

Deploying the pbtutor Application as a Web Form to IIS 7 on Windows Vista

VAZI OKHANDIAR AND ALON

I recently purchased a new laptop. Choosing an operating system was a decision that challenged me for two weeks. Finally, I decided to go with the flow. I ordered my first Vista system. That only was the start of a long and arduous journey and I'm still trying to get my environment and tools back to normal!

When I started the PowerBuilder 11.0 application, I first noticed that pbtutor, the sample application installed with PowerBuilder, wasn't running. The Help files weren't opening. If this article finds you starting out as I did, I hope you will, before too long, deploy your first application on IIS 7.0 ready to run it as a .NET Web form application.

The main issue is that Vista's new User Account Control (UAC) prevents PowerBuilder from connecting to the database and running pbtutor. The UAC is a new security component in Windows Vista. Unlike XP, Vista doesn't automatically grant users, members of the local Administrators group,

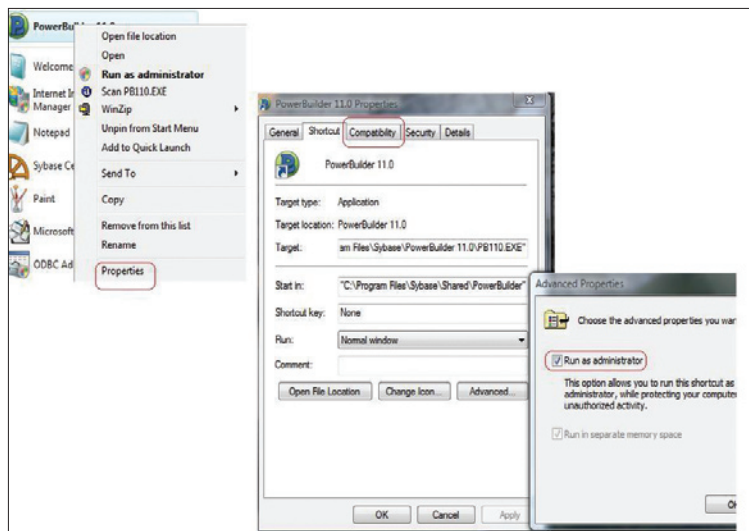


Figure 1

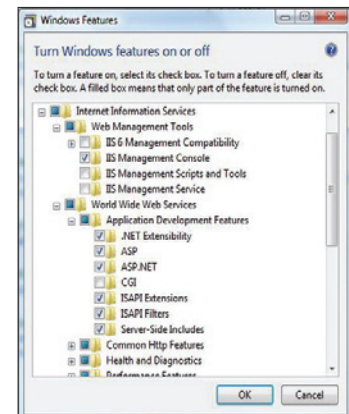


Figure 2

access to all of the Window's resources. For more information and how to modify UAC features, visit <http://technet.microsoft.com/> and search for "Windows Vista User Account Control Step by Step Guide."

The common denominator solution, regardless of the version of Vista you're running, is to set up the application properties to run as an administrator. To set up the PowerBuilder application with administrative privileges, from the All Program menu, right-click on the PowerBuilder Application, then click on the Properties, and select the Compatibility tab on the PowerBuilder 11.0 Properties Window. In the PowerBuilder 11.0 Properties window, click on the Advance... button. This will open up the Advanced Properties window. In the Advanced Properties window, check the box next to the Run as administrator option, and then, click OK (see Figure 1).

The next step is to turn on Internet Information Service (IIS).

Windows Vista ships with IIS 7.0 and the .NET Framework. To turn on the IIS Management

Console and ASP.NET, bring up the Control Panel window, select Programs and Features, and then select Turn Windows features on or off. This will bring up the Windows Feature Window. In the Windows Features Window, check the boxes to enable the IIS Management Console and the Application Development Features (see Figure 2).

To test that IIS is running, click on the Start button for Windows, select All Programs, select Accessories, and then select Run. This will bring up the Run window. In the Run window, type inetmgr and the Internet Information Services Manager window will pop up. In the Internet Information Services (IIS) Manager window, locate the Default Web site listed in the tree view, right-click and select Start from the dropdown menu as shown in Figure 3.

In the Internet browser, type `http://localhost/` and you'll see the IIS 7 welcome screen.

The next step is to give full control to IIS_IUSERS on Sybase and wwwroot directories.

To assign full control to the IIS_USERS in the wwwroot directory, right-click on the wwwroot folder under inetpub, select the properties option from the dropdown menu, and select the Security tab in the wwwroot Properties window. For the Group or user Names, select IIS_IUSERS (see Figure 4).

Once the IIS_IUSERS is selected in the wwwroot Properties window, click on the Edit button. This will display the Permission Dialog window for the IIS_IUSERS. In the Permission Dialog window, check all the boxes and click on the OK button to give full control to the IIS_IUSERS.

Note: If the IIS_IUSERS doesn't appear in the Group or user names sub-window, then click on the Add button on the Permission for wwwroot window. In the Select Users or Groups window, click on the Advance button, type IIS_IUSERS in the Name field, and click on the Find Now button to locate IIS_IUSERS group name. Select the IIS_IUSERS from the list and click on the OK button. Repeat the same steps to give full control to IIS_USERS to the Sybase directory (see Figure 5).

The next step is to ensure that the Web controls are available for the PowerBuilder application so that you can display the PowerBuilder application in the Internet Browser. You may download the Web controls from Microsoft at `http://www.asp.net/downloads/archived/ie-Web-controls/`. Execute `C:\Program Files\IE Web Controls\Build.bat` to install Internet Explorer (IE) Web Controls. Create `\Webctrl_client\1_0` directory under `C:\In-`

`etPub\wwwroot` and copy everything from `C:\Program Files\IE Web Controls` to the new directory.

Next, create an application pool for the PowerBuilder Web Form applications.

PowerBuilder uses the classic managed pipeline where ASP.NET runs as an ISAPI (Internet server API) extension. You should, however, provide an isolation level to avoid affecting other applications that might run in an integrated mode, where ASP.NET request processing integrates directly into the IIS7 request-processing pipeline; Start Inetmgr, right-click on the application pools and select add an application pool, provide it with a unique name, such as PB WebForms, set

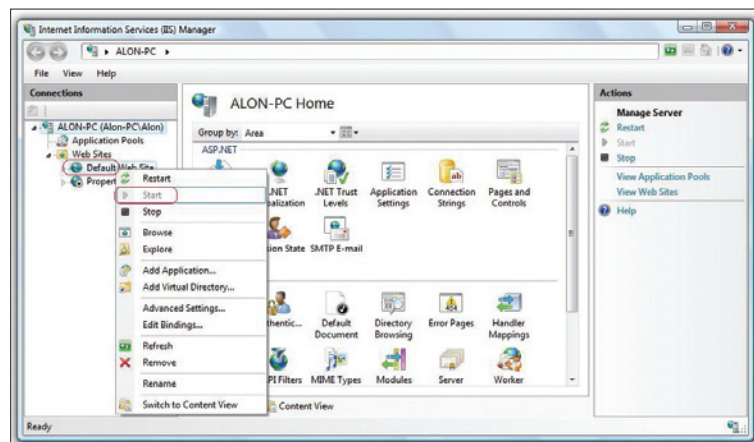


Figure 3

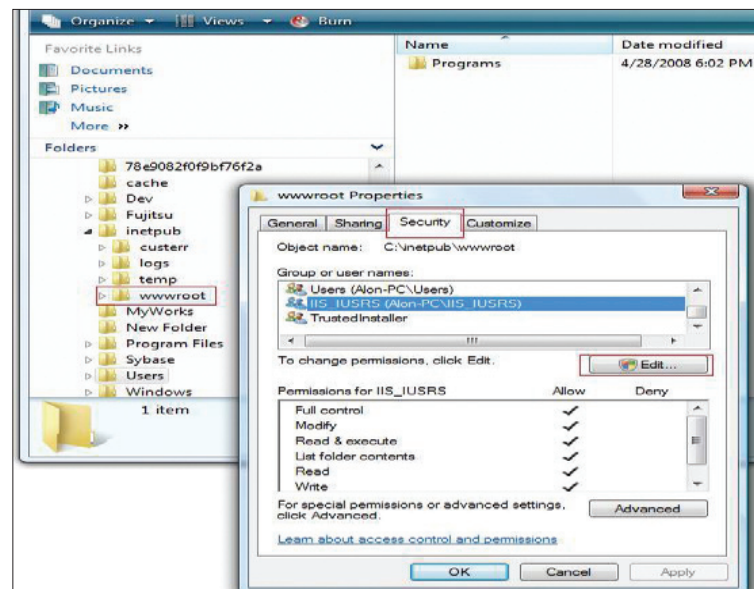


Figure 4

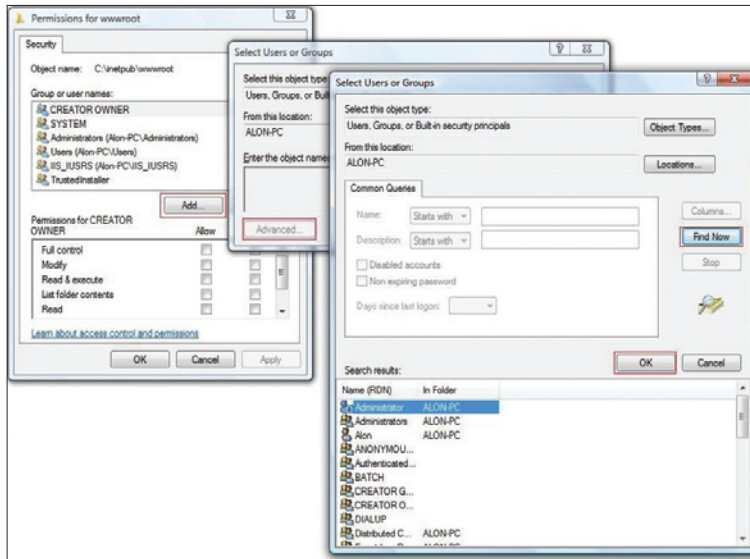


Figure 5

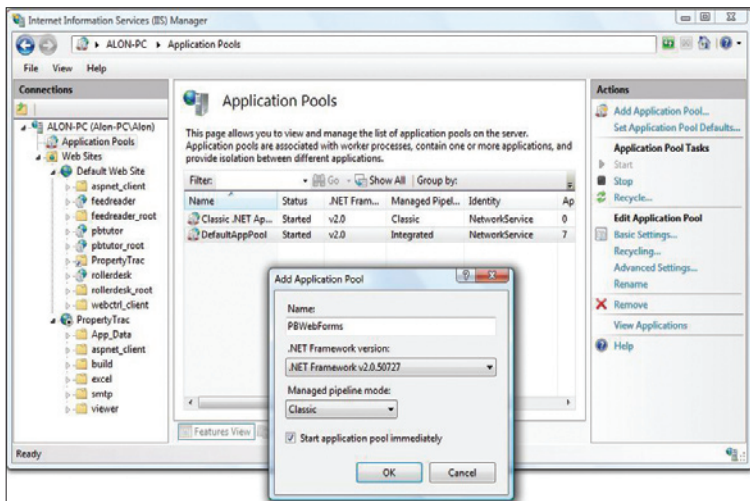


Figure 6

Managed Pipeline Mode to Classic, and click OK (see Figure 6). Later you will assign your application to this pool. To get more information on the difference between classic and integrated modes, refer to the MSDN Web site (<http://blogs.msdn.com/tmarq/archive/2007/08/30/iis-7-0-asp-net-pipelines-modules-handlers-and-preconditions.aspx>).

Now you're ready to create a new .NET Web Form application target in PowerBuilder.

You'll use the pbtutor application that ships with PowerBuilder. From the PowerBuilder IDE's File menu, select File new, and click on new .NET Web

Forms application, and click on the OK button.

This will open a Create the Application window. In the Create the Application window, select the option, Use the library and application object from an existing target as the first dialog of the wizard.

Then, continue to click on the Next button, accepting all defaults, until the Specify Resource Files/Directories Window is displayed. Click on the Add Files button. Locate the c:\Program Files\Sybase\PowerBuilder 11.x\tutorial\ folder, and select the image files. Click on the OK button when done.

Continue to click on Next, selecting all default values, until the Specify Deployment Options window is displayed.

In the Specify Deployment Options window, select directly deploy to IIS option and type localhost in the IIS server address field. When done, click the Next button and then click the Finish button on the last dialog. A new Web form target will appear in the workspace.

The next step is to deploy the application. Right-click on the pbtutor Web form target and select deploy from the dropdown menu (see Figure 7).

After you have successfully deployed pbtutor, you will find the pbtutor Web app under the Web sites in the IIS Manager (inetmgr). Now you're ready to run the pbtutor application.

In the Internet Browser, type <http://localhost/pbtutor> and you'll see this window:

Add this Web app to the application pool you created earlier.

Start IIS Manager (inetmgr), go to the Web Site

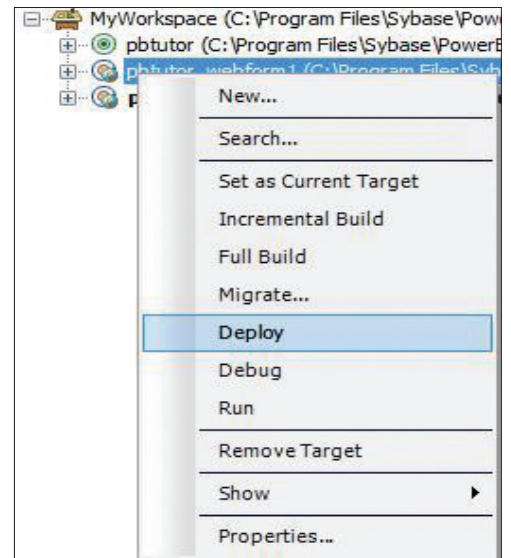


Figure 7

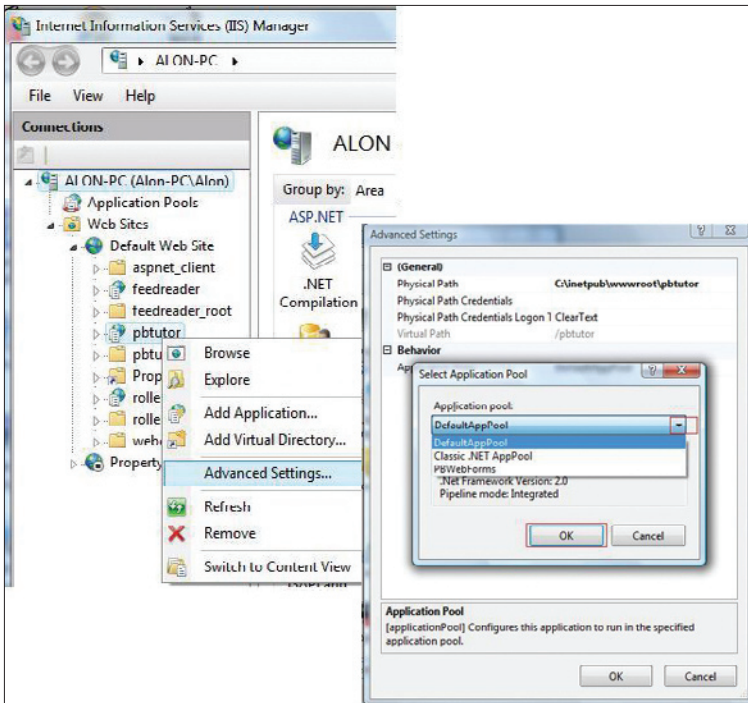


Figure 8

folder, locate the pbtutor Web application, right-click on the pbtutor, and select Advanced Settings.

In the Advanced Settings Dialog, select the PB WebForms for Application pool and click on the OK button to close (see Figure 8).

As you continue to use PowerBuilder 11, you may require assistance with the PowerBuilder help files. You may be unable to open these files, which require the Windows Help (WinHlp32.exe) as you may be accustomed to in the previous version of Windows. Microsoft has decided not to include WinHlp32.exe as a component of the operating system. It is, however, available as a download from the Microsoft Download Center <http://go.microsoft.com/fwlink/?LinkID=82148>.

To use the PowerBuilder help file, download WinHlp32.exe from the Microsoft site and install it

on your computer.

Create a new text file. Type in the following

On 64-bit Windows:

Save the file with .reg

extension, and run it by double-clicking on the file. This article was written to demonstrate the ease of use of PowerBuilder in deploying Web Form .NET applications to IIS in Vista. Indeed, once you've enabled the .NET environment, configured security, enabled Web controls, and set up the application pool, the actual deployment is easy and straight forward. We hope you will now be ready to consider greater challenges, which will surely surface, once you start to deploy your own application. ■

About the Author

Vazi Okhandiar and Alon Baadani

have collaborated in compiling this article. Vazi Okhandiar is the vice-president of technology for mLogica, Inc. For over 15 years, she has been architecting, designing, and managing PowerBuilder-enabled applications. She previously worked for EDS and Computer Science Corporations (CSC) and holds a MBA and BSEE from UC, Irvine and a MS computer science from IIT, Chicago. Alon Baadani, as the principal of ABM Consulting, Inc. has been developing PowerBuilder applications for both large corporations and small businesses over the last 12 years. Currently he is developing applications for RES-Q HealthCare Systems, an industry leader with presence in the U.S., Canada, and the EU.

vazi.okhandiar@mlogica.com

alon@abmssoftware.net

“Once you’ve enabled the .NET environment, configured security, enabled Web controls, and set up the application pool, the actual deployment is easy”