Chapter 2

Deployment of MVC Applications

Create PDSA Runtime License

To deploy a PDSA .NET Productivity Framework application, you must create a PDSA Runtime License. Please refer to Chapter 0 – Deployment Introduction for instructions on how to create a license. Other than the PDSA Runtime License, deploying a PDSA Framework MVC application is just like deploying any other MVC application.

Add Code to Application Start

Ensure the following code is in the Application_Start event procedure in your Global.asax file.

```
protected void Application_Start(object sender, EventArgs e)
{
  PDSA.Common.PDSACheckAssembly.SetEntryAssembly(
   System.Reflection.Assembly.GetExecutingAssembly());
}
```

This will initialize the license code with the main executing assembly. Without this line of code, your application will not run.

Publish Your Web Application

The best way to deploy an MVC project to a QA, Development or Production server is to use the Publish Web dialog from the Build menu of Visual Studio.

Bring up your MVC project in Visual Studio.

Click on your web project in the Solution Explorer window.

Select Build | Publish <NAME OF APP> from the VS Menu

Choose File System and enter the name of a folder somewhere on your hard drive as shown in Figure 1. In the screen shot you see a target location of D:\Samples\SampleProject. This means all of the files for your website, including the PDSA DLLs will be placed into this folder.

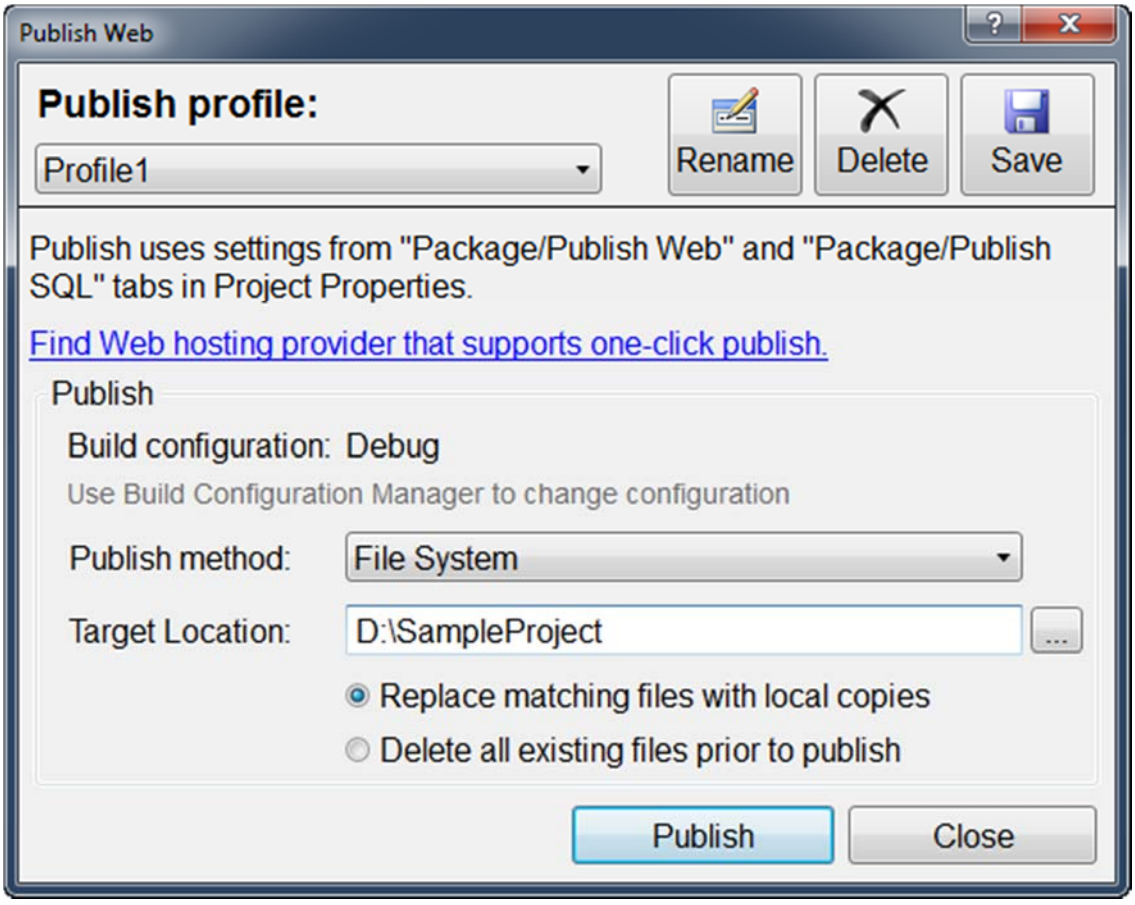

Figure 1: Publish Web dialog will help you deploy your project.

You can now copy all the folders under D:\Samples\SampleProject folder to your web server.

Make sure the license file is in the root of your web site.

Configuration Settings

Below are the things you should check prior to deploying an ASP.NET Web application developed with the PDSA .NET Productivity Framework.

First make sure you build with the **Release** option set and not debug.

Web.Config

Turn on the appropriate Log providers to handle logging and exception handling

Verify the SMTP server settings are correct (if being used)

Encrypt Connection Strings if required using the normal ASP.NET utility provided by Microsoft.

Change all connection strings for your correct development, QA or production database

Set the following attribute in the <system.web> element

<compilation debug="false" ... />

Ensure the <customErrors ...> is set correctly. The **mode** attribute should be either "On" or "RemoteOnly". Be sure the **defaultRedirect** attribute points at a valid page for any errors.

```
<customErrors mode="RemoteOnly"
   defaultRedirect="~/Unsecured/YourErrorPage.aspx" />
```

Set any Trace attributes off

```
<trace enabled="false" requestLimit="10" pageOutput="false"
   traceMode="SortByTime" localOnly="true" />
```

Set the appropriate <sessionState …/> attributes. If you are using a web farm, you will either need to set it to StateServer or SqlServer.

Summary

In this chapter you learned the different settings you should modify when you are ready to move your .NET application to production.