D

Understanding IIS 7.5

Internet Information Service 7.5 (IIS 7.5), the latest version of Microsoft's information service , is one of the most powerful information service introduced by Microsoft since IIS 1.0. IIS 7.5 provides an accessible and extensible architecture to develop and reliably host Web applications and services. However, this version does not introduce any significant changes in its architecture as compared to the last version (IIS 7.0). As a Web server platform, IIS 7.5 provides Web pages and Web services as requested by a browser or other applications. IIS 7.5 is shipped as a part of Windows 7 and Windows Server 2008 R2 release.

The first version of IIS Web server, IIS 1.0, was shipped with Windows NT. Later, Microsoft introduced IIS 5.0. This version was very popular, but unfortunately, it became a victim of the NIMDA worm. Microsoft then introduced IIS 6.0. This new version was highly reliable, secure, and fast-processing Web server. However, there was one drawback with IIS 6.0, that is, it had monolithic implementation of all the functionalities included in IIS 6.0. Monolithic implementation implies that the user had to implement either all the functionalities or none. In other words, users were not able to use a specific functionality of IIS 6.0 as per their requirements. Therefore, when IIS 7.0 was introduced, all the drawbacks of the earlier versions were addressed. IIS 7.0 implemented all the features in modules, wherein each module can be used independently. Moreover, IIS 7.0 includes some features of IIS 6.0, such as reliability, speed, and security. These features help administrators and developers to implement only the features they need. In addition, enhanced security and the modular approach in IIS 7.0 make it faster as compared to the previous versions of IIS.

The latest version of IIS, IIS 7.5, is a minor upgrade of IIS 7.0 and basically consists of features that were initially available as separate add-ons for IIS 7.0. IIS 7.5 incorporates improved management and deployment options as compared to IIS 7.0. In addition, the new pluggable nature of IIS 7.5 makes it more secure, reliable, and scalable Web server that provides an easy to manage platform to develop and host Web applications and services.

In this appendix, you learn about the architecture of IIS 7.5. You also learn about the new features of IIS 7.5. In addition, you learn about the IIS Manager interface, how to deploy a Web site on IIS 7.5, and perform troubleshooting in IIS 7.5.

Now, let's start by exploring the architecture of IIS 7.5.

Introducing the IIS 7.5 Architecture

As stated earlier, IIS 7.5 includes many significant changes as compare to IIS 7.0. In addition, IIS 7.5 is more extensible than IIS 7.0 and has a modular design as it allows third-party features to be directly integrated into IIS. Furthermore, the primary design goal of IIS 7.5 is to enable the Web administrators to easily develop, deploy, and manage Web applications. As compared to the previous versions, IIS 7.5 provides the following features:

- Allows you to select which module you want on a server
- Allows you to customize and extend server capabilities as per your requirement
- Allows you to replace existing modules or introduce new features by using custom modules

These features make the architecture of IIS 7.5 more secure and easy to administer than before. It reduces the server attack surface and the memory footprint by removing unnecessary modules. Server attack surface is generally known as the code running on the server. Memory footprint is the memory that a server worker process uses on a machine. IIS 7.5 also helps you to enable only those features that are necessary for your Web site or Web application, and disable the ones that you do not need.

IIS 7.5 includes a Web server engine in which you can add or remove components (called modules) according to your requirements. There are two types of modules, which are as follows:

Native modules

Managed modules

Now, let's discuss them in detail one by one.

Native Modules

Native modules are assemblies with a .dll extension and are also known as unmanaged modules, as these modules are not developed by using the ASP.NET model. Most of the features available in IIS 7.5 are implemented in an application by using the Native modules. In this appendix, we describe Native modules that are available after complete installation of IIS 7.5. You can remove or replace Native modules with custom modules, depending on the requirements. The different types of Native modules are as follows:

- Hypertext Transfer Protocol (HTTP) modules
- Security modules
- Content modules
- Compression modules
- Caching modules
- Logging and diagnostics modules
- Managed support modules

Now, let's discuss each module in detail.

HTTP Modules

In IIS 7.5, several modules perform tasks specific to HTTP in the request-processing pipeline. HTTP modules are responsible for responding to clients based on information available in header files. The header files are sent from a client to a server to make a request. Table D.1 lists various HTTP modules:

Table D.1: List of HTTP Modules

Module Name	Description
`CustomErrorModule`	Displays configured HTTP error messages when an error status code is set on a response; otherwise, a default error message is displayed.
`HttpRedirectionModule`	Maintains configurable redirection functionality for HTTP requests.
`ProtocolSupportModule`	Transacts protocol-related actions, such as setting response headers. It also redirects headers based on configuration.

Now, let's discuss about Security modules.

Security Modules

In IIS 7.5, there are separate modules for each authentication type and each module performs tasks related to the security of the request-processing pipeline. You can select modules based on the authentication required on a server. Table D.2 lists various Security modules:

Table D.2: List of Security Modules	
Module Name	**Description**
AnonymousAuthenticationModule	Authenticates users by using Anonymous authentication in case no other authentication method succeeds
BasicAuthenticationModule	Authenticates users by using Basic authentication
CertificateMappingAuthenticationModule	Authenticates users by using Certificate Mapping authentication through Active Directory
DigestAuthenticationModule	Authenticates users by using Digest authentication
IISCertificateMappingAuthenticationModule	Authenticates users by using Certificate Mapping authentication
RequestFilteringModule	Authenticates users by using URLScan tasks, such as configuring allowed verbs to limit the type of HTTP requests that are allowed by a Web server and file extensions, setting limits, and scanning for bad character sequences
UrlAuthorizationModule	Authenticates users by using URL authorization
WindowsAuthenticationModule	Authenticates users by using NT LAN Manager (NTLM) integrated authentication
IpRestrictionModule	Prohibits access of a Web site from Internet Protocol version 4 (IPv4) addresses listed in the ipSecurity list in a configuration file

Next, let's discuss about Content modules.

Content Modules

Content modules are responsible for processing requests for static files. These modules return a default page when a client does not specify a resource in a request. Table D.3 lists various types of Content modules:

Table D.3: List of Content Modules	
Module Name	**Description**
CgiModule	Executes the Common Gateway Interface (CGI) procedure to build the response output
DefaultDocumentModule	Returns a default document for requests made to the parent directory
DirectoryListingModule	Displays the content of a directory in the list format
IsapiModule	Hosts the .dll files included in the Internet Server Application Programming Interface (ISAPI) extension
IsapiFilterModule	Supports ISAPI filter DLLs to filter every request and provides the IIS with only those requests that are to be processed
StaticFileModule	Serves static files
FastCgiModule	Supports FastCGI, which provides a high-performance alternative to CGI

Now, let's discuss about Compression modules in IIS 7.5.

Compression Modules

In IIS 7.5, there are two modules used to perform compression in the request-processing pipeline. Table D.4 lists the modules used for compression functionalities:

Table D.4: List of Compression Modules	
Module Name	**Description**
`DynamicCompressionModule`	Compresses responses and applies `gzip` compression transfer coding to the dynamic content to respond to the user's request
`StaticCompressionModule`	Performs pre-compression of static content

Next, let's learn about Caching modules.

Caching Modules

In IIS 7.5, several modules perform tasks related to caching in the request-processing pipeline. Caching improves the performance of a Web site or Web application by storing the processed information (such as Web pages) in the memory on a server and reusing the information in subsequent requests for the same resource. Table D.5 lists the modules used for caching on a server:

Table D.5: List of Caching Modules	
Module Name	**Description**
`FileCacheModule`	Provides the user mode caching for files
`HTTPCacheModule`	Provides the kernel mode and user mode caching in an HTTP request
`TokenCacheModule`	Provides the user mode caching of the user name and Windows security tokens for the authentication types that are password-based
`UriCacheModule`	Provides the user mode caching of URL information

Now, let's discuss about Logging and Diagnostics modules.

Logging and Diagnostics Modules

In IIS 7.5, the Logging module help in loading custom modules to a Web server and passing information to the `HTTP.sys` component. Diagnostics modules keep track of events and report them to the Web server during request processing. Table D.6 lists the `Logging` and `Diagnostics` modules:

Table D.6: List of Logging and Diagnostics Modules	
Module Name	**Description**
`CustomLoggingModule`	Loads custom logging modules
`FailedRequestsTracingModule`	Enables the Failed Request Tracing feature to trace the failed request
`HttpLoggingModule`	Transacts information related to the HTTP request and the processing status to the HTTP.sys component
`RequestMonitorModule`	Follows requests currently executing in worker processes and reports information with Runtime Status and Control Application Programming Interface (RSCA)
`TracingModule`	Keeps track of events and reports them to Microsoft Event Tracing for Windows (ETW)

Next, let's learn about Managed Support modules.

Managed Support Modules

IIS 7.5 provides modules that support managed integration in the IIS request-processing pipeline. Table D.7 lists Managed Support modules:

Table D.7: List of Managed Support Modules	
Module Name	**Description**
`ManagedEngine`	Provides integration of managed code modules in the IIS request-processing pipeline
`ConfigurationValidationModule`	Verifies and validates configuration issues

Now that you have learned about the Native modules and its different types, let's discuss about the Managed modules next.

Managed Modules

In addition to Native modules, IIS 7.5 enables you to use managed code modules to extend the functionality of IIS. Managed modules are of managed types and developed by the .NET assemblies by using the ASP.NET model. Some Managed modules, such as `UrlAuthorization`, have a Native module counterpart that provides a native alternative to the Managed module. It is important to note that Managed modules depend on the `ManagedEngine` module. Table D.8 lists the Managed modules that are available with the complete installation of IIS 7.5:

Table D.8: List of Managed Modules	
Module Name	**Description**
`AnonymousIdentification`	Identifies anonymous users that support anonymous identification, such as the ASP.NET profile
`DefaultAuthentication`	Assures that an authentication object is present in the context of the requesting Web site
`FileAuthorization`	Validates and verifies that a user has permission to access the requested file
`FormsAuthentication`	Maintains authentication by using Forms authentication
`OutputCache`	Enables output caching
`Profile`	Administers users profiles by using the ASP.NET profile, which stores and retrieves the user settings in a data source, such as a database
`RoleManager`	Administers a `RolePrincipal` instance for the current user
`Session`	Maintains a session state that enables storage of data specific to a single client within an application on a server
`UrlAuthorization`	Determines if the current user is permitted to access the requested URL or not, based on the user name or the list of roles of which the user is a member
`UrlMappingsModule`	Maps a real URL to a more user-friendly URL
`WindowsAuthentication`	Sets the identity of a user for an ASP.NET Web application when Windows authentication is enabled

This completes our discussion on Managed modules in IIS 7.5. Next, let's discuss about the new features of IIS 7.5.

Exploring the New Features of IIS 7.5

Most of the new features in IIS 7.5 have been introduced to ease the management tasks of Web administrators through the IIS Manager interface and the command-line based options. Some of the features and functionalities that have either been newly introduced or enhanced in IIS 7.5 are as follows:

- **WebDAV extension and File Transfer Protocol (FTP) publishing service**—Refers to the enhancements made to the WebDav extension and FTP publishing service that enable the Web content creators to publish content. The FTP publishing service and WebDav extension also provide Web administrators various

options for authentication, auditing, and logging. In the earlier versions of IIS, the FTP publishing service and WebDav extension were available as separate extensions. However, support for FTP publishing service and WebDav extension has now been added to IIS 7.5 in all editions of Windows 7.

- **Request Filtering module**—Refers to the built-in security feature that enables Web administrators to restrict or block certain HTTP requests that are suspected to cause harm when they reach the Web server. The Request Filtering module was available as an extension in the earlier versions of IIS. The elements added to the Request Filtering module in IIS 7.5 are as follows:
  - **<alwaysAllowedURLs> element**—Contains a collection of URLs that are allowed by the RequestFiltering module
  - **<alwaysAllowedQueryStrings> element**—Contains a collection of query strings that are allowed by the RequestFiltering module
  - **<denyQueryStringSequences> element**—Contains a collection of query string sequences that are denied by the RequestFiltering module. The <denyQueryStringSequences> element of the RequestFiltering module enables the Web administrators to block any query string they detect as being harmful.
  - **<filteringRules> element**—Contains a collection of custom request filtering rules. The <filteringRules> element helps in creating customized filtering rules that are based upon certain conditions.
- **IIS PowerShell provider**—Refers to a Windows PowerShell snap-in that enables you to perform IIS administrative tasks and manage IIS, ASP.NET, and custom configuration and run-time data. PowerShell is used by Web administrators as it helps in simplifying the administration of websites by scripting common management tasks and automatically executing recurring tasks. In addition to the PowerShell provider, IIS 7.5 offers a collection of low-level commandlets that enable you to manage IIS settings. Furthermore, the collection of task-oriented commandlets in IIS 7.5 offers an easy and simple way to manage Web sites, Web applications, and backup and restore Web server configuration.
- **Configuration logging and tracing**—Refers to the feature that enables you to audit the access to IIS configuration and allows tracking any successful or failed modifications. Any successful or failed modification can be traced by enabling new logs that become available in Event Viewer.
- **Extensibility of IIS functionality and features**—Refers to the design goal of IIS 7.5, which allows the existing functionalities and features of IIS 7.5 to be extended. In other words, it means that you can build or buy any software and then integrate it into IIS 7.5. In addition, IIS 7.5 enables any software to be integrated with it in a way that the software seems to be a part of IIS 7.5.
- **Enhancements in application hosting**—Refers to the feature that enables the hosting of Web applications developed using Hypertext Preprocessor (PHP) and ASP.NET.
- **Enhancements in application pool security**—Refers to the increased security and reliability of applications and services running on IIS 7.5 as now each application pool runs with a unique yet less-privileged identity.
- **Failed request tracing for FastCGI**—Enables PHP developers who use the FastCGI module to implement IIS trace calls within their applications. If any error occurs in an application, then the developers can debug the code during development by using IIS Failed Request Tracing rules.

  Now, let's learn how to install IIS 7.5.

Installing IIS 7.5

To take full advantage of the new features of IIS 7.5 in a Web site, you need to first install and configure the features of IIS 7.5 on your computer. As IIS 7.5 is shipped with Windows 7 and Windows 2008 R2 release; therefore, to install IIS 7.5 one of these operating systems must be installed on your computer.

Perform the following steps to install IIS 7.5:

1. Open the Control Panel window and select the Programs option.
2. Click the Turn Windows feature on or off link in the Programs and Features group, as shown in Figure D.1:

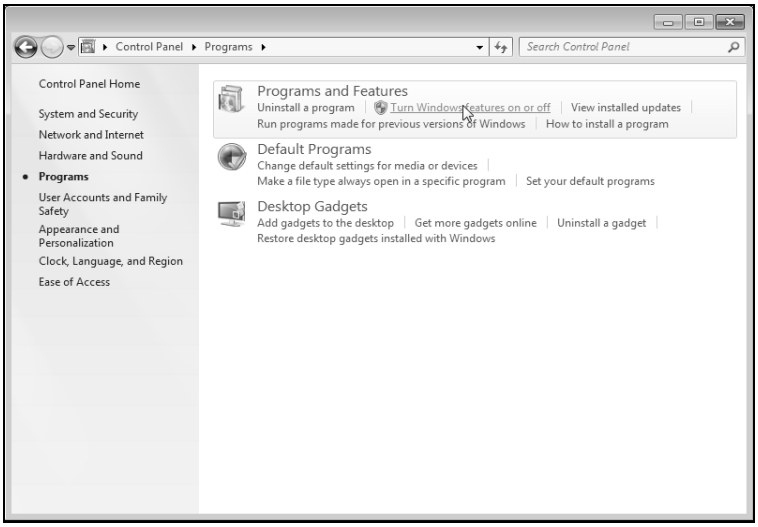

Figure D.1: Showing the Turn Windows Feature On or Off Link

The Windows Features window opens.

3. Expand the folder named Internet Information Services by clicking on the + (plus) sign (Figure D.2).
4. Select all the features and click the OK button, as shown in Figure D.2:

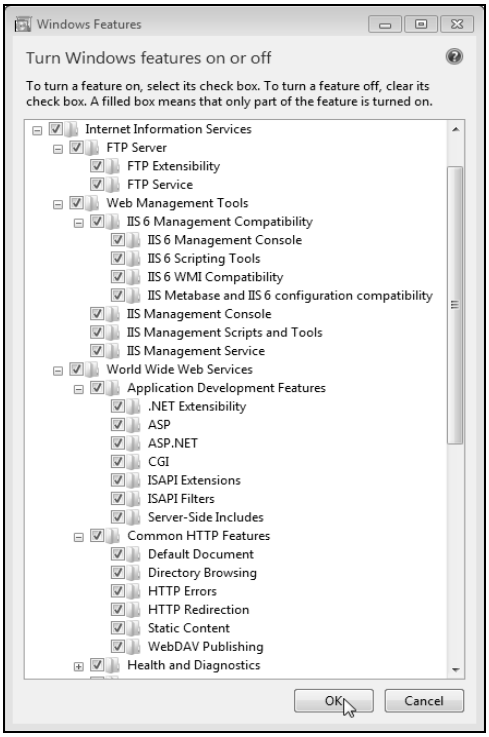

Figure D.2: Installing Features of IIS 7.5

A message box appears, stating that the selected features are configured, as shown in Figure D.3:

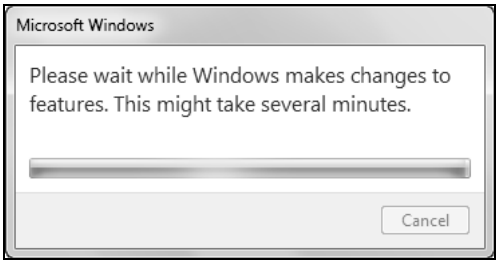

Figure D.3: Showing the Process of Configuring Features

After the configuration process is completed, you may be asked to restart your computer. In that case, restart your computer and check for the proper functioning of IIS 7.5.

Now that you have successfully installed and configured IIS 7.5 on your system, let's learn about the interface of IIS 7.5.

Describing the Interface of IIS 7.5

IIS 7.5 provides developers and administrators with a new configuration tool called IIS Manager. This is an Extensible Markup Language (XML)-based configuration tool, which makes it easy to implement the modules and features of IIS 7.5 in a Web site. IIS Manager brings the ASP.NET specific settings and IIS specific settings together where you can configure settings related to both, ASP.NET and IIS. You can open IIS Manager by either typing inetmgr in the Run dialog box or by selecting the Start→Control Panel→Administrative Tools→Internet Information Services (IIS) Manager option. Figure D.4 shows the Internet Information Services(IIS) Manager interface:

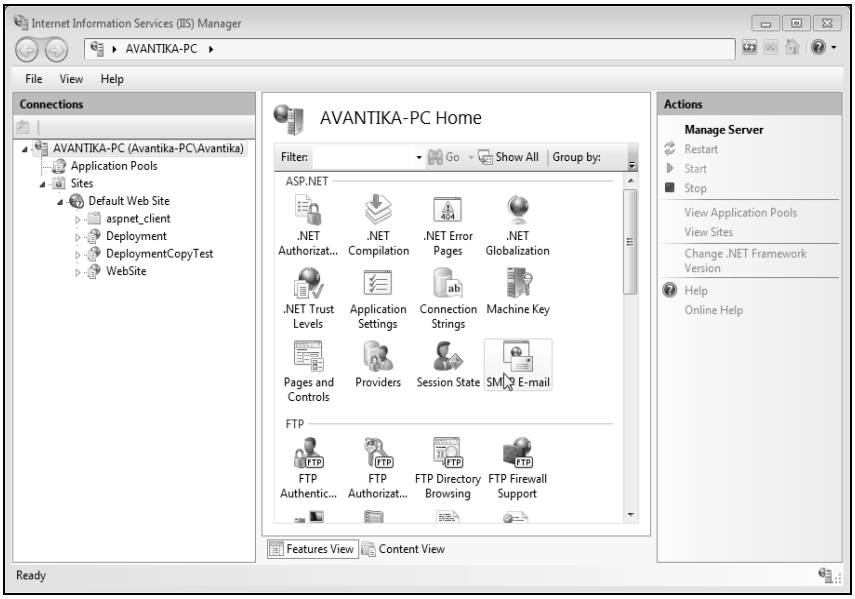

Figure D.4: Displaying the IIS Manager Interface

As you can see in Figure D.4, the IIS Manager interface is divided vertically into three parts. The Connections pane resides on the left, the Actions pane on the right, and the Home page in the middle. The Connections pane displays all the Web sites hosted on IIS. You can select your Web site from the Connections pane to configure it. When you select a Web site, the corresponding actions that you can perform are displayed in the Home page. You can select any feature and perform the related action from the Actions pane, such as disabling or enabling a

feature. The Actions pane is customized according to the selected feature. The various features related to the entire Web server or specific Web site are as follows:

- `.NET Compilation`—Configures the properties to compile managed code, including setting batch compilation to true or false, setting the Time-out limit, and the maximum file size.
- `.NET Globalization`—Configures globalization properties for managed code, including culture-specific properties and encoding settings.
- `.NET Profiles`—Configures the options to track user-selected preferences in ASP.NET applications. This feature is specific to a Web site and cannot be configured at the `machine.config` level.
- `.NET Roles`—Creates and configures the roles for .NET users and forms authentication.
- `.NET Trust Levels`—Configures the trust level policy files and the selected trust level for ASP.NET applications.
- `.NET Users`—Creates and manages users who belong to specific role and use forms authentication.
- `Application Settings`—Configures name and value pairs for managed code to use the managed code at runtime.
- `Connection Strings`—Configures the string that a Web site uses to connect to a data source.
- `Machine Key`—Configures hashing and encryption settings for managed application services.
- `Pages and Controls`—Configures the properties for Web pages and controls in ASP.NET applications.
- `Providers`—Configures providers for provider-based applications services.
- `Session State`—Configures session-state settings and cookie settings in forms authentication.
- `SMTP E-mail`—Configures the e-mail address and sending options to send e-mails from Web applications.
- `ASP`—Configures the properties for ASP.NET applications.
- `Authentication`—Configures the authentication settings for sites and applications.
- `Authorization Rules`—Configures the rules for authorizing users to access Web sites and Web applications.
- `CGI`—Configures the properties for CGI programs, which are specific functionalities intended to run on Web.
- `Compression`—Configures the settings to compress responses.
- `Default Document`—Specifies the default document to display when a client does not specify any specific page.
- `Directory Browsing`—Configures the information to be displayed in a directory listing.
- `Error Pages`—Configures the pages to be displayed when an error occurs.
- `Failed Request Tracing Rules`—Configures the path where the log failed requests that have been traced by the Web server are saved.
- `Handler Mappings`—Specifies the resources to handle a specific type of request.
- `HTTP Redirect`—Specifies the rules to redirect a request to another file or Web page.
- `HTTP Response Headers`—Specifies the headers that are added to the responses from the server.
- `FTP IPv4 Address and Domain Restrictions`—Restricts or grants access to Web content based on IPv4 addresses or domain names.
- `ISAPI and CGI Restrictions`—Restricts or enables specific ISAPI and CGI extensions on a server.
- `ISAPI Filters`—Specifies the ISAPI filters to modify IIS functionalities.
- `Logging`—Specifies how IIS will log a request on a Web server.
- `MIME Types`—Configures the extensions and associated content types that are served as static files to users.
- `Modules`—Configures Native and Managed modules used to process requests on a server.
- `Output Caching`—Specifies the rules for caching served content in the output cache.

- `SSL Settings`—Configures Secure Socket Layer (SSL) and client certificate settings to secure Web site content.
- `Server Certificates`—Requests for certificates and manages them for Web sites that use SSL.
- `Worker Processes`—Displays the information about worker processes and the requests running in these processes on IIS.
- `Feature Delegation`—Configures the default delegation state for the features at the lower levels in IIS Manager.
- `Shared Configuration`—Specifies whether to use the IIS configuration settings on the local computer or a remote location. You can also import your configuration setting from a specified location.

Now that you have learned about the architecture, interface, and various features of IIS 7.5, let's discuss how to deploy a Web site or Web application on IIS 7.5.

Deploying a Website on IIS 7.5

You must deploy a Web site or a Web application on IIS 7.5 to take advantage of the new functionalities introduced in IIS 7.5. Deploying a Web site implies using IIS 7.5 to host your Web site. You can deploy an existing Web site on IIS 7.5, irrespective of where it was created earlier. There are two ways of deploying a Web site on IIS 7.5.

Let's discuss these ways in detail now.

Method 1

This method is used to deploy a Web site by creating a virtual directory of the Web site folder on IIS 7.5 and mapping it with the physical path of Web site folder. Perform the following steps to deploy a Web site on IIS 7.5:

Create an ASP.NET Web site by selecting the File→New→Web Site option from the menu bar. The New Web Site dialog box appears (Figure D.5).

Select the ASP.NET Web Site option from the New Web Site dialog box (Figure D.5).

Select the HTTP option from the Web location drop-down list and enter an appropriate name for your Web site, and click the OK button as shown in Figure D.5:

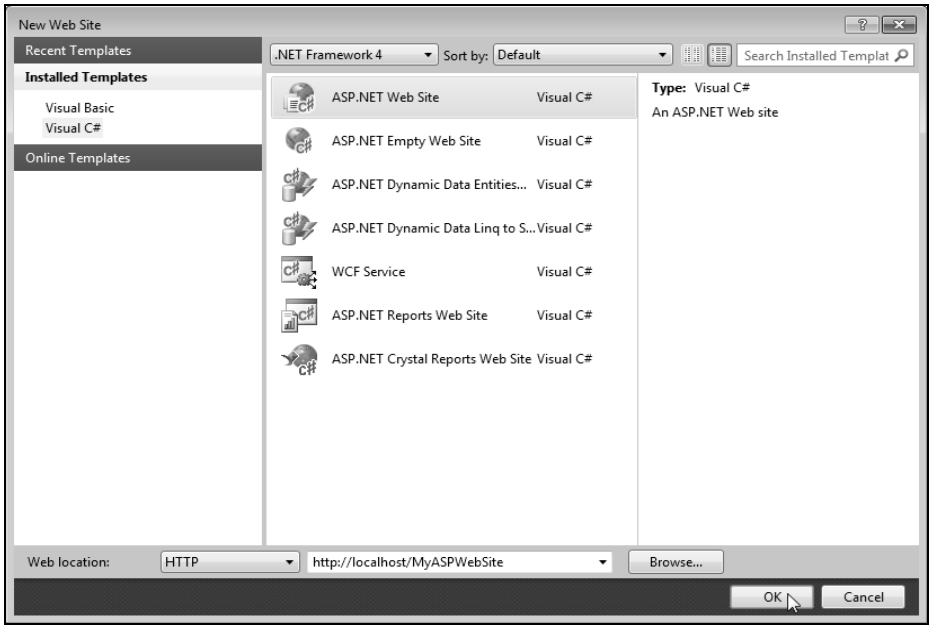

Figure D.5: Creating an ASP.NET Web Site

1. Open IIS Manager. Right-click the Default Web Site option from the Connections pane and select the Add Application option from the context menu, as shown in Figure D.6:

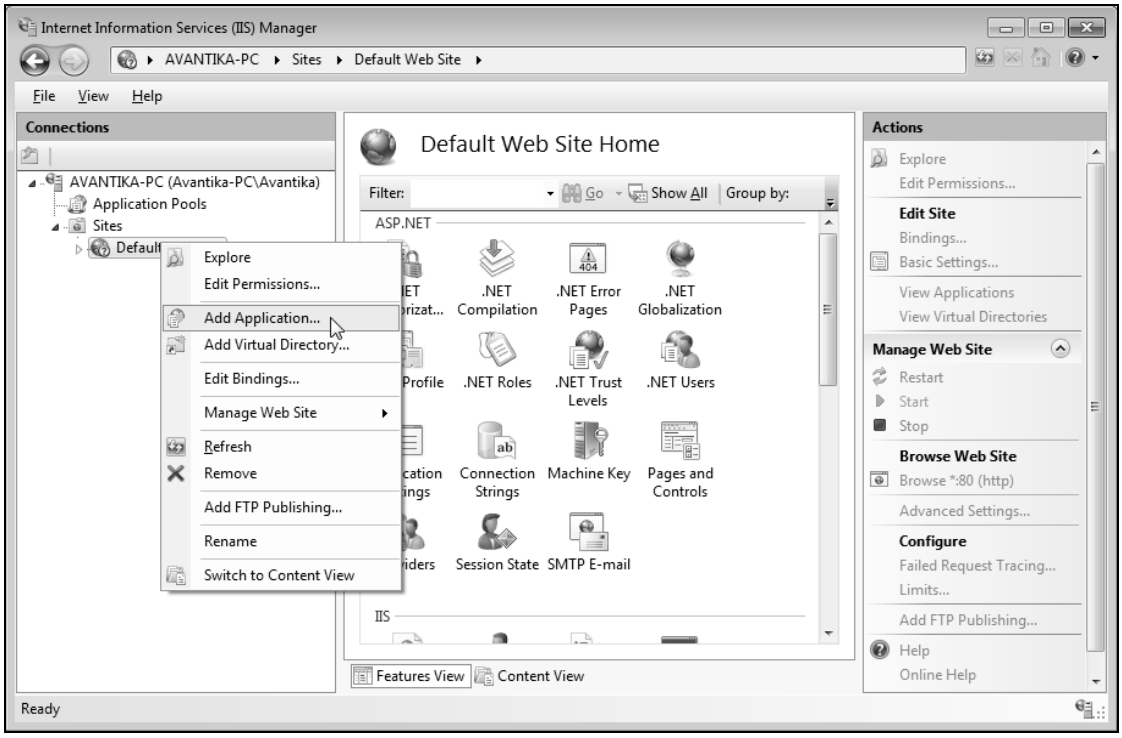

Figure D.6: Adding a Web Application

The Add Application dialog box appears, as shown in Figure D.7:

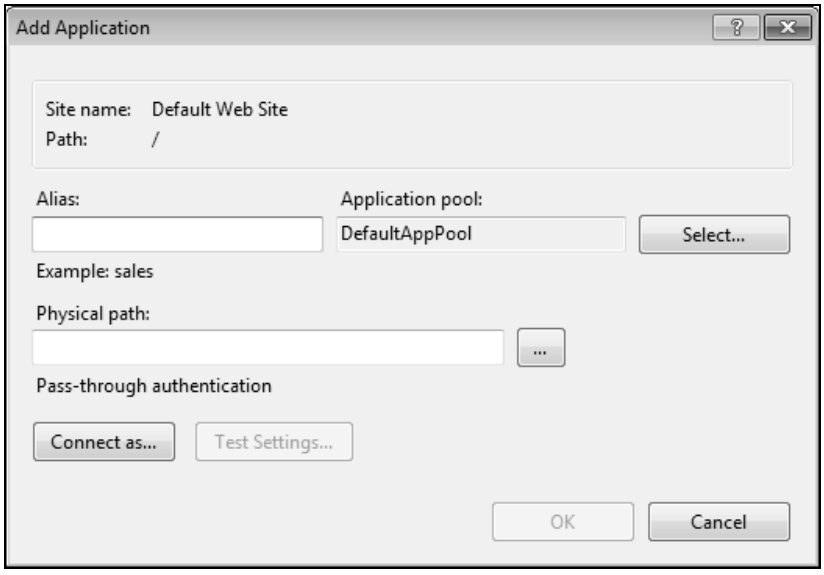

Figure D.7: Displaying the Add Application Dialog Box

2. Specify the alias name for the Web site in the Alias text box. In our case, we have specified `TestSite` (Figure D.8).
3. Click the Select button to open the Select Application Pool dialog box, as shown in Figure D.8:

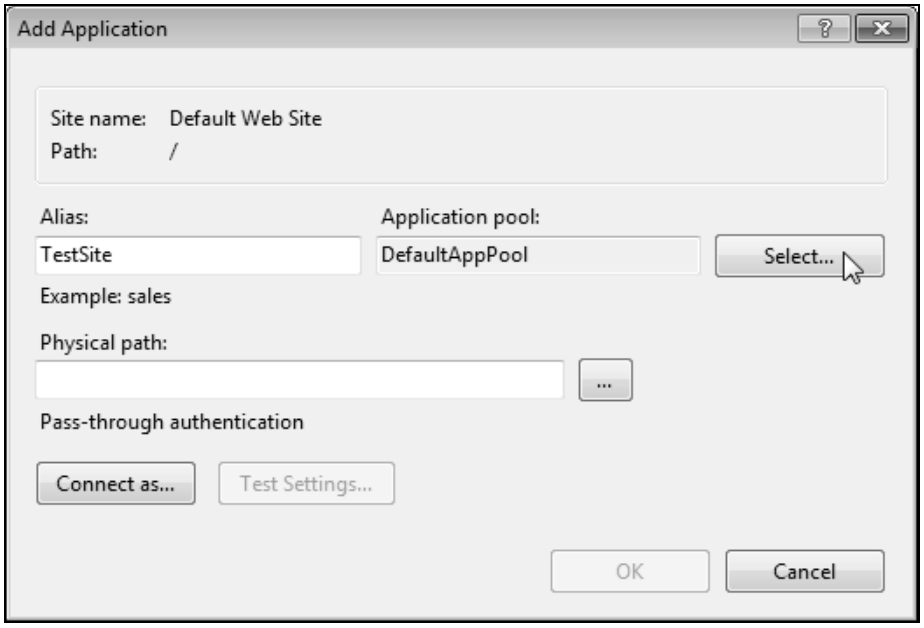

Figure D.8: Specifying an Alias Name for a Web Site

4. Select the ASP.NET v4.0 option from the Application pool drop-down list in the Select Application Pool dialog box and click the OK button, as shown in Figure D.9:

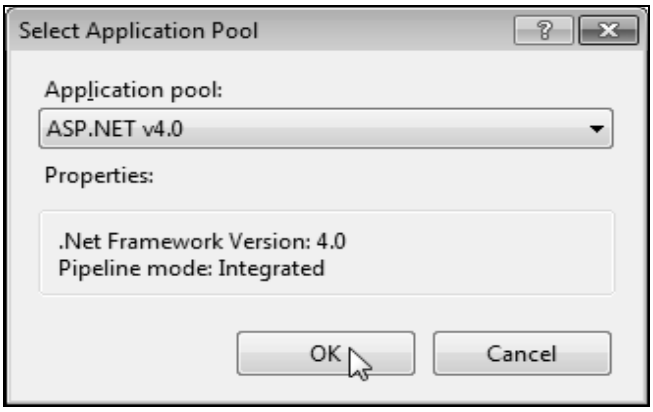

Figure D.9: Selecting an Application Pool

5. Click the ellipsis button beside the Physical Path text box in the Add Application dialog box to open the Browse For Folder dialog box (Figure D.10).
6. Select the path of the Web site that you want to deploy from the Browse For Folder dialog box. In our case, we have selected the Web site with the name MyASPWebSite (Figure D.10).
7. Click the OK button, as shown in Figure D.10:

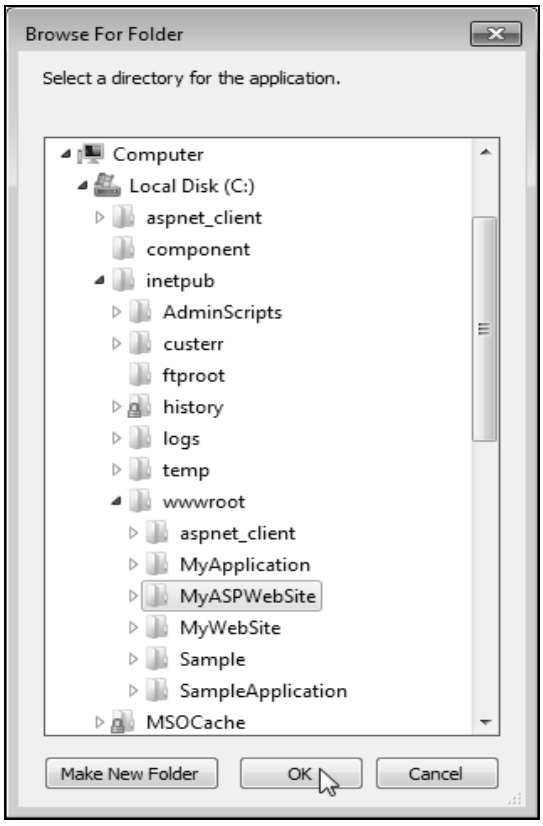

Figure D.10: Specifying Physical Path of a Web Site

This adds the Web site, MyASPWebSite, in the Connections pane with the specified Alias name, TestSite.

11. Click the Connect as button in the Add Application dialog box to open the Connect As dialog box.
12. Select the Specific user radio button in the Connect As dialog box and click the Set button to open the Set Credentials dialog box (Figure D.11).
13. Specify the user name and password in the User name, Password, and Confirm password text boxes in the Set Credentials dialog box and click the OK button, as shown in Figure D.11:

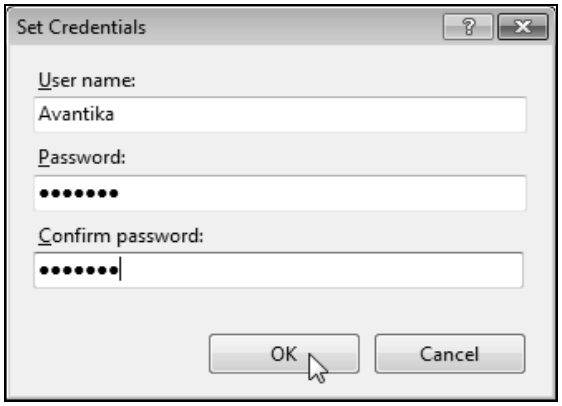

Figure D.11: Setting Credentials in the Set Credentials Dialog Box

The Connect As dialog box appears (Figure D.12).

14. Click the OK button in the Connect As dialog box, as shown in Figure D.12:

Figure D.12: Specifying a User

The Connect As dialog box closes and the Add Application dialog box reappears (Figure D.13).

15. Click the OK button in the Add Application dialog box, as shown in Figure D.13:

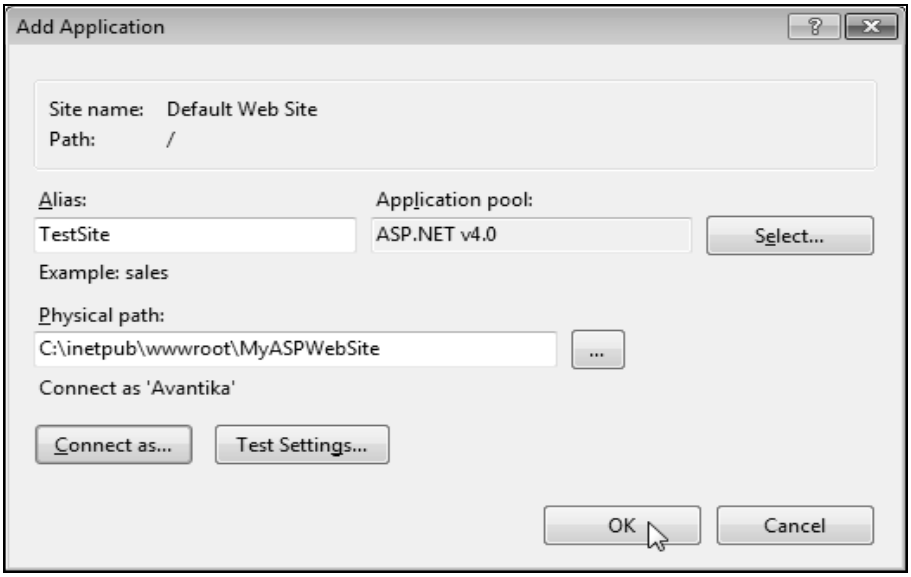

Figure D.13: Closing the Add Application Dialog Box

This adds the Web site, MyASPWebSite, on IIS 7.5. Now, you need to run it in on a Web browser.

16. Specify the path of your Web site in the Web browser as http://localhost/TestSite and press the Enter key. In this case, `TestSite` is the alias name for the Web site, MyASPWebSite. This name may be different for different Web applications. The default page of the Web site is displayed, as shown in Figure D.14:

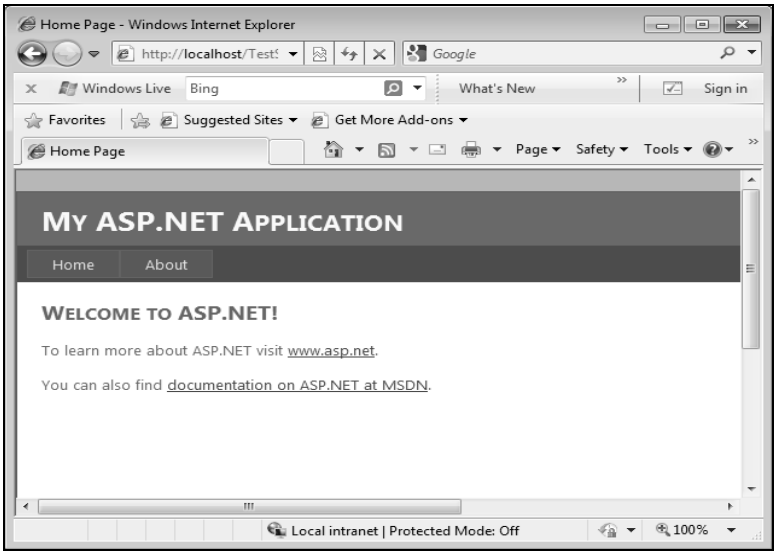

Figure D.14: Displaying a Web Site Being Deployed on IIS

NOTE

By default in ASP.NET 4.0, a Web site that you create has the Anonymous Authentication status as Enabled. In case, the Anonymous Authentication status for your Web site is disabled, then perform the following steps to enable it:

1. Select the Web site from the Connections pane and then double-click the Authentication option in the IIS group of the Home page, as shown in Figure D.15:

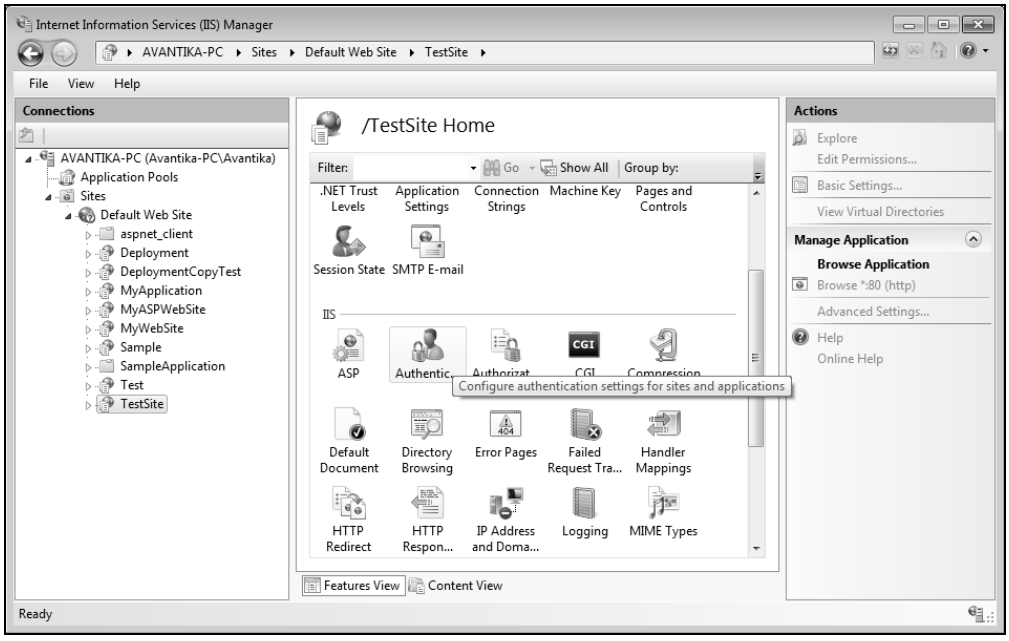

Figure D.15: Selecting the Authentication Template

2. Right-click the Anonymous Authentication option from the given authentication types and select the Enable option from the context menu, as shown in Figure D.16:

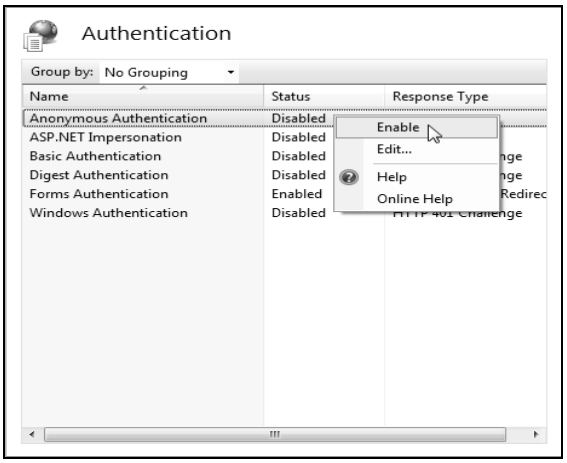

Figure D.16: Enabling Anonymous Authentication

NOTE

Now, you can deploy your Web site on a Web browser. Also note that after you have successfully deployed your Web site on IIS 7.5, you need to configure all the settings again.

With this, you have learned how to deploy an ASP.NET Web site on IIS 7.5. Note that while performing the steps to deploy a Web site on IIS 7.5, we selected the Application pool as ASP.NET v4.0. This is useful because each application pool gets its own worker process and the applications in a pool do not affect the applications in other pools. For instance, if an application causes a process to crash, the crash will affect that application's pool only. The Web server and other pools continue to run normally.

There are four built-in application pools in IIS 7.5. These application pools are DefaultAppPool, Classic .NET AppPool, ASP.NET v4.0, and ASP.NET v4.0 Classic. However, you can also create your own application pool.

Perform the following steps to create your own application pool:

3. Open the IIS Manager, right-click the Application Pools option in the Connections pane, and select the Add Application Pool option from the context menu, as shown in Figure D.17:

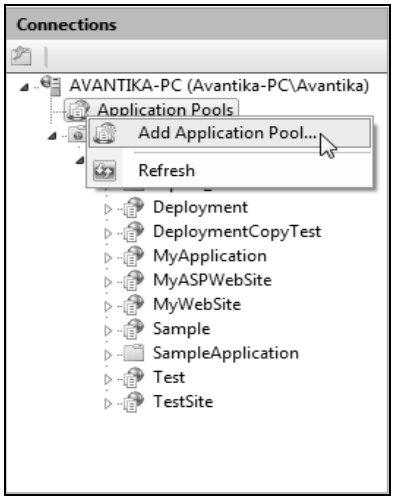

Figure D.17: Selecting the Add Application Pool Option

The Add Application Pool dialog box appears.

4. Specify the name of the application pool in the Name text box. In our case, we have specified MyApplicationPool (Figure D.18).
5. Select the .NET Framework v4.0.30319 option from the .NET Framework version drop-down list (Figure D.18).

Now, you have to select the managed pipeline mode. You can select either `Integrated` (to use IIS and ASP.NET request-processing pipeline) or `Classic` (to use Aspnet_isapi.dll for processing requests). In our case, we have selected the `Integrated` mode.

6. Select the `Integrated` managed pipeline mode and click the OK button, as shown in Figure D.18:

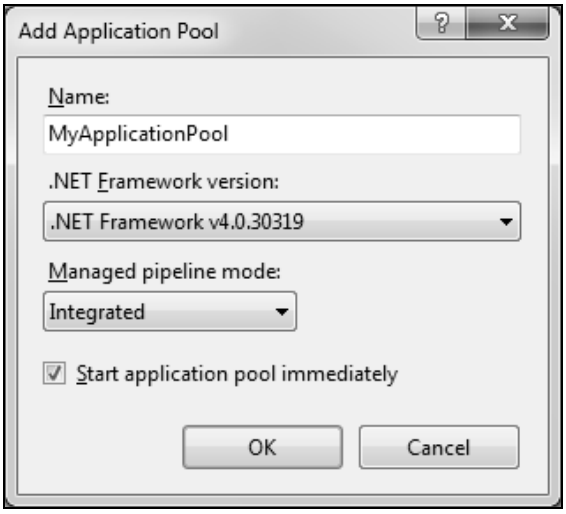

Figure D.18: Creating an Application Pool

You can see your newly created application pool, MyApplicationPool, along with the default ones in the Application Pools dialog box, as shown in Figure D.19:

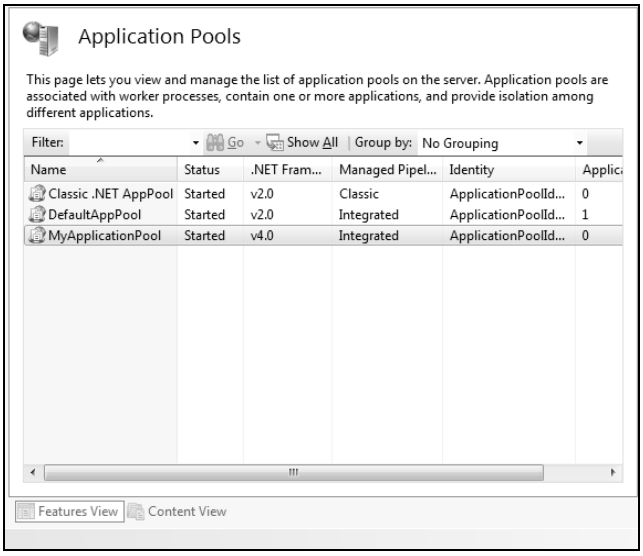

Figure D.19: Displaying an Application Pool Added to IIS Manager

After creating an application pool, you can add your Web site in this pool while deploying it on IIS 7.5. Moreover, you can also change the application pool of an already deployed Web site with this new application pool.

Perform the following steps to change already deployed Web application's pool:

7. Select a Web application or Web site that you have already deployed on IIS 7.5 from the Connections pane.
8. Click the Basic Settings link in the Actions pane, as shown in Figure D.20:

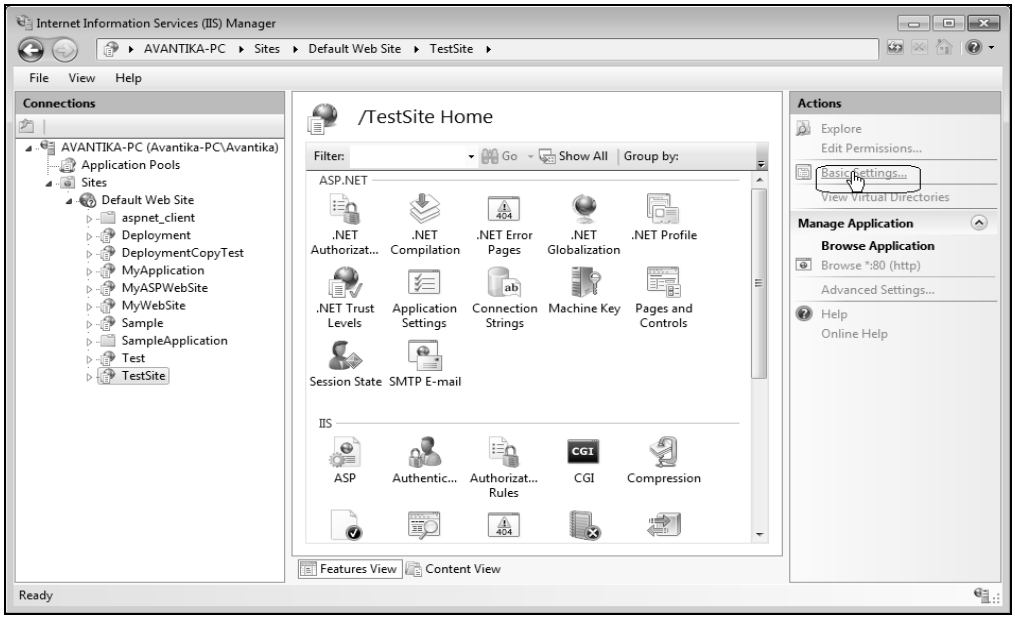

Figure D.20: Selecting the Basic Settings Link

The Edit Application dialog box appears, as shown in Figure D.21:

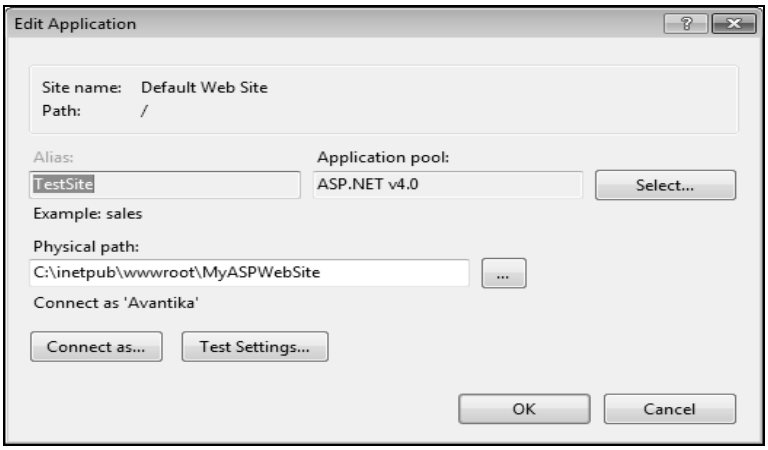

Figure D.21: Showing the Edit Application Dialog Box

9. Click the Select button beside the Application Pool text box in the Edit Application dialog box. The Select Application Pool dialog box appears (Figure D.22).
10. Select the desired application pool from the Select Application Pool dialog box and click the OK button, as shown in Figure D.22:

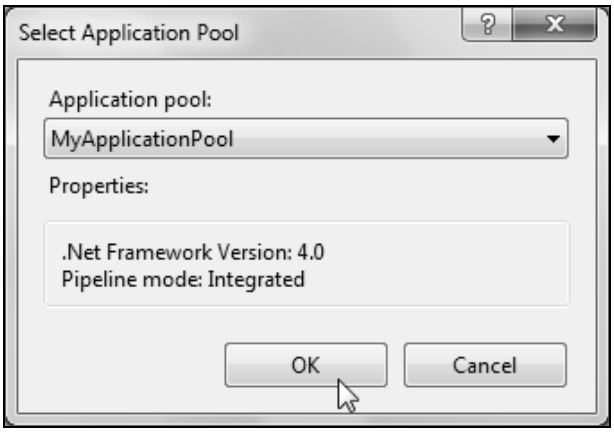

Figure D.22: Selecting Desired Application Pool

Now, let's discuss the second method, Method 2.

Method 2

This method is used to deploy a Web site by copying the actual Web site folder on IIS 7.5. Perform the following steps to deploy a Web site on IIS 7.5:

1. Copy a Web site folder in the C:/inetpub/wwwroot folder and open IIS Manager. In our case, we have added the SimpleWebSite folder. You can see the folder under the Default Web Site directory, as shown in Figure D.23:

Figure D.23: Displaying the Folder Added in the wwwroot Folder

Now, you need to convert the Web site, SimpleWebSite, into an application.

2. Right-click the Web site folder and select the Convert to Application option from the context menu, as shown in Figure D.24:

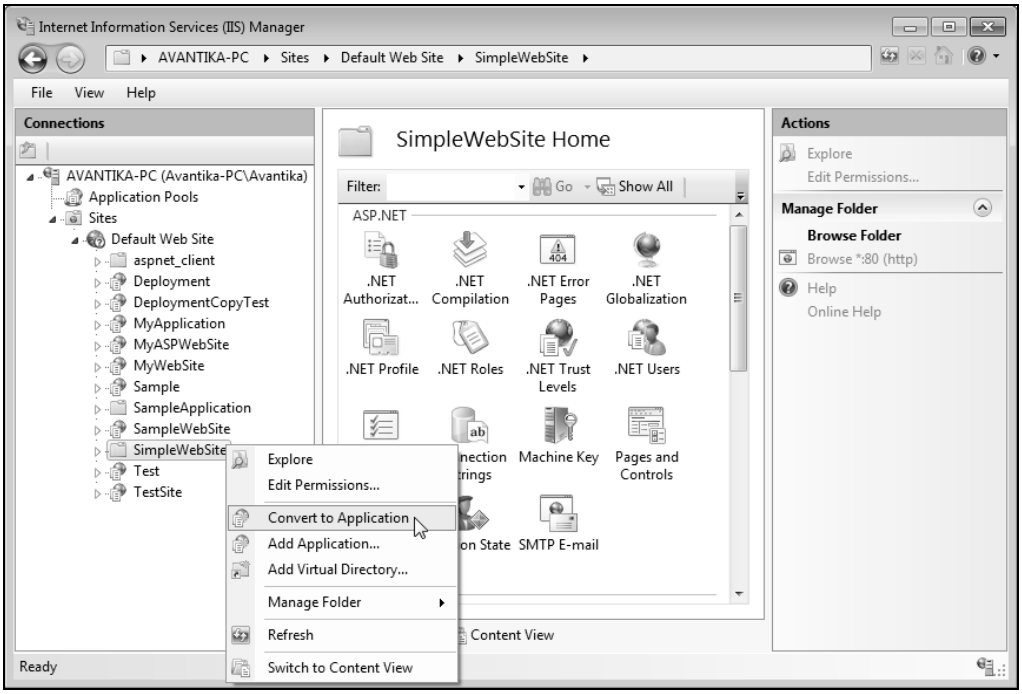

Figure D.24: Converting the Web Site to an Application

The Web site's icon has been changed, as shown in Figure D.25:

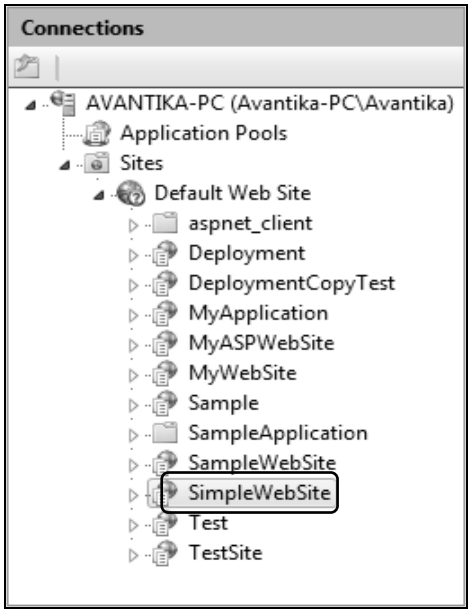

Figure D.25: Deploying a Converted Web Site on IIS

When you convert the Web site into an application, your Web site is automatically deployed on IIS. Now, you can change the configuration of the Web site from the Home page.

Let's now learn how to perform troubleshooting in IIS 7.5.

Troubleshooting in IIS 7.5

After deploying a Web site on IIS 7.5, you can run it on a Web browser. However, sometimes an error message, stating The page cannot be displayed, appears instead of the requested Web page. This may happen due to one or more reasons whose details are given in the error page. Sometimes, the user-friendly error messages are displayed; whereas, the actual errors are kept hidden. Therefore, to troubleshoot the problem, first you need to ensure that your browser displays the actual error message by disabling the user-friendly error message feature of your Web browser.

Perform the following steps to disable the user-friendly error message feature:

1. Open Internet Explorer and select Tools→Internet Options from the menu bar. The Internet Options dialog box appears (Figure D.26).
2. Select the Advanced tab from the Internet Options dialog box, as shown in Figure D.26:

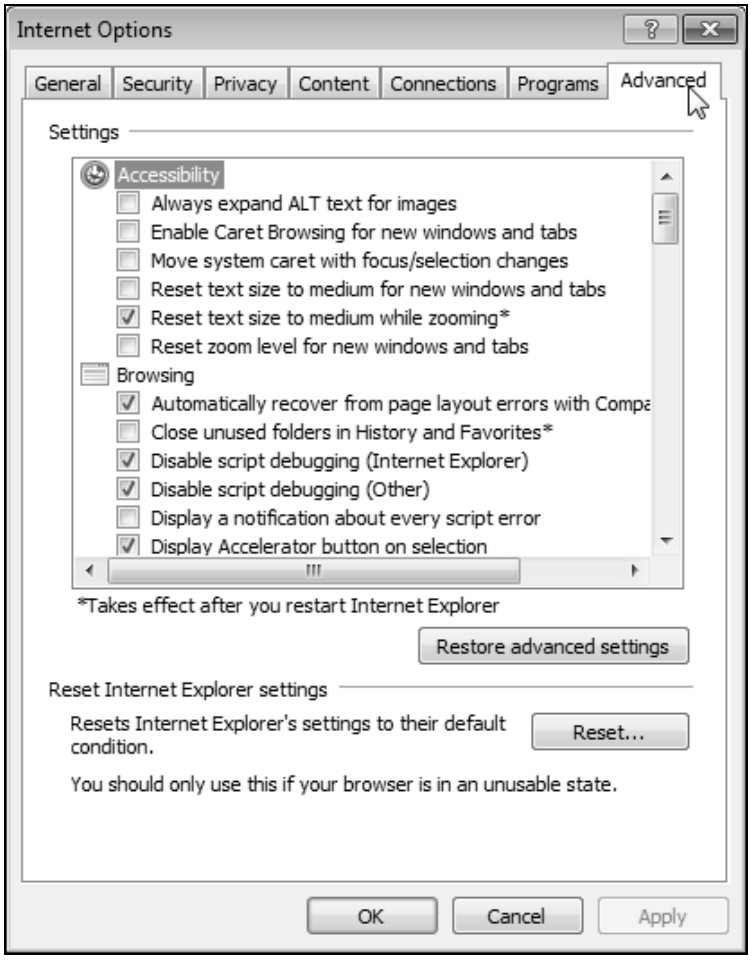

Figure D.26: Selecting the Advanced Tab

A list of check boxes about the advanced settings of Internet Explorer appears.

3. Clear the Show friendly HTTP error messages check box, as shown in Figure D.27:

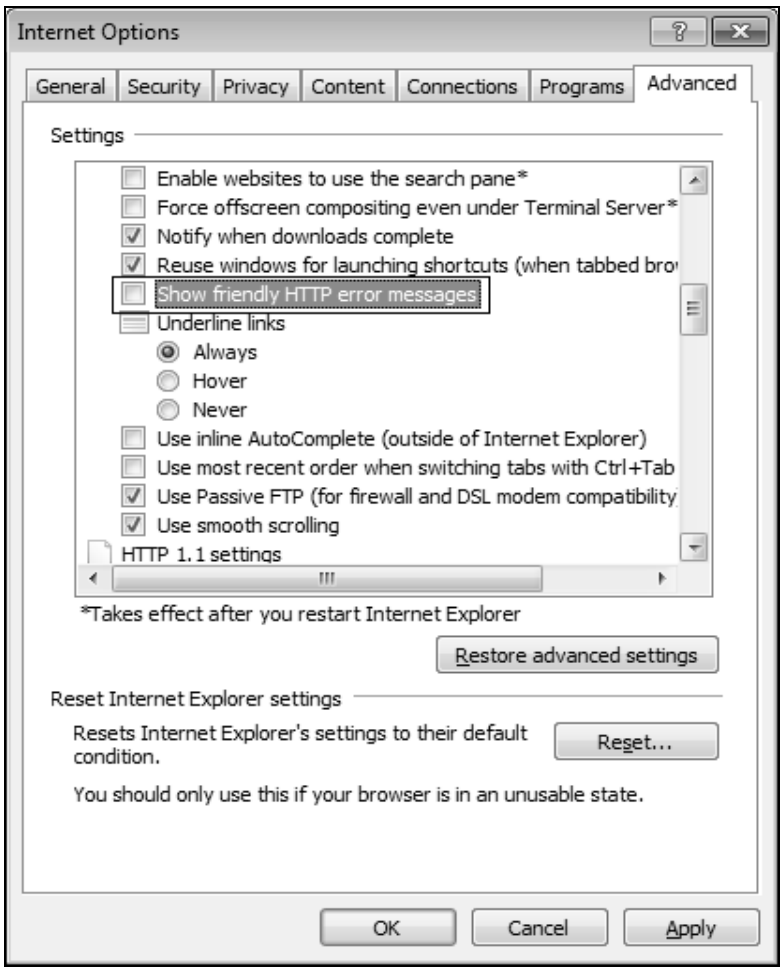

Figure D.27: Disabling the User-Friendly Error Message Check Box

Next, you need to enable the detailed error feature for the Web site. If you are accessing the Web site locally from the server, you get the detailed error message by default. However, if you are accessing the Web site remotely, then a customized error message is displayed that can make it difficult to get into the actual cause of the error. In such a scenario, you need to make changes in the `web.config` file to get the detailed error message by adding the <customErrors> tag within the <system.web> tag, as shown in the following code snippet:

```
<system.web>
        <compilation debug="true" targetFramework="4.0"/>
    <customErrors mode="Off"/>
  </system.web>
```

After making the changes, you can view the actual cause for the errors that prevented you from running the Web site on the browser. There will be an actual error code, which will help you to identify the problem and find a possible solution to resolve the error. Figure D.28 displays an example that shows an error, which is raised while running a Web site on the Web browser, and also a detailed description is given about the causes for the error and the possible actions that you can perform to resolve the error:

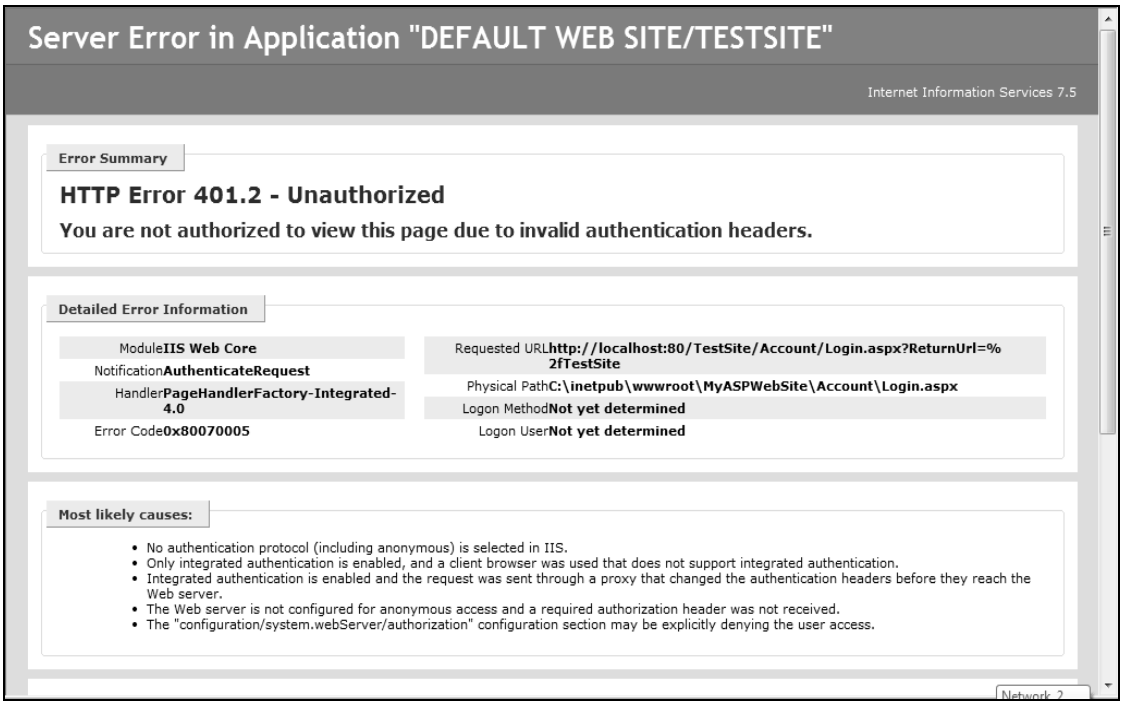

Figure D.28: Displaying a List of Most Likely Causes and the Corresponding Solutions

If you get an HTTP error as shown in Figure D.28, then open the Visual Studio Command Prompt (2010) window and type the following command:

cd C:\Windows\Microsoft.NET\Framework\v4.0.30319

After entering the preceding command, type the aspnet_regiis –iru command, as shown in Figure D.29:

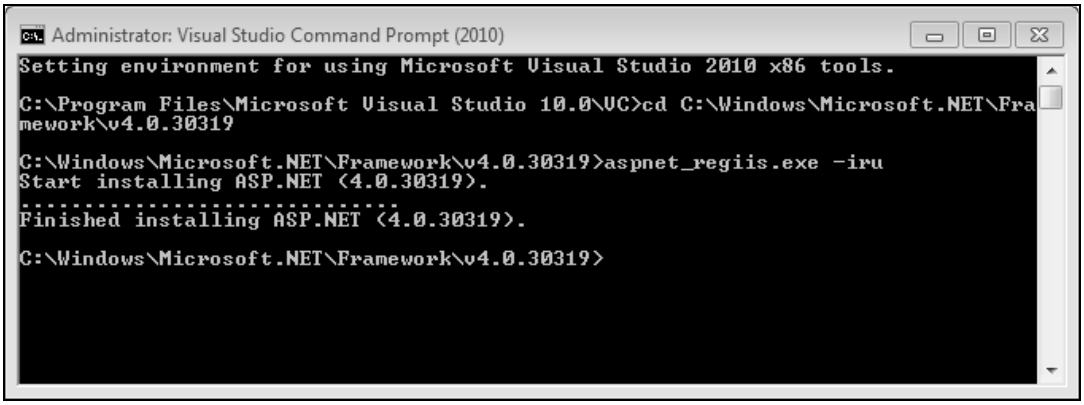

Figure D.29: Registering ASP.NET with IIS

aspet_regiis is an ASP.NET IIS registration tool that is used to register ASP.NET with IIS. Executing the aspnet_regiis.exe –iru command ensures that ASP.NET is correctly registered with IIS.

After executing the aspet_regiis.exe –iru command, deploy your Web application again on IIS.