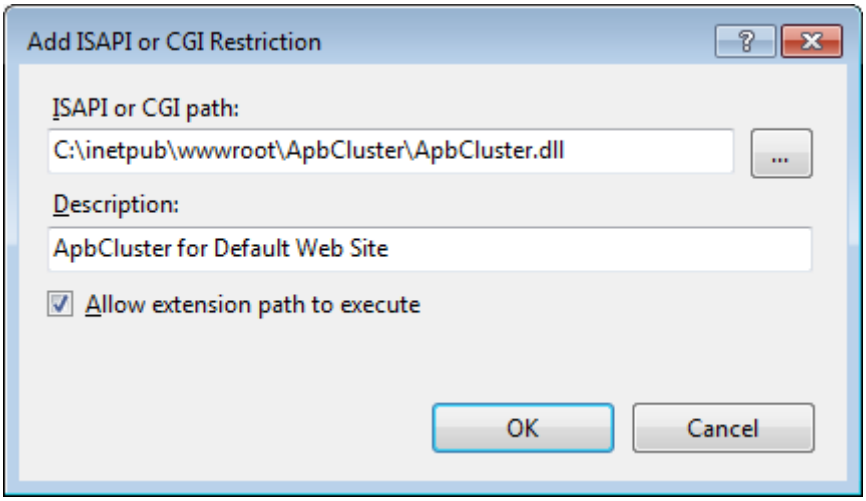
   Step 2: Select the local machine (the top node), and then double-click **ISAPI and CGI Restrictions** on the right side.

Step 3: Right-click the blank area of the **ISAPI and CGI Restrictions** page and click **Add**.

Step 4: Specify the local path of **ApbCluster.dll** (or **ApbCluster64.dll**) in the **ISAPI or CGI path** box and input the descriptive text in the **Description** box, and then select **Allow extension path to execute**.

Step 5: Click **OK**.

**Figure 6.17: Add ISAPI or CGI**




8. Enabling ISAPI-dll.

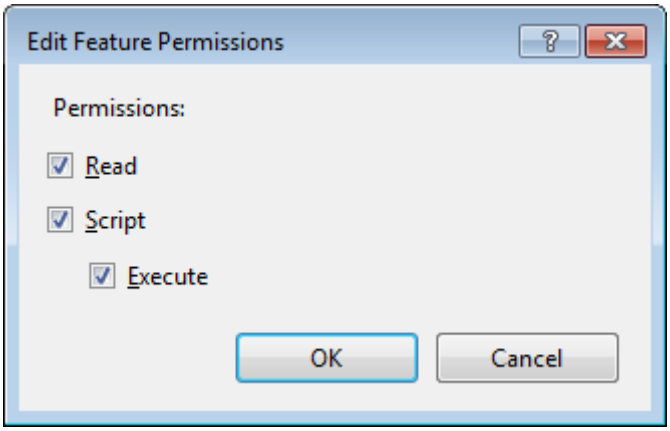
Step 1: Open **Control Panel** | **Administrative Tools** | **Internet Information Services (IIS) Manager**.

Step 2: Expand **Default Web Site**, select **ApbCluster**, and then double-click **Handler Mappings** on the right side.

Step 3: Right-click **ISAPI-dll**, and select **Edit Feature Permissions**.

Step 4: Select **Read**, **Script**, **Execute**, and click **OK**.

**Figure 6.18: Edit feature permissions**



9. Adding MIME type for loading the pages without extension names or pages without MIME types defined.

   Step 1: Select **Default Web Site** and then double-click **MIME Types** on the right of the page.

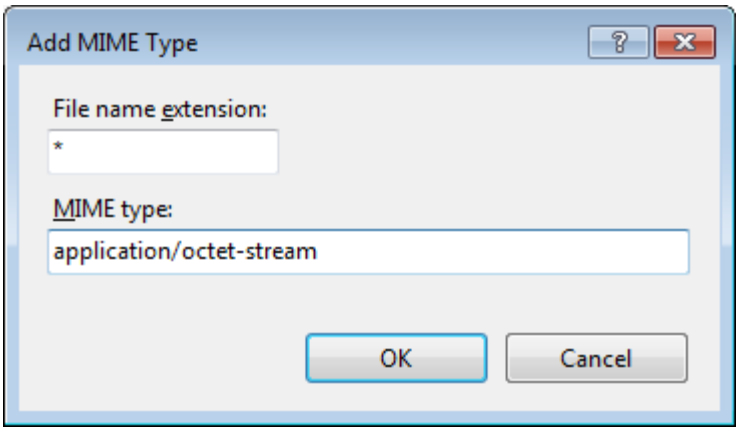
   Step 2: Right-click the blank area of the **MIME Types** page, and click **Add**.

   Step 3: Specify the following settings:

   - Extension: *
   - MIME Type: application/octet-stream

   Step 4: Click **OK**.

   **Figure 6.19: Add MIME type**




10. Restarting IIS.

    Restart IIS to make the new settings effective.

#### Configuring IIS 7.x with J2EE server

J2EE servers refer to the Java application servers, such as WebLogic, WebSphere, JBoss, JEUS etc. The configuration is the same for the J2EE application servers that Appeon supports.

Choose an existing Web site, or create a new site. The **Default Web Site** will be used as examples in the following guide.

1. Installing PowerServer Web component to the Web root of the IIS server.

   Make sure you have installed PowerServer Web Component to the Web root of the IIS server. To verify it, go to the IIS Web root (by default C:\inetpub\wwwroot) and check if there is an **appeon** folder which contains the **IISSupport** and **weblibrary_ax** sub-folders. If any folder is missing, go to the default installation directory of PowerServer Web Component (C:\Program Files\Appeon\WebComponent2017) and manually copy the entire **appeon** folder to the IIS Web root.

2. Creating a virtual directory.

Step 1: Open **Control Panel** | **Administrative Tools** | **Internet Information Service (IIS) Manager**.

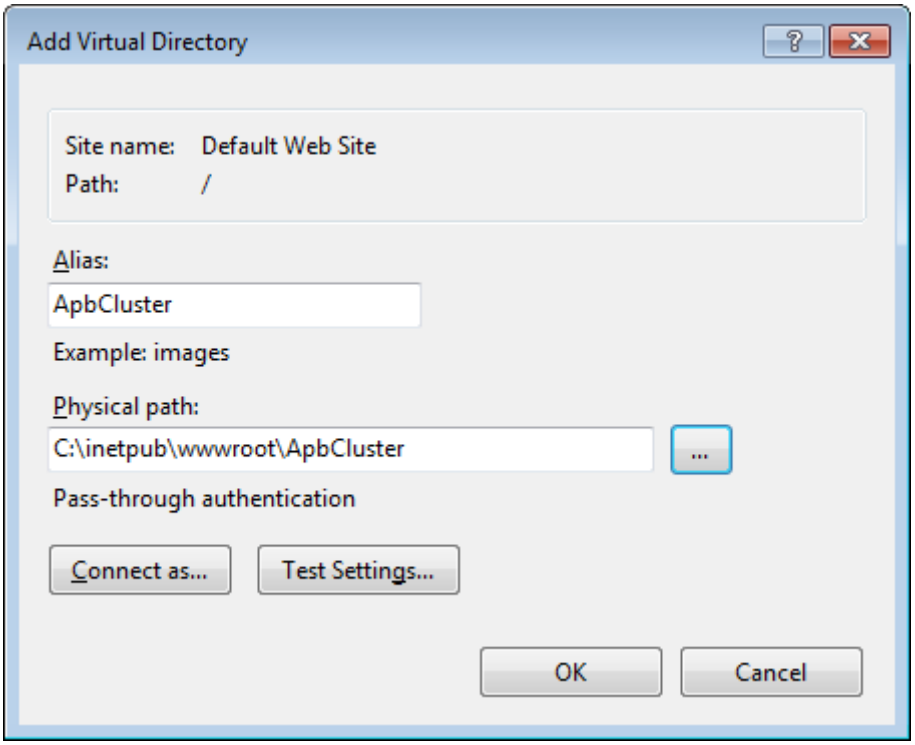
Step 2: Right click **Default Web Site** and select **Add Virtual Directory**.

Step 3: Input the alias name, for example, "**ApbCluster**".

Step 4: Choose a local path in the **Physical path** box; for example, "**C:\inetpub\wwwroot\ApbCluster**".

Step 5: Click **OK**.

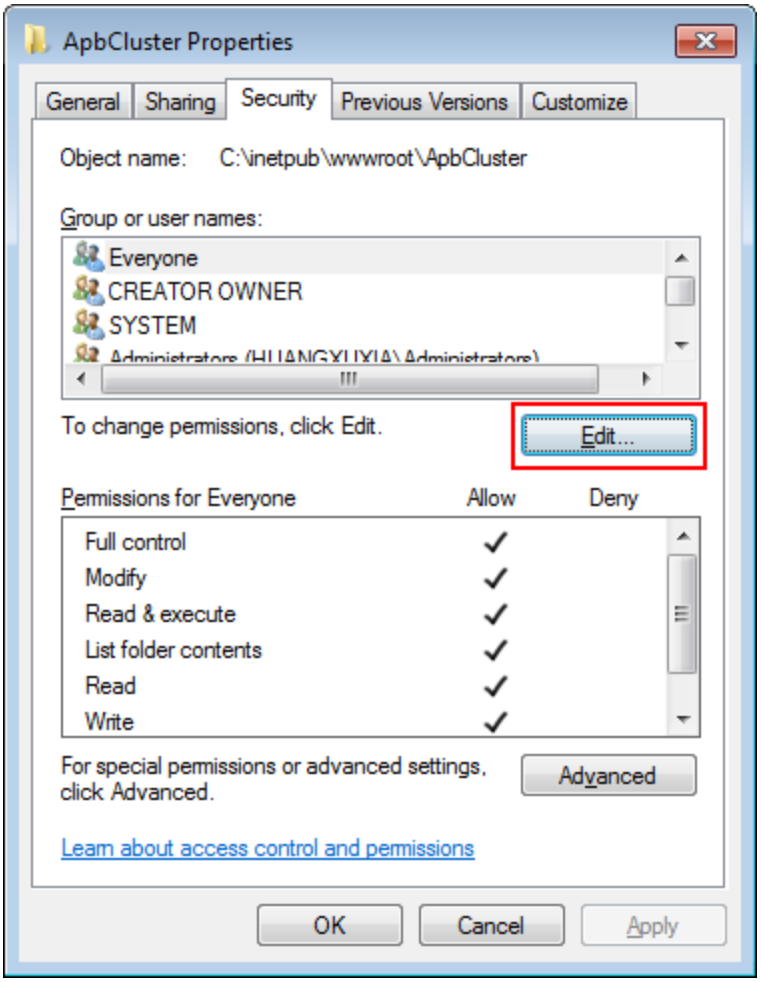
**Figure 6.20: Add virtual directory**




Step 6: Right click the **ApbCluster** virtual directory and select **Edit Permissions**.

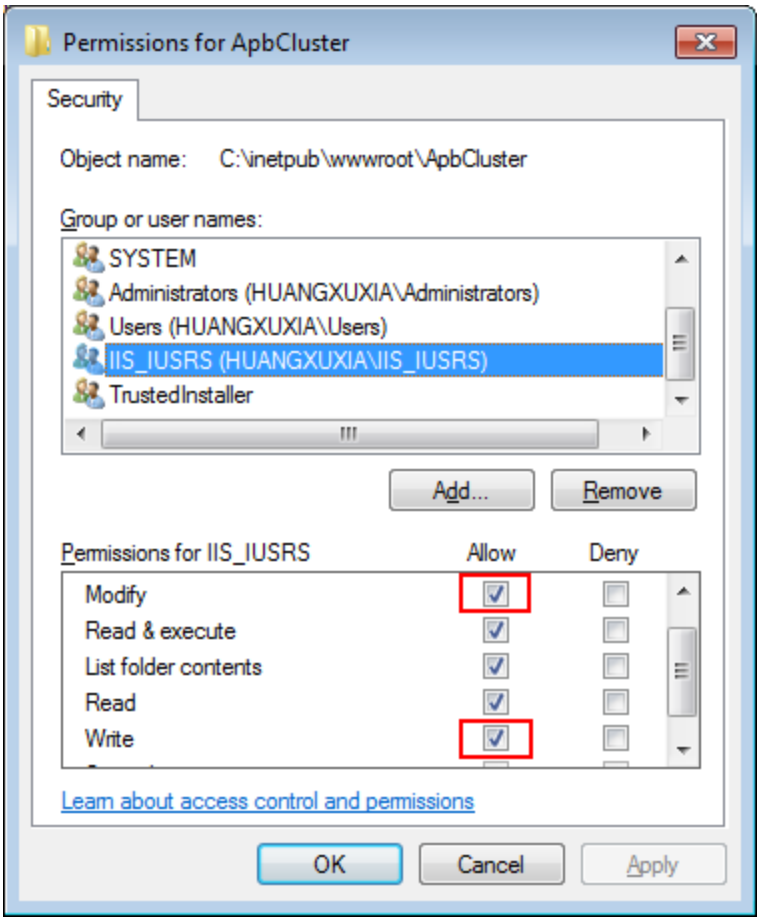
Step 7: Choose the **Security** tab page and then click the **Edit** button.

**Figure 6.21: appeoncluster properties**




Step 8: Select **IIS_IUSRS**, for example, IIS_IUSRS (HUANGXUXIA\IIS_IUSRS) from the **Group or user names** list, and then select the **Allow** check box for **Modify** and **Write** in the **Permissions for IIS_IUSRS** list.

**Figure 6.22: appeoncluster properties**




Step 9: Click **OK** to go back to the IIS window.

3. Copying the Appeon plug-in (**ApbCluster.dll** or **ApbCluster64.dll**).

   Copy the **ApbCluster.dll** (for 32-bit OS) or **ApbCluster64.dll** (for 64-bit OS) plug-in from **appeon\IISSupport\modules** to the newly created virtual directory (**C:\inetpub \wwwroot\ApbCluster** in this guide).

   In the 64-bit OS, you can either use **ApbCluster.dll** or **ApbCluster64.dll**. If you use **ApbCluster.dll**, you will need to set the **Enable 32-Bit Applications** to **True** for the application pool used by the current Web site. If you use **ApbCluster64.dll**, then set this option to **False**.

4. Creating the Appeon cluster configuration file (**cluster-config.xml**).

   Create the **cluster-config.xml** file in the newly created virtual directory (**C:\inetpub \wwwroot\ApbCluster** in this guide). The **cluster-config.xml** file acts as the redirector configuration file which helps the Web server redirect requests to the PowerServer according to the IP addresses/ports of PowerServer, and the load balancing algorithm that it stores.

```
<?xml version="1.0" encoding="UTF-8" ?>
<cluster-config local="true" log="debug" timeout="30" polling="5"
 arithmetic="sequence">
    <extention>/ApbCluster/ApbCluster.dll</extention>
```

```
    <filters>
        <filter>/AEM</filter>
        <filter>/servlet</filter>
        <filter>/reportfile</filter>
        <filter>/imagefile</filter>
        <filter>/dwfile</filter>
        <filter>/ajsdown</filter>
    </filters>
    <app-servers>
        <app-server host="192.0.0.168" port="88" />
        <app-server host="192.0.3.131" port="80" />
        <app-server host="192.0.3.145" port="80" />
    </app-servers>
</cluster-config>
```

Notes:

- **local** specifies whether to read the configurations on the local server or the cloud server. For the cloud server, set **Local** to **False**.
- **log** specifies the level of information to be recorded in the log file. It can be set to the following level: **debug**, **info**, **warning**, **error**, or **off**.
- **timeout** specifies the number of seconds the Web server waits between sending an HTTP request to PowerServer and receiving information from it. The HTTP connection timeout value is 2 times of this value.
- **polling** indicates the number of seconds when the configurations will be read again.
- **arithmetic**="random" indicates that the random algorithm is used to pick a PowerServer; **arithmetic**="sequence" indicates that the round-robin algorithm is used.
- **extention** specifies the location of the virtual directory which contains ApbCluster.dll (or ApbCluster64.dll), that is /*virtual-directory-alias-name*/ApbCluster.dll (/**ApbCluster/ApbCluster.dll** in this guide).
- **filter** specifies the type of pages to redirect.
- **host** indicates the IP address (recommended) or machine name of the PowerServer.
- **port** indicates the port number of the PowerServer.

5. Installing the Appeon plug-in as ISAPI Filters.

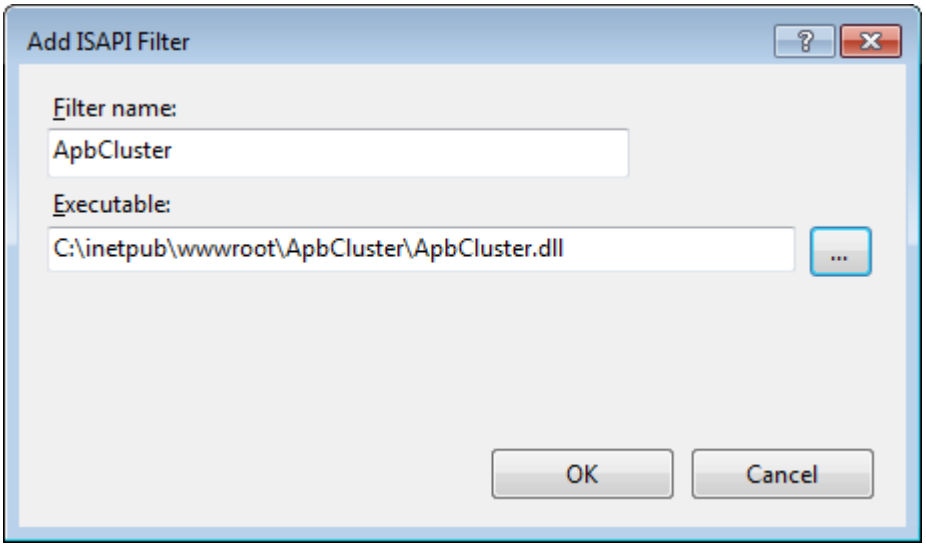
   Step 1: Open **Control Panel** | **Administrative Tools** | **Internet Information Service (IIS) Manager**.

   Step 2: Select **Default Web Site** and then double-click **ISAPI Filters** on the right side.

   Step 3: Right-click the blank area on the **ISAPI Filters** page, and click **Add**.

   Step 4: Input **ApbCluster** in the **Filter name** box and specify **ApbCluster.dll** (or **ApbCluster64.dll**) as the ISAPI filter in the **Executable** box (***C:\inetpub\wwwroot\ApbCluster\ApbCluster.dll*** in this guide). Click **OK**.

**Figure 6.23: Add ISAPI filter**




6. Copying the redirector configuration file (**ApbCluster.cfg**).

   Copy the **ApbCluster.cfg** redirector configuration file from **appeon\IISSupport\conf** to the virtual directory (**C:\inetpub\wwwroot\ApbCluster** in this guide) and edit the file when necessary.

```
Extension_URI=/ApbCluster/ApbCluster.dll
MatchExpression=/AEM
MatchExpression=/servlet
MatchExpression=/reportfile
MatchExpression=/imagefile
MatchExpression=/dwfile
MatchExpression=/ajsdown
Log=On
```

   Notes:

   - The **Extension_URI** command specifies the location of the virtual directory which contains **ApbCluster.dll** (or **ApbCluster64.dll**), that is /*virtual-directory-alias-name*/ ApbCluster.dll (/**ApbCluster/ApbCluster.dll** in this guide).
   - The **MatchExpression** commands specify the type of pages to redirect.
   - The **Log** command specifies whether to enable logging. On indicates logging is enabled; Off indicates logging is disabled.
   - The commands and parameters are case insensitive.

7. Setting ISAPI and CGI Restrictions.

   Step 1: Open **Control Panel** | **Administrative Tools** | **Internet Information Services (IIS) Manager**.

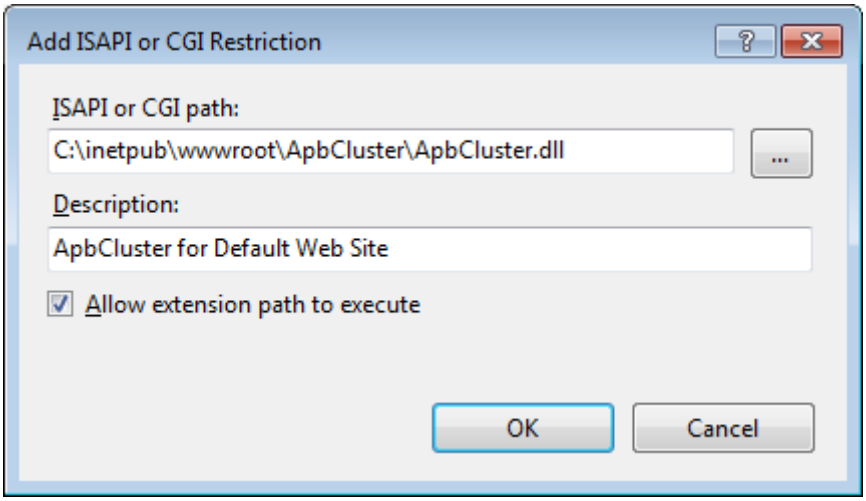
   Step 2: Select the local machine (the top node), and then double-click **ISAPI and CGI Restrictions** on the right side.

Step 3: Right-click the blank area of the **ISAPI and CGI Restrictions** page and click **Add**.

Step 4: Specify the local path of **ApbCluster.dll** (or **ApbCluster64.dll**) in the **ISAPI or CGI path** box and input the descriptive text in the **Description** box, and then select **Allow extension path to execute**.

Step 5: Click **OK**.

**Figure 6.24: Add ISAPI or CGI**




8. Enabling ISAPI-dll.

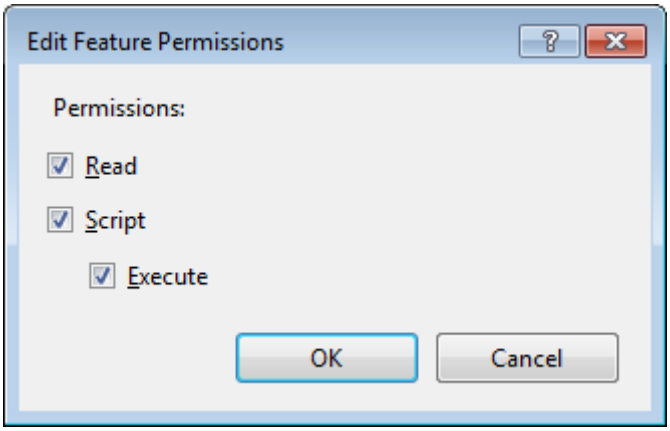
   Step 1: Open **Control Panel** | **Administrative Tools** | **Internet Information Services (IIS) Manager**.

   Step 2: Expand **Default Web Site**, select **ApbCluster**, and then double-click **Handler Mappings** on the right side.

   Step 3: Right-click **ISAPI-dll**, and select **Edit Feature Permissions**.

   Step 4: Select **Read**, **Script**, **Execute**, and click **OK**.

**Figure 6.25: Edit feature permissions**




9. Adding MIME type for loading the pages without extension names or pages without MIME types defined.

Step 1: Select **Default Web Site** and then double-click **MIME Types** on the right of the page.

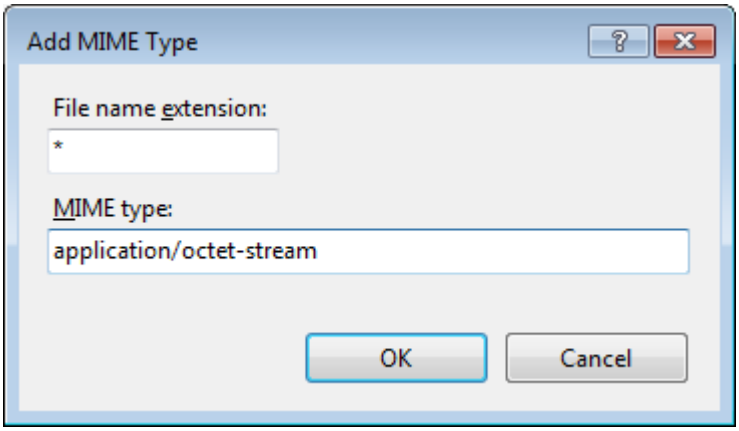
Step 2: Right-click the blank area of the **MIME Types** page, and click **Add**.

Step 3: Specify the following settings:

- Extension: *
- MIME Type: application/octet-stream

Step 4: Click **OK**.

**Figure 6.26: Add MIME type**




10. Restarting IIS.

    Restart IIS to make the new settings effective.

#### 6.4.3.2 Type 2: Apache Web server

##### 6.4.3.2.1 Manual configuration

**Configuring Apache 2.0/22 with J2EE servers**

J2EE servers refer to Java application servers, such as WebLogic, WebSphere, JBoss, JEUS etc. The configuration is the same for the J2EE application servers that Appeon supports.

1. Installing PowerServer Web component to the document root of the Apache Web server.

   Make sure you have installed PowerServer Web Component to the document root of the Apache Web server. To verify it, go to the Apache document root (by default %APACHE %\htdocs) and check if there is an **appeon** folder which contains the **Apache20Support**, **Apache22Support** and **weblibrary_ax** sub-folders. If any folder is missing, go to the default installation directory of PowerServer Web Component (C:\Program Files \Appeon\WebComponent2017) and manually copy the entire **appeon** folder to the Apache document root.

2. Copying Appeon plug-in.

   Copy the **mod_appeon2.so** or **mod_appeon22.so** plug-in file from %APACHE%\htdocs \appeon\Apache2xSupport\modules to the %APACHE%\modules folder.

3. Copying Appeon Cluster configuration file.

   Copy the **cluster-config.xml** file from the %appeon%\repository\<instancename>\config folder to the %APACHE%\conf folder, where %appeon% is the installation directory of PowerServer, and <instancename> refers to the name of the PowerServer instance.

   Note: The **cluster-config.xml** file acts as the redirector configuration file which helps the Web server redirect requests to PowerServer according to the IP addresses/ports of PowerServer, and the load balancing algorithm that it stores. This file should not be modified manually as it will be automatically updated when you change the settings in the **Cluster** tool of AEM. Each time after you change the settings, be sure to copy this file and replace the old file in the %APACHE%\conf folder and restart Apache.

4. Modifying httpd.conf.

   If the application server (such as WebLogic, WebSphere, JBoss, or JEUS) is running on Windows, add the following statements into the httpd.conf file under the %APACHE% \conf folder (take mod_appeon2.so as an example):

```
LoadModule appeon_module modules\mod_appeon2.so
<IfModule mod_appeon.c>
MatchExpression /AEM
MatchExpression /servlet
MatchExpression /reportfile
MatchExpression /imagefile
MatchExpression /dwfile
MatchExpression /ajsdown
</IfModule>
```

   If the application server is running on UNIX/LINUX, add the following statements into the httpd.conf file under the %APACHE%\conf folder (take mod_appeon2.so as an example):

```
LoadModule appeon_module modules/mod_appeon2.so
<IfModule mod_appeon.c>
MatchExpression /AEM
MatchExpression /servlet
MatchExpression /reportfile
MatchExpression /imagefile
MatchExpression /dwfile
MatchExpression /ajsdown
</IfModule>
```

5. Restarting Apache.

   Restart Apache for the new settings to take effect.

#### Configuring Apache 1.3 with J2EE servers

J2EE servers refer to the Java application servers, such as WebLogic, WebSphere, JBoss, JEUS etc. The configuration is the same for the J2EE application servers that Appeon supports.

1. Installing PowerServer Web component to the document root of the Apache Web server.

   Make sure you have installed PowerServer Web Component to the document root of the Apache Web server. To verify it, go to the Apache document root (by default %APACHE %\htdocs) and check if there is an **appeon** folder which contains the **Apache20Support**,

**Apache22Support** and **weblibrary_ax** sub-folders. If any folder is missing, go to the default installation directory of PowerServer Web Component (C:\Program Files \Appeon\WebComponent2017) and manually copy the entire **appeon** folder to the Apache document root.

2. Copying Appeon plug-in.

   Copy the **mod_appeon.so** plug-in file from %APACHE%\htdocs\appeon \Apache13Support\modules to the %APACHE%\modules folder.

3. Copying Appeon Cluster configuration file.

   Copy the **cluster-config.xml** file from the %appeon%\repository\<instancename>\config folder to the %APACHE%\conf folder, where %appeon% is the installation directory of PowerServer, and <instancename> refers to the name of a server instance.

   Note: The **cluster-config.xml** file acts as the redirector configuration file which helps the Web server redirect requests to PowerServer according to the IP addresses/ports of PowerServer, and the load balancing algorithm that it stores. This file should not be modified manually as it will be automatically updated when you change the settings in the **Cluster** tool of AEM. Each time after you change the settings, be sure to copy this file and replace the old file in the %APACHE%\conf folder and restart Apache.

4. Modifying httpd.conf.

   Add the following statements into the httpd.conf file under the %APACHE%\conf folder:

```
LoadModule appeon_module modules\mod_appeon.so
<IfModule mod_appeon.c>
MatchExpression /AEM
MatchExpression /servlet
MatchExpression /reportfile
MatchExpression /imagefile
MatchExpression /dwfile
MatchExpression /ajsdown
</IfModule>
```

5. Restarting Apache.

   Restart Apache for the new settings to take effect.

### 6.4.4 Special instructions for Auto Scaling in Amazon Web Service

Auto scaling in the Amazon Web Service (not Windows Azure) cloud server is supported by PowerServer for .NET (not J2EE). Make sure the cloud server is activated against the Appeon cloud license; for instructions on activating the cloud server, refer to Section 5.3.4.1.1, “Product Activation” in *PowerServer Configuration Guide for .NET*. In this section, instructions are provided for supporting auto scaling in the Amazon Web Service cloud server.

#### 6.4.4.1 Modifying the Appeon cluster config file

After installing the Appeon cluster plug-in, modify the Appeon cluster config file under the installation folder: AppeonCluster\AppeonCloud\Config\ClusterManager.config.

Specify the following parameters in the ClusterManager.config file.

```
<AWSStorage accessKey="" secretKey="" mode="" defaultPort="" keywordPrefix=""/>
```

- accessKey (access key ID), secretKey (access secret key), and keywordPrefix (keyword prefix), by following instructions in the section called “AWS server instance settings” in *PowerServer Configuration Guide for .NET*.
- defaultPort: port number of PowerServer, by default it is 80.
- mode: the switch for supporting auto scaling.
  - 1: supports auto scaling by automatically detecting PowerServers and updating the PowerServer list in the cluster.
  - 0: does not support auto scaling (so the developer needs to manually update the PowerServer list in the cluster).

#### 6.4.4.2 Modifying the PowerServer config file

Under the PowerServer installation folder, modify the PowerServer config file: appeon\AEM \config\cluster-config.config.

Specify the following parameters under the cluster-config node in the cluster-config.config file, by following the aforementioned instructions.

```
mode="" defaultport=""
```

## 6.5 Task 5: Deploy or Install the application to the PowerServer cluster and Web server(s)

To install an Appeon application to the PowerServer cluster and Web server(s), you can either deploy the application via the **Appeon Deployment Wizard** (see Chapter 6, *Deploying PowerBuilder Applications* in *PowerServer Toolkit User Guide*, or run the deployment project generated by using the **Appeon Application Package Wizard** (see Section 10.3, “Packaging a stand-alone mobile project” in *PowerServer Toolkit User Guide*. This section will talk about how to install an application by running the deployment project generated via the **Appeon Application Package Wizard**.

### 6.5.1 Points to note before installation

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

- The Setup program of the Web application or the mobile application runs on the Windows platform only. To install applications to servers running on Unix\Linux, you must run the Setup program on the Windows platform and then install the application to remote servers running on Unix\Linux.
- The target Web servers must have the PowerServer Web Component correctly installed.

### 6.5.2 Installing an application

You can copy the generated package to any computer running Windows and install the application to any number of PowerServer and Web servers. Installing an application

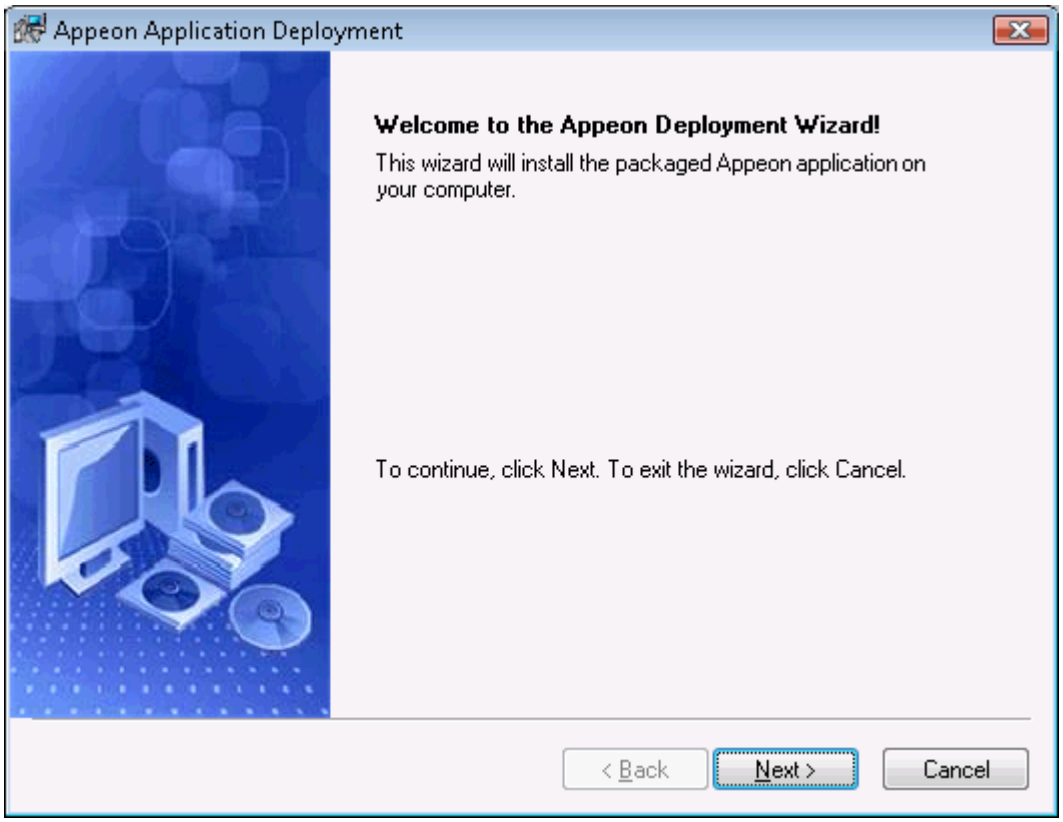
using the generated package is the same as deploying an application using the **Appeon Deployment Wizard**.

Step 1: Start the target PowerServer(s) and Web server(s) where you want to install the application.

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

The Appeon Application Deployment wizard is displayed. Click **Next** to proceed.

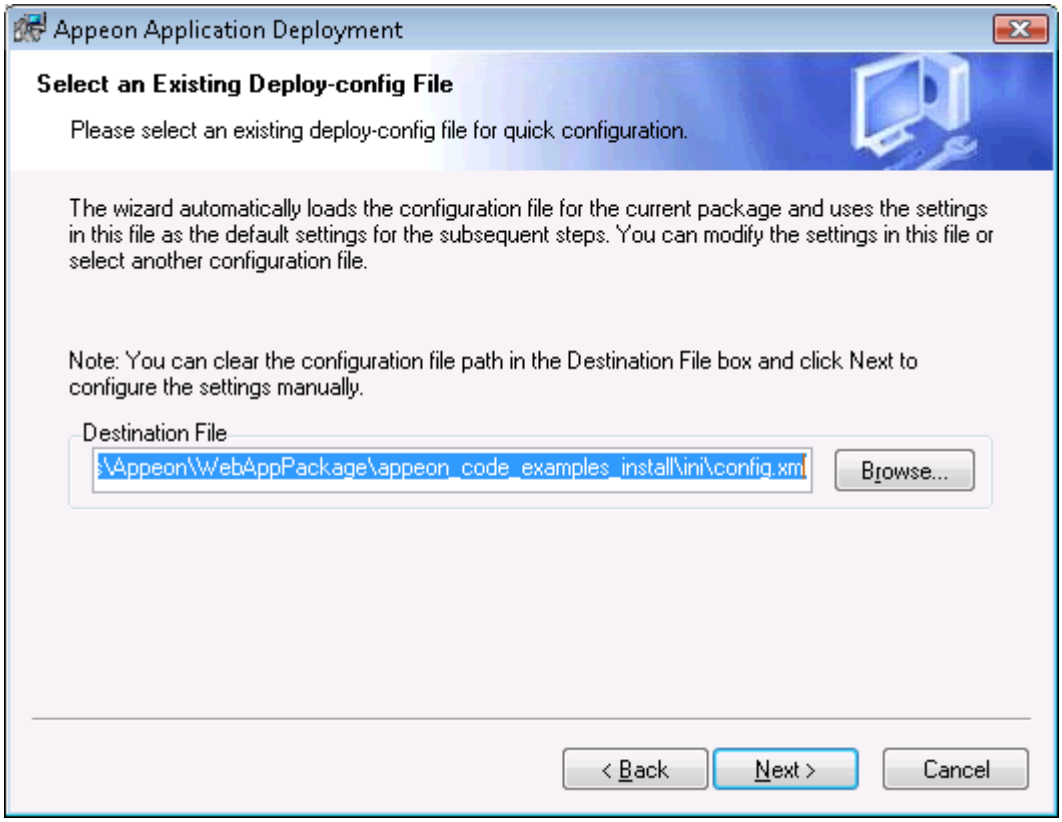
**Figure 6.27: Welcome page**




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

If you do not want to use the deploy-config file, simply skip this option and click **Next** to configure the settings step by step.

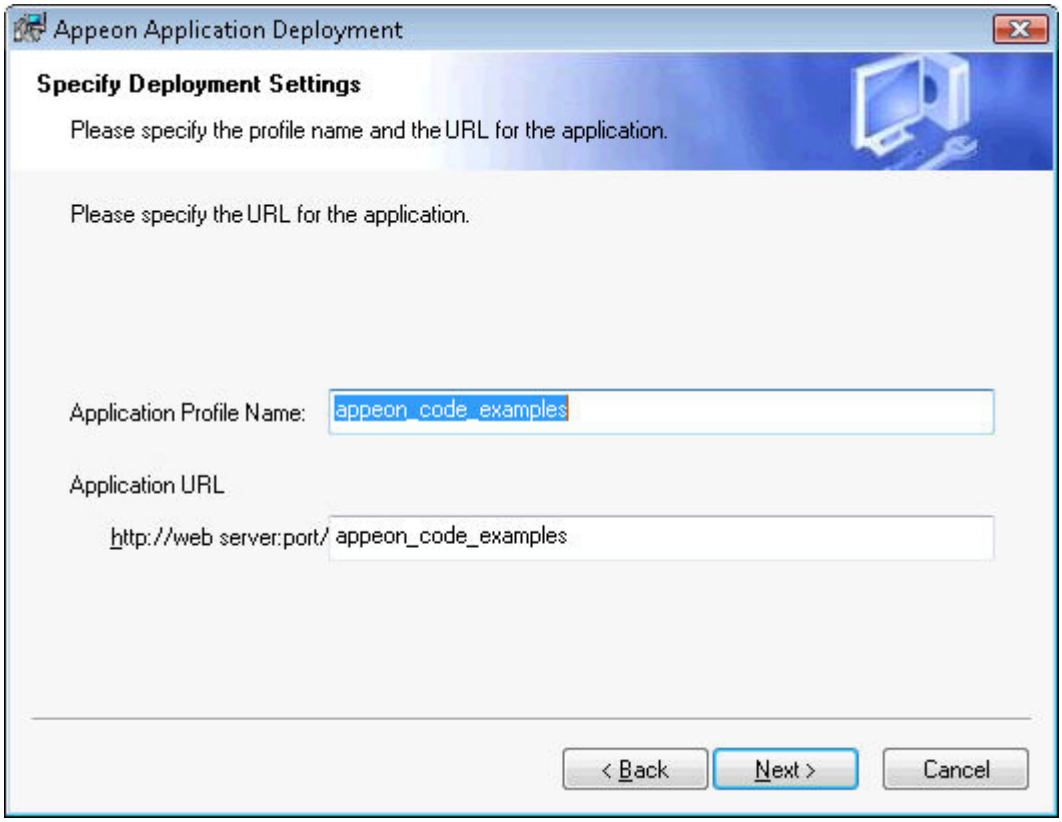
**Figure 6.28: Select the configuration file**




Step 4: Input the Application Profile Name and the Application URL for the application and click **Next**.

The Application Profile Name is used as the application name to identify an application. If you want to install multiple instances of the same application on the same server, you can run the setup package repeatedly and specify different Application Profile Name here. For example, input appeon_code_examples_**test** as the Application Profile Name in the first installation, and input appeon_code_examples_**production** as the Application Profile Name in the second installation. The Application Profile Name will be used as the application name to distinguish the multiple application instances on the same server, so they can be run independently from each other.

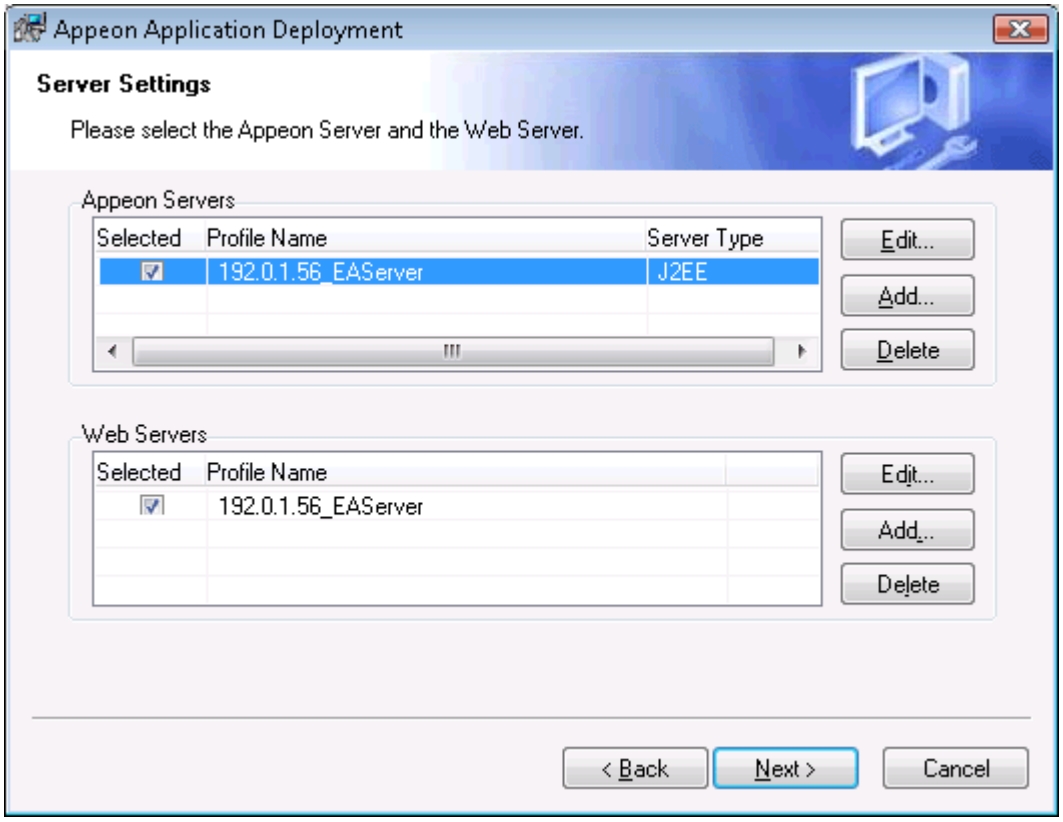
**Figure 6.29: Specify Application Profile Name and Application URL**




Step 5: Configure and select profiles for the PowerServer(s) and the Web server(s) where the application will be installed.

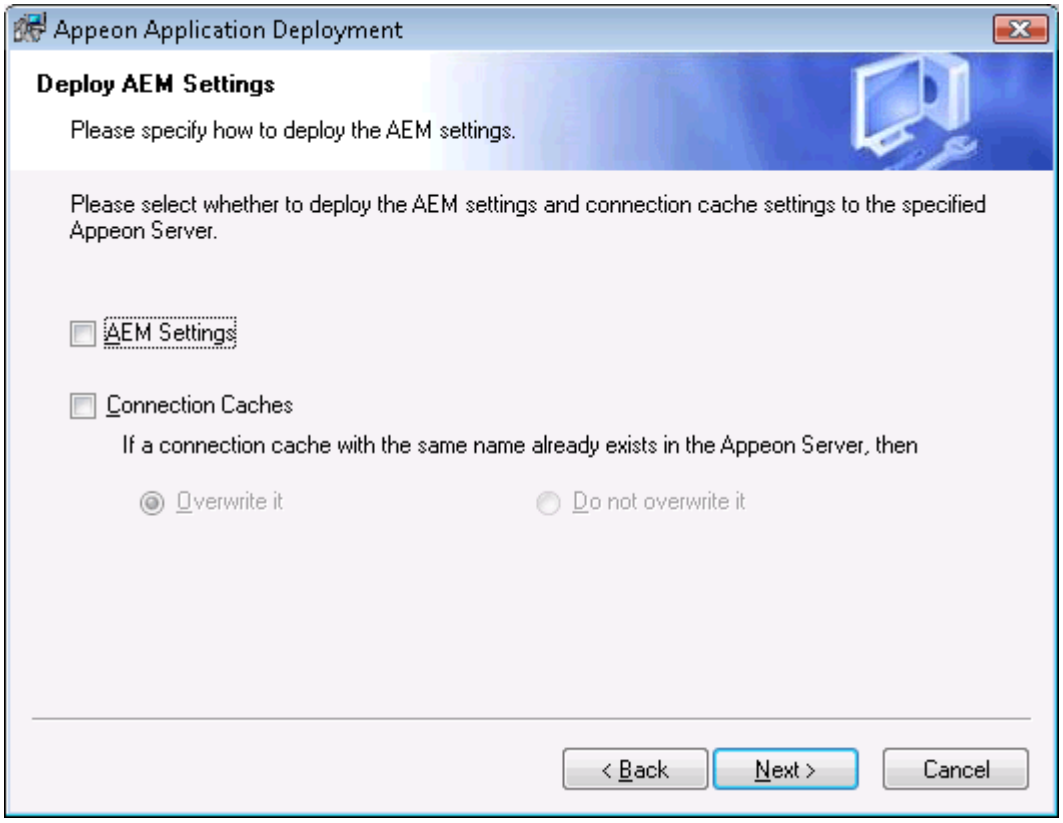
To install the application to a PowerServer cluster, please create a PowerServer profile for each PowerServer in the cluster. For detailed instructions on how to create the PowerServer profile and Web server profile, refer to Section 4.2.3, “Managing server profiles” in *PowerServer Toolkit User Guide*.

**Figure 6.30: Specify server settings**



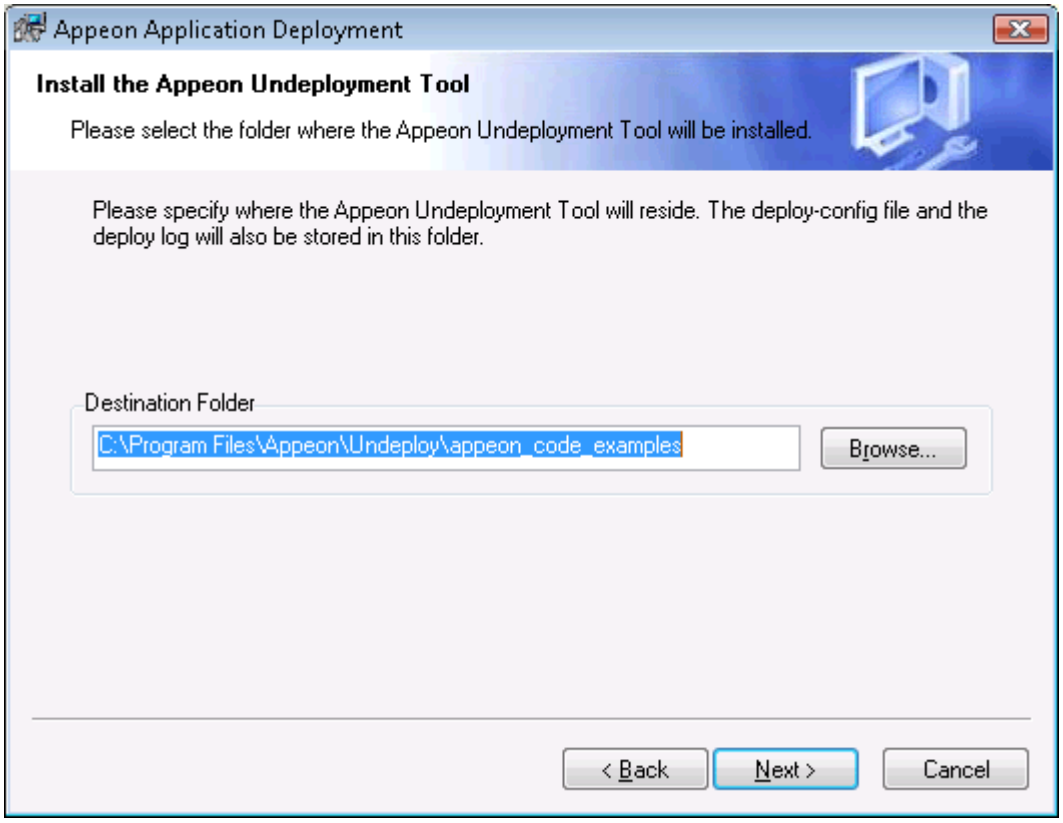

Step 6: Select whether to install AEM settings and data sources for the current application and click **Next**.

**Figure 6.31: Deploy AEM settings**



Step 7: Specify the destination folder for the undeployment tool and the log file and click **Next**.

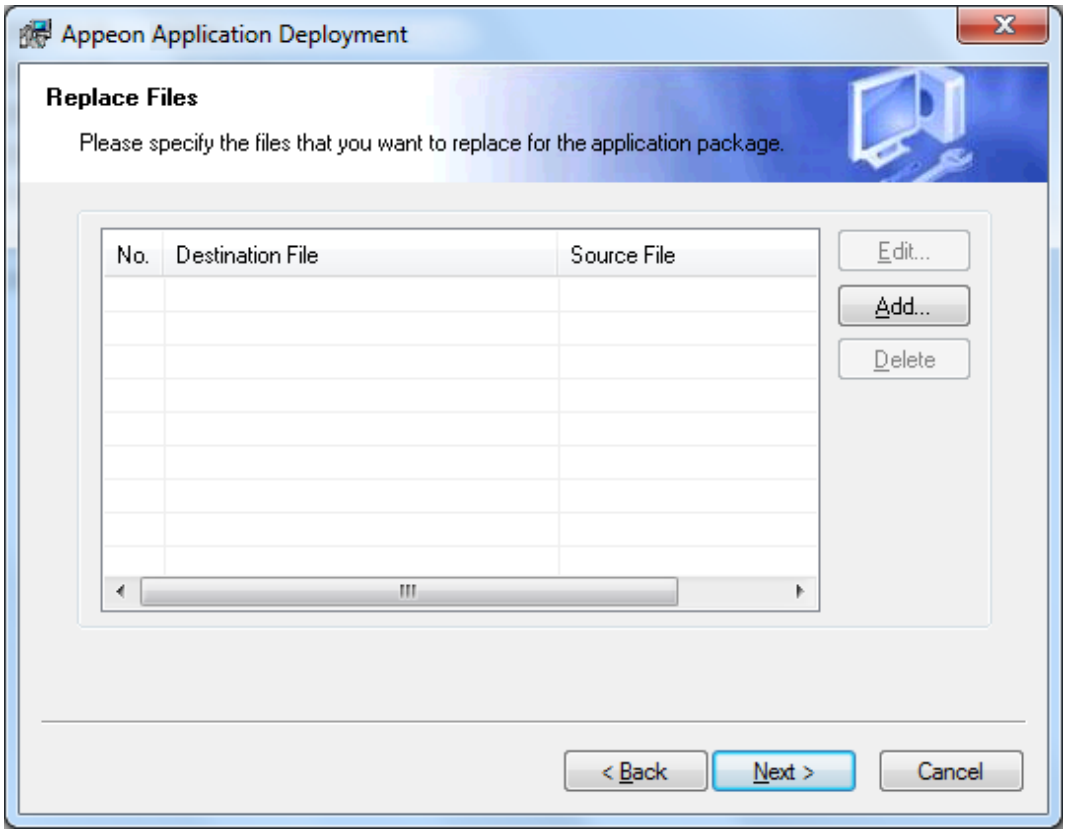
**Figure 6.32: Specify location for the Appeon Undeployment Tool**




Step 8: Specify the destination file and source file that you want to replace for the Appeon application package, and click **Next**.

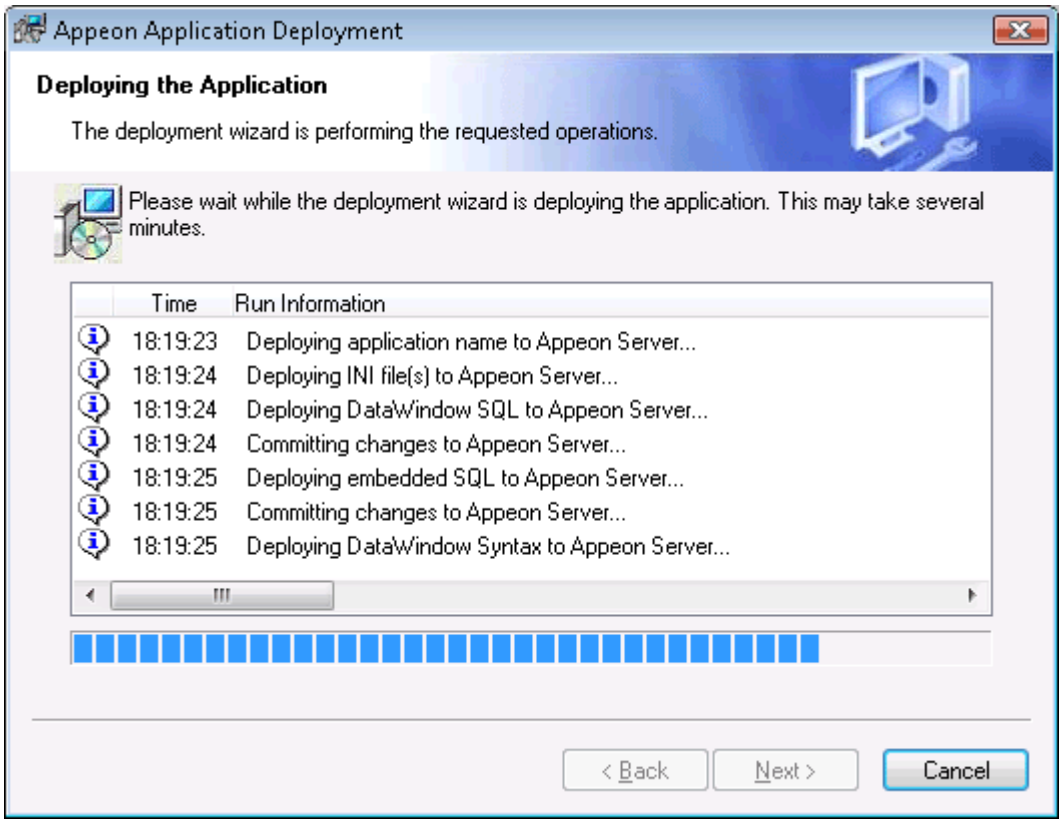
Files (also called "destination file") under the "wcode" folder of Appeon application package can be replaced by files of the same type (also called "source file"), so that the source file will be installed instead of the destination file. When you click the **Add** button, the **File Replacement** dialog box is opened. To select the destination file, click **Browse** to display the Open file dialog which will automatically open the "wcode" folder of Appeon application package, so you can conveniently select a file from the "wcode" folder or from its subfolder. To select the source file, click **Browse** to navigate to the folder where the source file is stored.

**Figure 6.33: Specify files to replace for the application package**




Step 9: Wait while the wizard is installing files and settings.

**Figure 6.34: Deploying the application**

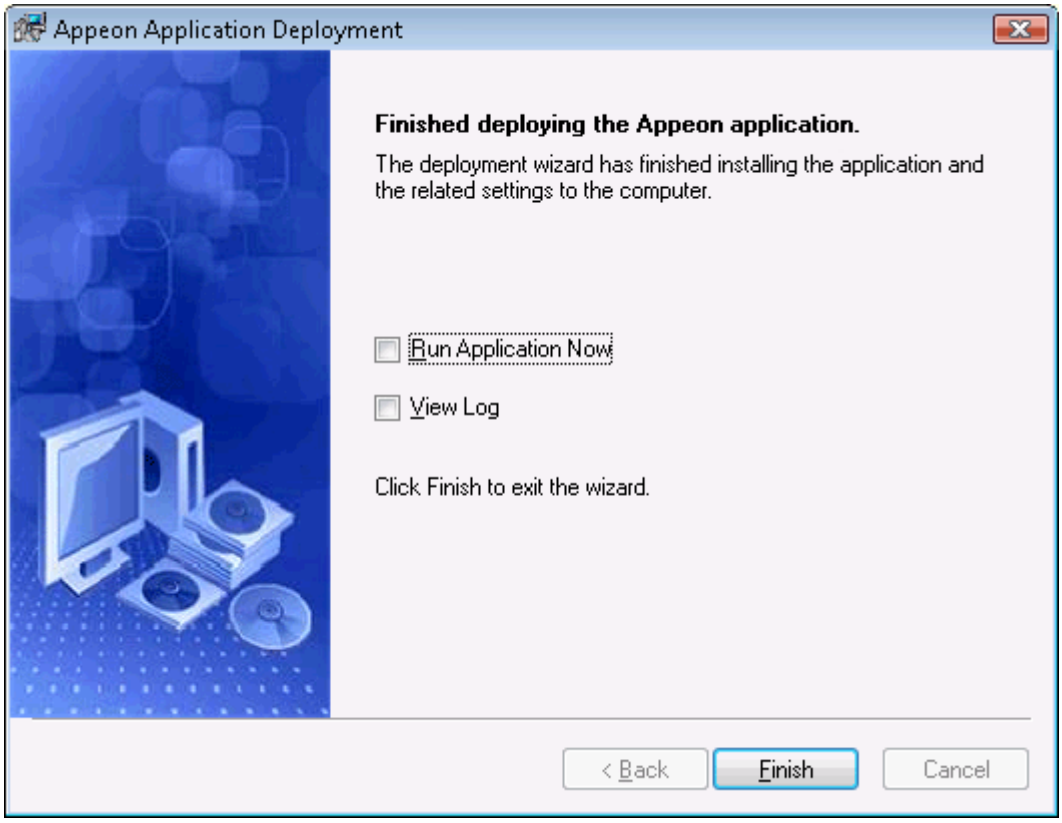


Step 10: Once the operation is complete, click **Finish** to exit the installation wizard.

To run the application immediately, select the **Run Application Now** option and click **Finish**. Or you can run the application later from the Windows **Start** | **Programs** | **Appeon Web Application** | *ApplicationName*.

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

**Figure 6.35: Deployment complete**



# Index