

Contents

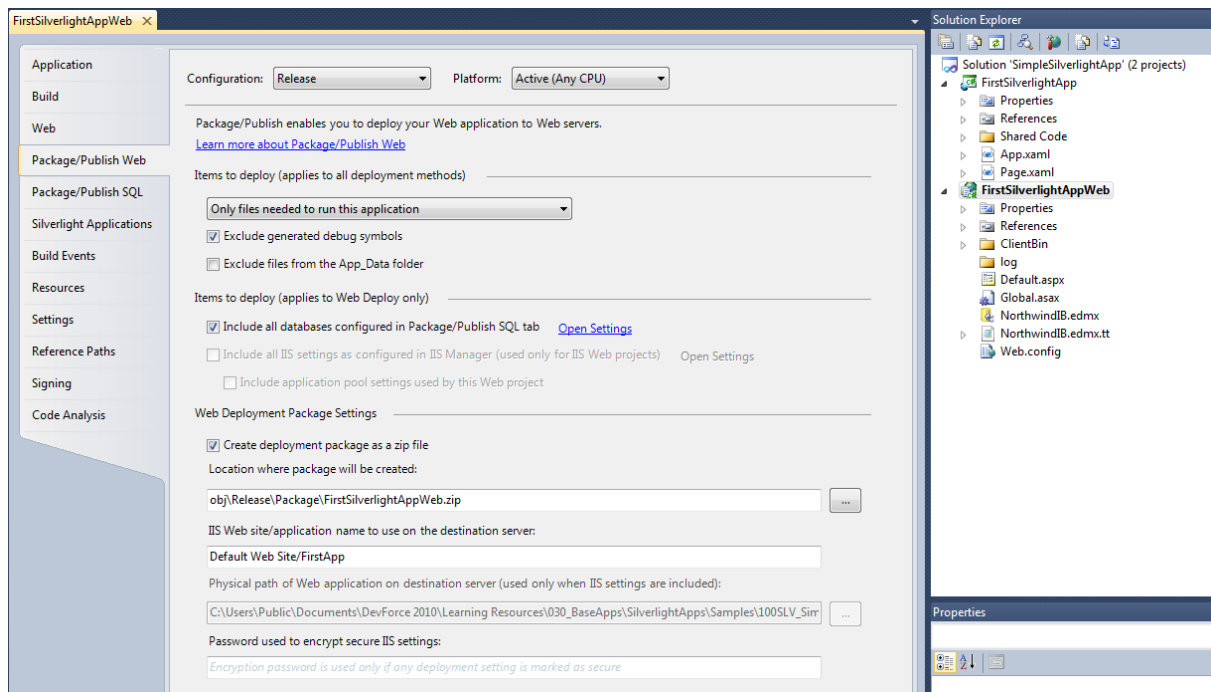
- [Step 1 - Settings](#)
- [Step 2 – One-Click Publish Wizard](#)
- [Step 3 – DevForce Changes](#)
 - [Log file](#)
 - [Cleanup](#)
- [Next Steps](#)
 - [Creating a Deployment Package](#)
- [Resources](#)

Once you're ready to [deploy](#) your Silverlight application to a web server, you can use the **Package/Publish** feature in Visual Studio to easily accomplish this. Microsoft has a thorough [overview](#), which we encourage you to read. Here we'll briefly discuss the steps and any DevForce-specific requirements.

Although this walk through is for a Silverlight application, most steps also apply to any IIS deployment of the *EntityServier*.

Step 1 - Settings

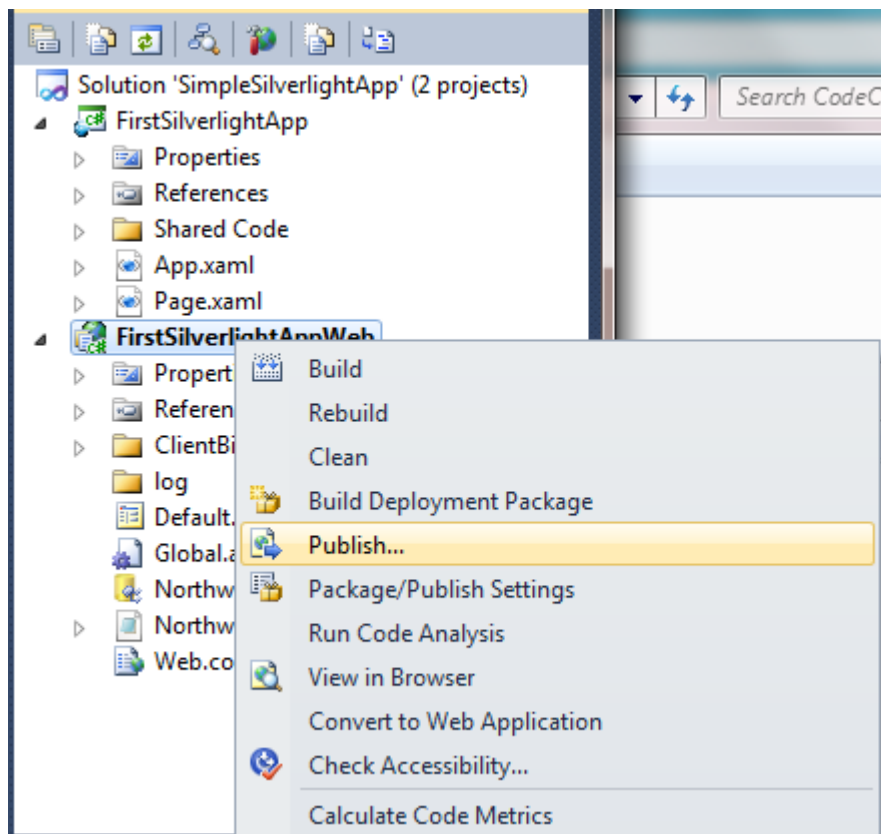
Access the **Package/Publish Web** settings tab from the project properties for the web application.

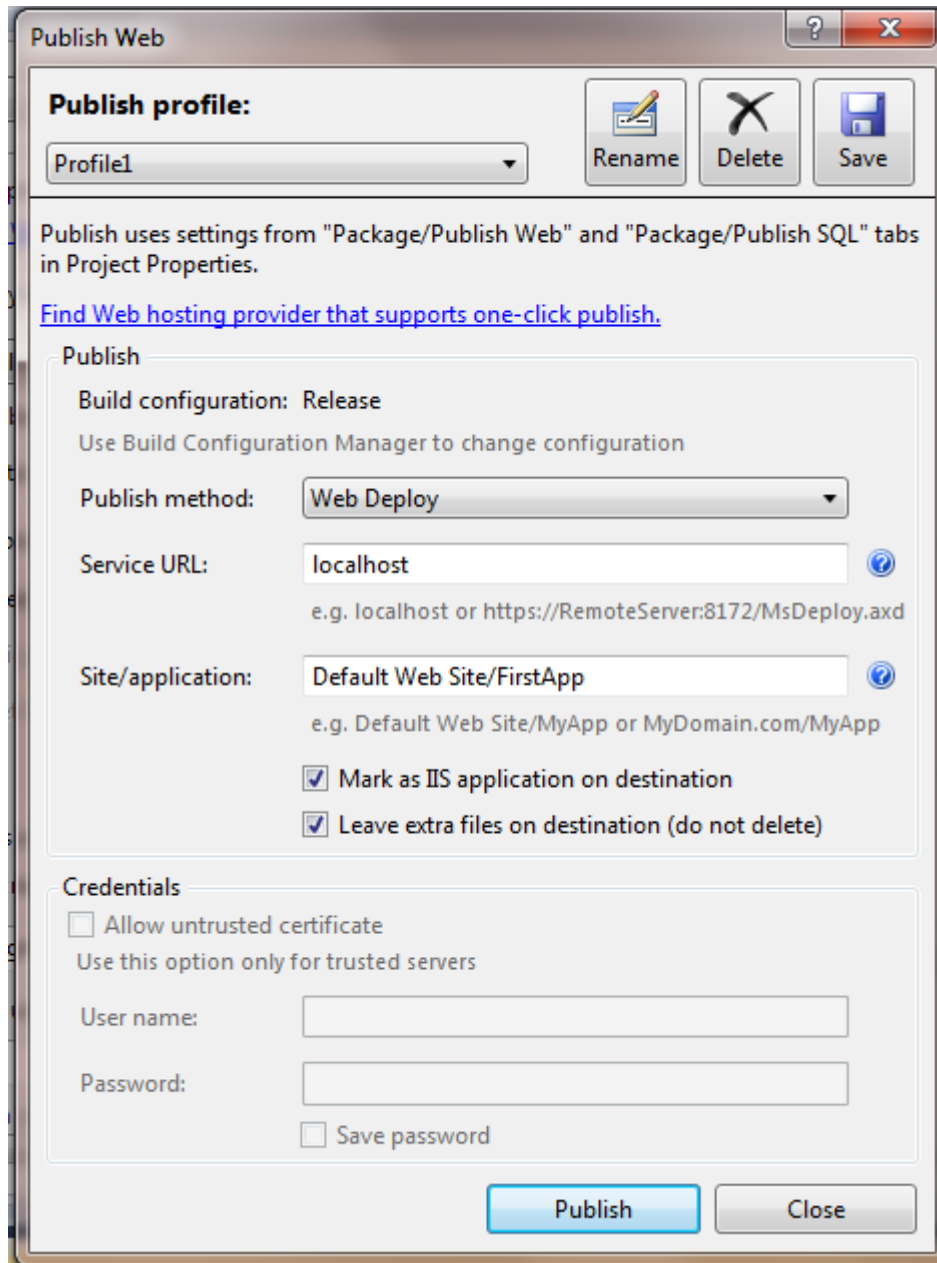


Here we're choosing to deploy "Only files needed to run this application". This will include .dlls, .config, .svc, .asax, .aspx and other non-code files.

Step 2 – One-Click Publish Wizard

Once settings have been created you're ready to publish.

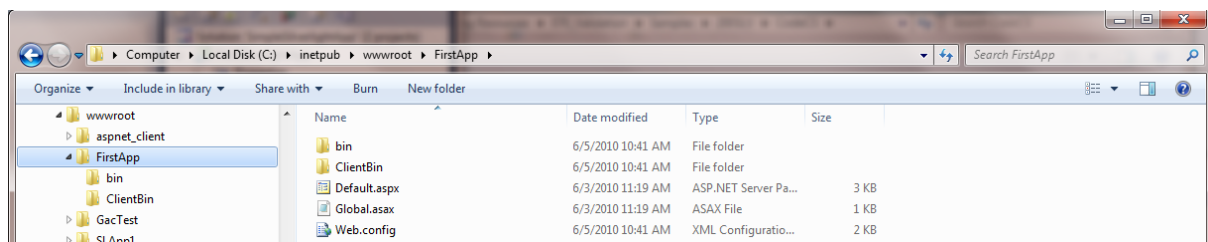




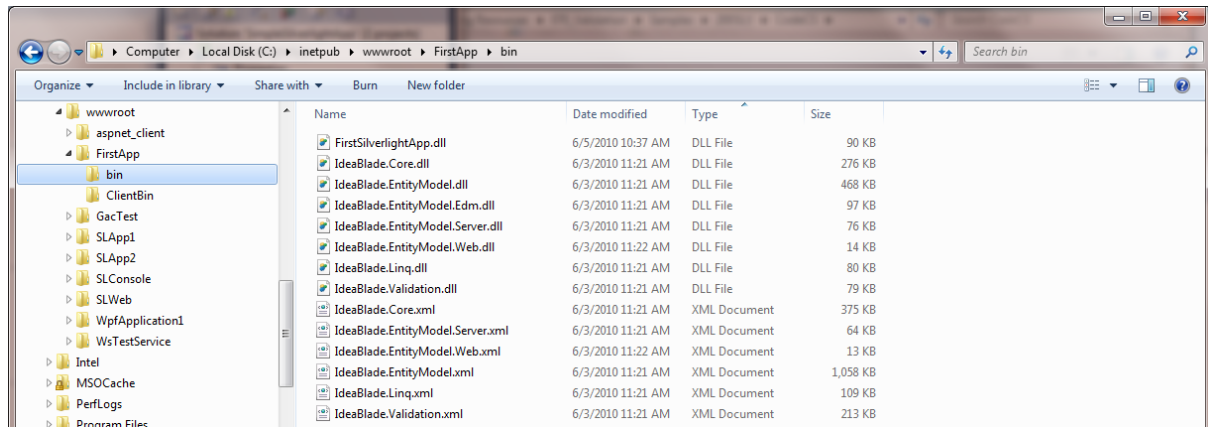
Here we're using a simple "one-click" Web deployment to the same computer on which Visual Studio is installed. We've also set the application name to "FirstApp" and asked IIS to mark the virtual directory created as an IIS application.

After hitting the **Publish** button we can check what's been published.

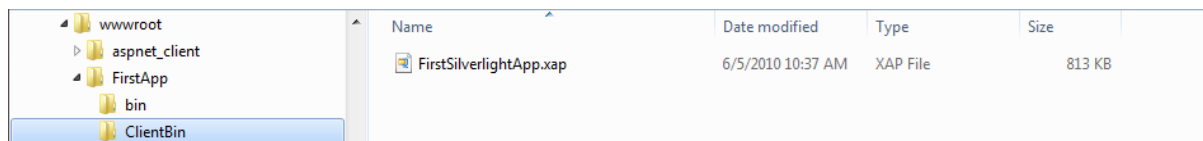
An application folder has been created with the required files:



Assemblies are in bin:



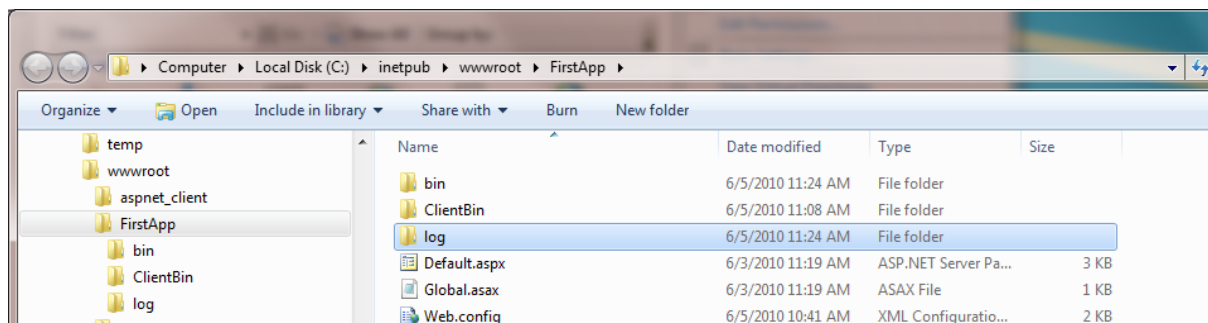
The XAP is in ClientBin:



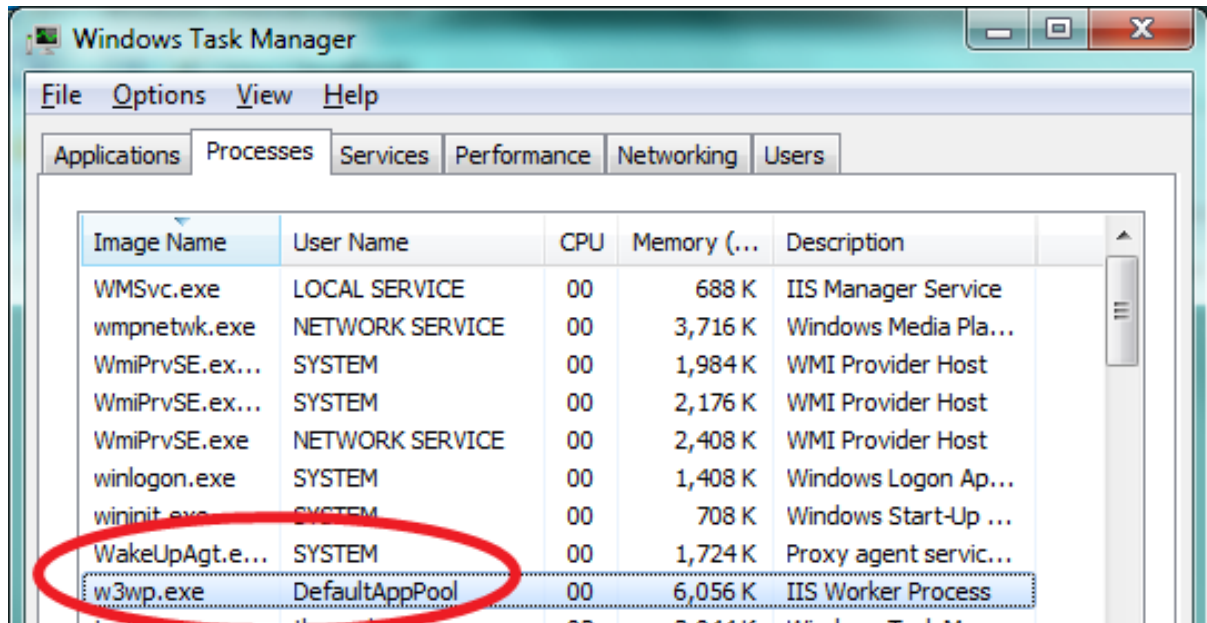
Step 3 – DevForce Changes

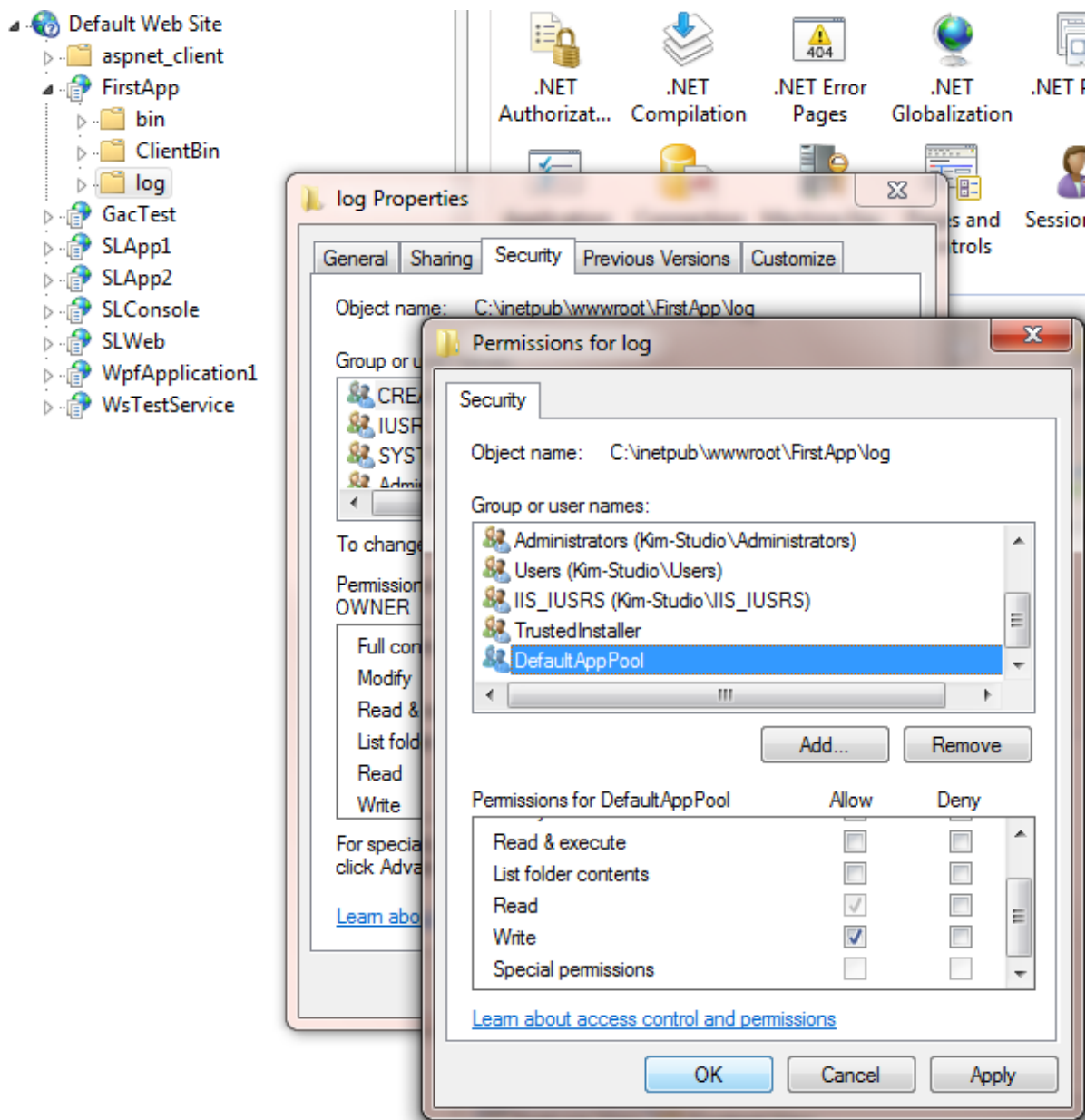
Log file

Note that a log subfolder was not automatically created by the wizard. Unless you've changed the logFile location in your web.config file DevForce will by default try to write its debugLog to the log folder. We can easily create a log folder manually and grant the appropriate permissions.



Write permissions are required since the log file will be created and appended to. Here the application runs under the identity of the application pool, *DefaultAppPool*, so we grant this account write permissions to the log folder.





See the [security](#) topic for information on securing the log file and folder.

Cleanup

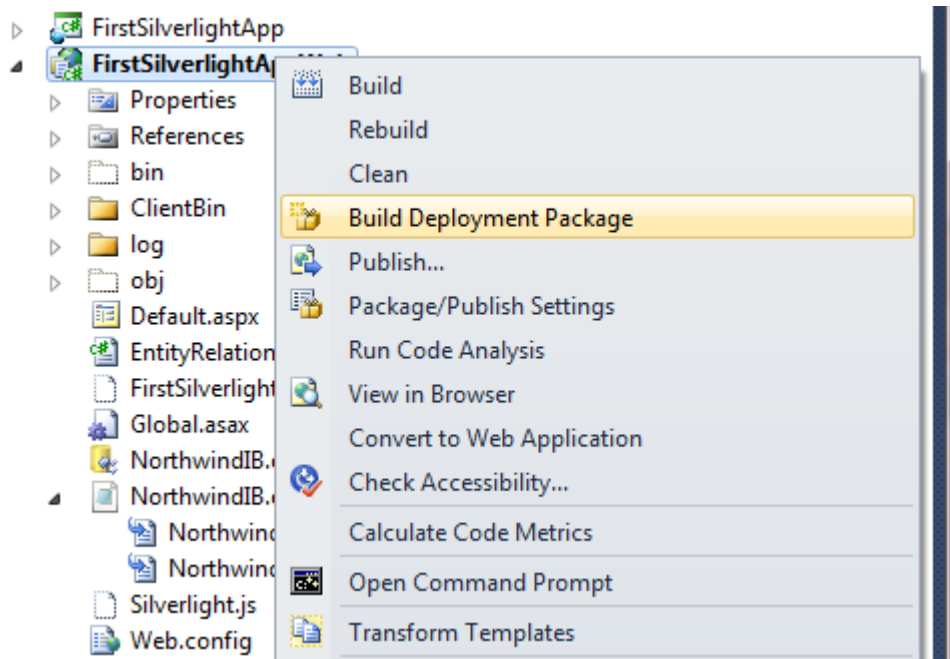
You may have noticed the .xml files in the bin folder. These are Intellisense files for the DevForce assemblies and are not needed at run time. You can delete them.

You can also extend the web publishing pipeline to remove these files from the package. See the resources below for more information.

Next Steps

Creating a Deployment Package

You can also use Visual Studio to create a deployment package.



Visual Studio will create a package, along with a sample script and other files:

Name	Date modified	Type	Size
PackageTmp	6/17/2010 3:18 PM	File folder	
FirstSilverlightAppWeb.deploy.cmd	6/17/2010 3:20 PM	Windows Comma...	11 KB
FirstSilverlightAppWeb.deploy-readme.txt	6/17/2010 3:20 PM	Text Document	4 KB
FirstSilverlightAppWeb.SetParameters.xml	6/17/2010 3:20 PM	XML Document	1 KB
FirstSilverlightAppWeb.SourceManifest.xml	6/17/2010 3:20 PM	XML Document	1 KB
FirstSilverlightAppWeb.zip	6/17/2010 3:20 PM	WinZip File	2,304 KB

The package can then be deployed via the command line with Web Deploy (MSDeploy) or imported directly into IIS Manager.

Resources

- [How to: Deploy a Web Application Project Using One-Click Publish and Web Deploy](#)
- [Automating Deployment with Microsoft Web Deploy](#)
- [Deploying Services with Web Deploy](#)
- [Web Packaging: Installing Web Packages using Command Line](#)