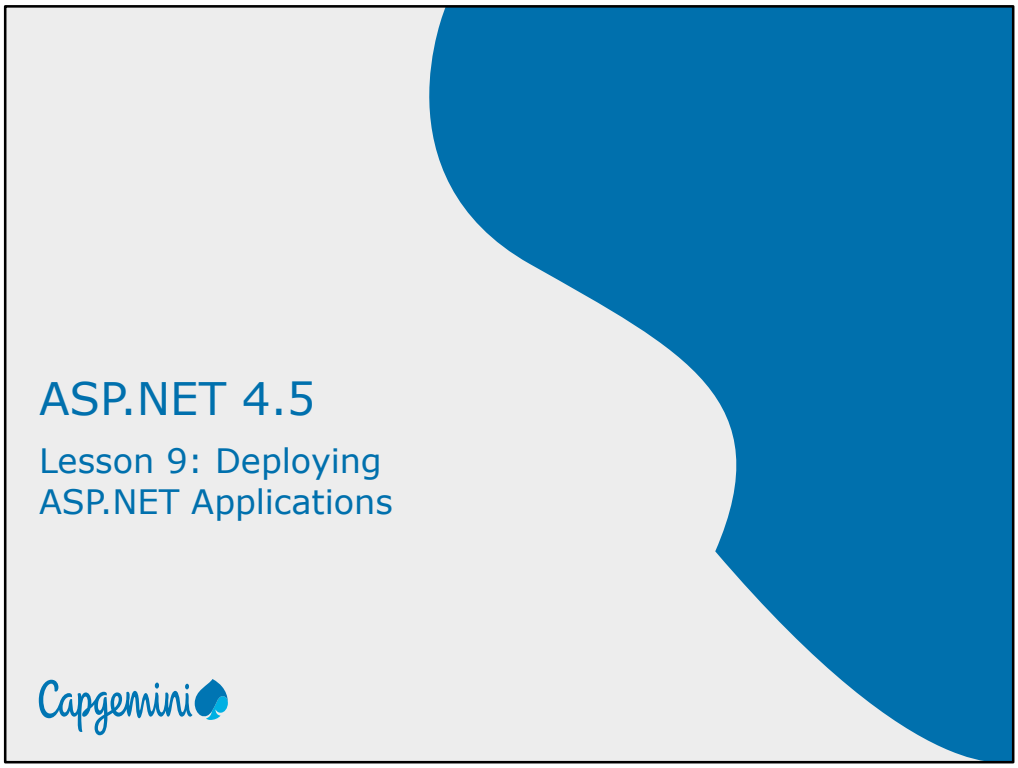

ASP.NET 4.5
Lesson 9: Deploying
ASP.NET Applications
Capgemini

Lesson Objectives

- In this lesson, you will learn:
  - Deployment options for ASP.Net application

9.!: What to Deploy?

An Insight

- ASP.NET application has lot of pieces, which are required for the application to function properly:
  - . aspx page and the code-behind file
  - User controls,
  - .html files
  - Image files
  - ASP.Net system folders
  - .js files
  - Configuration files
  - .Net components and Compiled assemblies

What to Deploy?

Packaging and deploying an ASP.Net application are most often ignored. In this lesson, we will take a look at packaging and deploying an ASP.Net application.

It is very important to package and deploy applications so that the end user finds it convenient to install the products in their environment.

In ASP.Net 2.0, the packaging and deployment process is simpler than previous versions of ASP.Net.

A typical ASP.Net application contains a lot of pieces that are required for the application to function properly. Some of the items are potentially part of ASP.Net application and need deployment consideration, namely:

.aspx page and the code behind file
User controls
.html files
Image files .jpg or .gif
ASP.Net system folders such as App_Code and App_Themes
.js files
Configuration files, web.config
.Net components and Compiled assemblies

9.2: Steps to be taken before Deploying

Sequence of Steps

- Before deploying, ensure the following:
  - Turn off- debugging through web.config.

```
<configuration>
    <system.web>
            <compilation debug="false" />
  </system.web>
</configuration>
```

  - Build application in release mode.

Steps to be taken before Deploying:
Before deploying an application, some steps have to be taken by a ASP.Net developer.
The first step is to switch off debugging in the web.config file.

You can do this by setting the debug attribute in the <compilation> element to false.

```
<configuration>
        <system.web>
  <compilation debug="false" />
 </system.web>
</configuration>
```

If this attribute is true, then debug symbols are inserted into the code which degrades the applications performance.

The next step is to build the application in release mode.

You can accomplish this by changing the Active Solution configuration from Debug to Release through the drop down list in the Visual studio menu.

9.3: Methods to Deploy - XCopy

The Usage

- You can use XCopy to copy .Net assemblies
  - XCopy is command line utility to copy and paste files that you want to move.
  - XCopy enables you to move files, directories, and even entire drives from one to another
  - Example:

```
xcopy c:\WebSites\MyWebSite z:\Websites\ /f /e /k /h
```

Methods to Deploy:
XCopy:
Once a Web Application is packaged you need to deploy it. There are options on how to deploy the Web Application. We will first take a look at XCopy command. The XCopy functionality is easy to use because in .NET the application compiles down to assembly. The .Net assembly has all the information within itself. Hence XCopy functionality is possible.
XCopy is command line utility to copy and paste files you want to move.
It enables you to move files, directories, and even entire drives from one to another.
Following is an example of using XCopy:

```
xcopy c:\WebSites\MyWebSite z:\Websites\ /f /e /k /h
```

9.4: Methods to Deploy - VS copy

The Usage

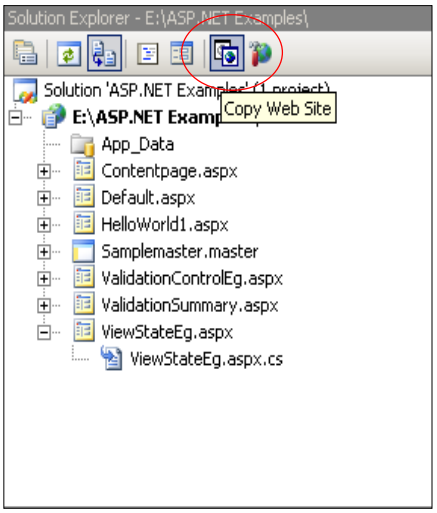
- Using the GUI provided by Visual Studio 2012, you can:
  - Deploy Web sites to Remote Server or some development server.
  - Open the Copy web site dialog from Visual Studio 2012 Server Explorer, or
  - Open the dialog from Websites → Copy Web site from Visual Studio menu.

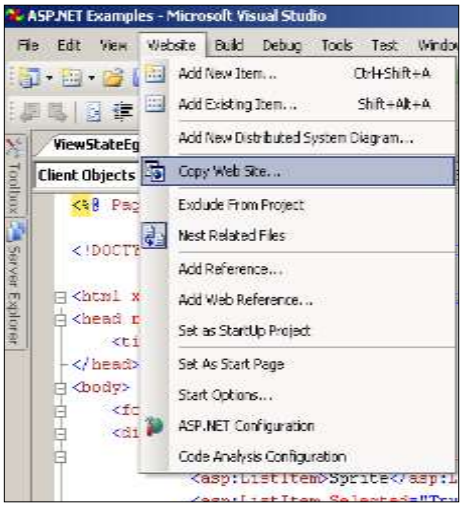
Methods to Deploy:
VS copy:
The next option to deploy your Web Site is using the GUI provided by Visual Studio 2005.
This GUI enables you to copy Web sites from your development server to either the same server or remote server.
The GUI can be opened in two ways:

By selecting Copy Web Site icon from the Visual Studio Server Explorer.
By selecting Websites → Copy Website from the Visual Studio menu

9.5: Copy Web Site Dialog

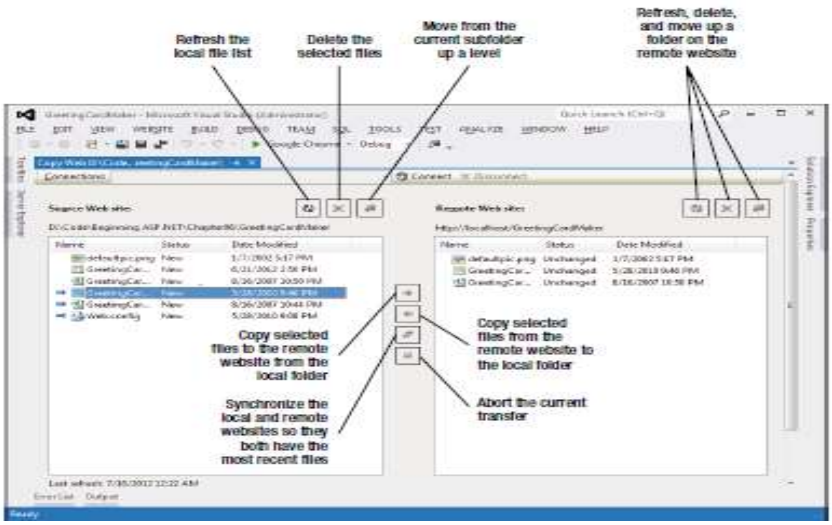
The Navigation





9.6: The CopyWeb site Dialog Box

The Navigation (Contd…)



The CopyWeb site Dialog Box:
From this GUI, you can click the Connect To a Remote Server button. After doing this, the Open Web Site dialog pops up, which allows you to specify the destination for connection. The options available are File system, Local IIS, FTP site, and Remote Site.
After connecting to the server you can copy contents of your Web application.

9.7: Deploying a Pre-Compiled Application

The Process

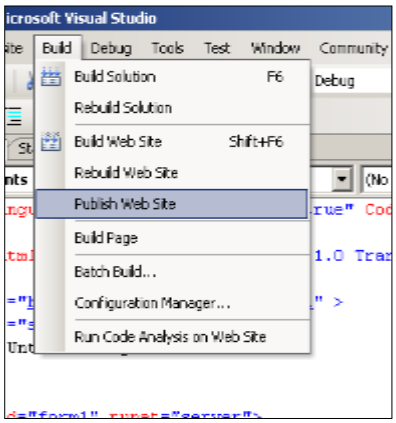
- A pre-compiled application has all the page code compiled into one DLL, which makes it difficult for modification
  - You can use aspnet_compiler utility to precompile and deploy the application
  - Alternatively, you can choose Visual Studio for the precompilation and deployment process

Deploying a Pre-Compiled Application:
You can also deploy a Web Application from one location to another wherein the application is pre-compiled.
During pre-compilation, all the pages in the Web Application are built and compiled into a single application DLL and some placeholder files. These files can be deployed together to another server and executed from there.
Since the code is placed into one DLL, it makes it more difficult for your code to be stolen or changed.
You can use the aspnet_compiler utility to pre-compile your web application and deploy the web application. Alternatively, you can also use Visual Studio 2012 to accomplish the pre-compilation and deployment process.

9.8: Using Visual Studio Menu

The Navigation

- Open the dialog box, by selecting Build → Publish Web Site

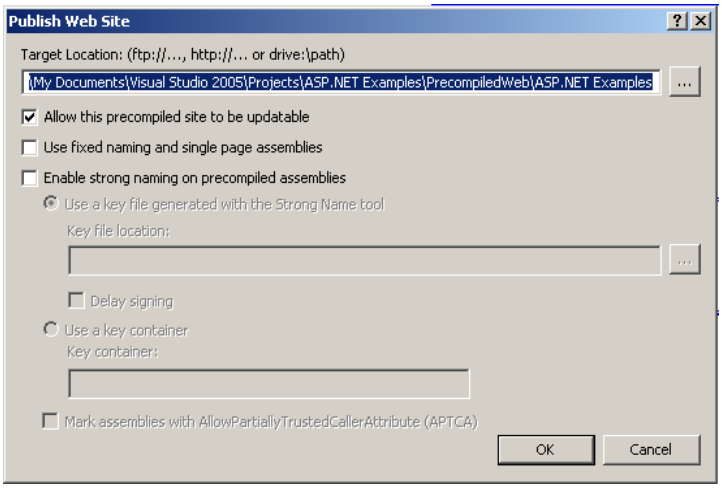


Using Visual Studio Menu:
To use the Visual Studio option, first open the Web site for deployment and get it ready deployment by turning off the debugging capabilities as mentioned earlier in this lesson. Open the dialog box by selecting Build→ Publish Web site option from the menu. The dialog box can be seen on the next slide with the following UI elements.

Target Location: It specifies the output path for the precompilation process. The results of precompilation are written to the specified location, overwriting (without warning) any existing content in that location. You can specify an FTP path, HTTP path, or a physical drive location.
Allow this precompiled site to be updatable: It specifies that the content of .aspx pages are not compiled into an assembly. Instead, the markup is left as-is, allowing you to change HTML and client-side functionality after precompiling the Web site. Selecting this check box is equivalent to adding the -u option to the aspnet_compiler.exe command.
Use fixed naming and single page assemblies: It specifies that batch builds are turned off during precompilation in order to generate assemblies with fixed names. Themes and skin files will continue to be compiled to a single assembly. This option is not available for in-place compilation.

9.8: Using Visual Studio Menu

The Navigation (Contd..)



Publish Web Site Dialog (contd.):

Enable strong naming on precompiled assemblies: It specifies that the generated assemblies are strongly named by using a key file or key container to encode the assemblies and ensure that they have not been tampered with. After you select this check box, you can do the following:

Specify the location of a key file to use to sign the assemblies. If you use a key file, you can select Delay signing, which signs the assembly in two stages – first with the public key file, and then with a private key file that is specified later during a call to the aspnet_compiler.exe command.

Specify the location of a key container from the system's cryptographic service provider (CSP) to use to name the assemblies.

Specify whether to mark the assembly with the AllowPartiallyTrustedCallers property, which allows strongly named assemblies to be called by partially trusted code. Without this declaration, only fully trusted code can use such assemblies.

When you are ready to deploy, click OK and then the open application is built and published.

9.8: Demo on Deployment in ASP.NET

Demo on Deployment in ASP.NET

- Trying out the following Deployment Options:
  - XCopy
  - VS Copy
  - Copy Web Site
  - PreCompilation
  - Publishing the Web Site

Web Farms
Some applications run on web farms, a group of server computers that share the responsibility of handling requests. Usually web farms are reserved for high-powered web applications that need to be able to handle heavy loads, because multiple computers can deal with more simultaneous surfers than can a single web server. However, web farms are overkill for many small and midsize websites.

Web hosting companies use web farms to host multiple websites. For example, your website might be running on more than one web server, but each of these web servers might also host multiple websites. This provides a flexible deployment model that lets different web applications share resources.

9.9: Deployment: Best Practices and Guidelines

Best Practices

- Avoid XCOPY under heavy load
  - For production environments, remove a server from rotation, stop IIS, perform the XCOPY update, restart IIS, and then put the server back into rotation.
- Consider pre-compiling pages
  - By precompiled pages, users do not have to experience the batch compile of your ASP.NET files. It will increase the performance that your users will experience.

9.9: Deployment: Best Practices and Guidelines

Best Practices (Contd…)

- Disable tracing and debugging before deploying the application
  - Set debug attribute to false and trace enabled to false as shown below:

```
<configuration>
<system.web>
<trace enabled="false" pageOutput="false" />
<compilation debug="false" />
</system.web>
</configuration>
```

Summary

- In this lesson, you have learnt :
  - Deployment options for ASP.NET application

Review Question

- Question 1: Which of the following will help me in precompiling and deploying an ASP.NET application?
  - XCopy
  - VS Copy
  - aspnet_compiler
  - Copy Web Site
- Question 2: Which of the following is a command line utility to copy and paste files that you want to move?
  - XCopy
  - VS Copy
  - aspnet_compiler
  - Copy Web Site