MCTS Self-Paced Training Kit (Exam 70-643): Configuring Windows Server® 2008 Application Platform

J.C. Mackin and
Anil Desai

To learn more about this book, visit Microsoft Learning at
http://www.microsoft.com/MSPress/books/11756.aspx

978-0-7356-2511-2

Table of Contents

Chapter 2

Installing and Configuring Web Applications

Modern Web sites provide functionality that is on par with the experience found in many locally installed client applications. They provide access to databases in both public and intranet environments and enable users to customize their experience based on specific needs. These programs are known as Web applications or Web services, and they can rely upon a broad variety of different standards, protocols, and development technologies.

The Windows Server 2008 operating system includes *Internet Information Services (IIS)* 7.0, a complete Web services platform that is capable of supporting a broad variety of different types of Web content and applications. IIS 7.0 provides significant enhancements in manageability, scalability, and reliability. It also provides backward compatibility to support the millions of Web sites that are already hosted on previous versions of IIS.

In this chapter, you'll learn how to install and configure the *Web Server (IIS)* and Application Server roles in Windows Server 2008. There are numerous features and services that you can enable based on the needs of your environment. You'll also learn about how you can configure IIS to meet a wide variety of requirement types. This information will help you deploy and configure IIS and its related features in production environments.

Exam objectives in this chapter:

- Configuring a Web Services Infrastructure
  - Configuring backup.
  - Configuring Web applications.
  - Configuring Application Pools.
  - Configuring IIS components.
  - Publishing IIS Web sites.
  - Migrating sites and Web applications.

Lessons in this chapter:

Before You Begin

The steps in the practice exercises assume that you will be using either server1.contoso.com or server2.contoso.com to perform all the necessary steps and that the Application Server role has not yet been installed. You can perform most of the steps on another computer running Windows Server 2008, but you might need to make some adjustments to the steps. Also, although there are some licensing differences between the different editions of Windows Server 2008, the basic architecture of the Web Server (IIS) role is consistent among them.

Real World

Anil Desai

The success of a server installation is often based on how well its configuration matches the needs of users and developers. If some features are missing, applications will not run as expected. If too many features are enabled, there could be security, compatibility, or performance implications. The goal is to "get it right." This is one area in which communications are important.

In many IT departments, I've seen a significant disconnect between development teams (such as groups of Web developers) and the systems administrators who are responsible for deploying and supporting the applications that the developers create. Often, the responsibilities blur between these two parts of the organizations, and it can become difficult to figure out who is ultimately responsible for the final configuration.

Fortunately, these types of problems can be solved. On the systems administration side, IT staff should try to determine the specific business and technical needs of the Web applications that they support. Web developers can do their part by proactively communicating upcoming requirements and potential implications for the configuration of production servers. Documentation is helpful for thinking through and communicating the most important points. Finally, it's important not to forget about end users. Whether these are people who are part of your organization or the public at large, it's important to understand their specific reasons for visiting your Web sites. Marketing input can often help in this area.

Lesson 1: Installing the Web Server (IIS) Role

Although the basic steps and processes required for enabling IIS and its related components is usually a simple procedure; the primary challenge lies in understanding the architecture, components, and available features of the platform. In this lesson, you will learn about the modular architecture of IIS and how you can configure a computer running Windows Server 2008 as a Web server.

After this lesson, you will be able to:

- Describe the architecture of IIS 7.0, including new features.
- Define the purpose of the Application Server role.
- Describe the purpose of role services related to the Web Server (IIS) role.
- Install the Web Server (IIS) role and add and remove role services.
- Perform command-line installations and automated installations of the Web Server (IIS) role.

Estimated lesson time: 45 minutes

Understanding Web Server Security

IIS 7.0 represents the evolution of the Microsoft IIS platform over several years. It includes a broad array of features and options to support different types of Web services and applications. The process of installing IIS and its related features and options is simplified through using the Server Manager utility. As a systems administrator, you will be responsible for deploying IIS based on different needs and requirements. Therefore, it is important to understand the design of IIS before learning methods for installing the Web Server and Application Server roles. This section will provide details about deployment options for the IIS platform.

MORE INFO Other features of IIS

In addition to supporting Web applications, the IIS platform also provides server components for the File Transfer Protocol (FTP) and the Simple Mail Transfer Protocol (SMTP). This chapter focuses on Web-based applications. For more information about these other features, see Chapter 4, "Configuring FTP and SMTP Services."

Web Standards and Protocols

To understand the purpose and function of the IIS platform, you must first understand (or review) basic information about how Web services operate. *Hypertext Transfer Protocol (HTTP)* is the primary protocol used to communicate with Web services. HTTP is designed to provide a request–response model for communicating between different computers across a network. HTTP traffic is accessed by using Transmission Control Protocol/Internet

Protocol (TCP/IP)–based network connections. Due to the importance of Web-based traffic, most organizations allow their users to access the Internet by using TCP port 80, the default HTTP port. The HTTP protocol is stateless; that is, it provides no built-in mechanism to keep track of conversations between clients and servers. Each request must include details that identify the requester and any other data that might be required to complete a transaction.

By default, HTTP traffic is transmitted using a plaintext stream that can be decoded easily. Although this is acceptable when users are accessing public content, many Web sites and applications need to transmit information securely between clients and servers. The most common example is that of a payment-processing site that accepts credit card information over the Internet. The *HTTP Secure (HTTPS)* protocol is designed to provide support for encryption of HTTP-based traffic. By default, HTTPS connections use TCP port 443 for communications, although any other port can be used as well. The most commonly used encryption mechanisms are *Secure Sockets Layer (SSL)* and *Transport Layer Security (TLS)*. Other encryption mechanisms can also be used, especially in intranet environments.

Apart from using standards-based communications protocols, the popularity of Web-based content is based on a standard method of presenting information. The Hypertext Markup Language (HTML) is the primary specification for Web pages. The tag-based format of HTML pages enables developers to use a wide variety of technologies to create their own content in a way that is accessible to different Web browsers. The development tools can range from text editors such as Microsoft Windows Notepad to full-featured development environments such as the Microsoft Visual Studio platform.

The HTTP and HTML specifications were designed to provide basic communication and presentation services. Modern Web applications include features that enable complex application functionality to be presented using these standards. Web developers can use development platforms such as *ASP.NET* (a component of the Microsoft .NET Framework) to build active Web sites. These sites can keep track of user sessions and can provide access to databases and other information that is stored within the environment.

MORE INFO Further details about Internet standards

For more information about specific Internet and Web-based standards, see the World Wide Web Consortium (W3C) Web site at *http://www.w3.org* and the Internet Engineering Task Force (IETF) Web site at *http://www.ietf.org*. Both sites include the official specifications and descriptions for basic Internet protocols.

The IIS platform is designed to support standard Web services protocols and specifications as well as numerous additional features that can improve Web server performance and reliability. You'll learn more about these specific features later in this lesson.

Web Server Usage Scenarios

In modern IT environments, Web servers can be deployed to support a wide range of usage types. The primary advantage of using Web-based content and applications is accessibility from a broad range of client computers. Unlike standard applications, there is generally no need to install or configure any software on users' computers. Because modern operating systems include or support standards-based Web browsers such as Microsoft Internet Explorer, most users already have the basic client tools they need to access content. IT staff and software developers can use a variety of different technologies to present content and deploy applications to both internal and external users.

The IIS platform has been designed to support a wide variety of usage scenarios. Some examples include:

- **Public Web sites** Many businesses have relatively simple needs for communicating information on the Internet. For example, a small business might want to provide contact information and details about its services on a simple Web site.
- **Online shopping** The Internet has become a commercial marketplace that enables vendors to display and sell a wide variety of products. Online sites include shopping-cart functionality, order processing, and customer support features.
- **Intranet scenarios** The Web provides a simple method for all users within an organization to access and present content. Company tasks such as creating expense reports or verifying benefits can often be performed online without the need to contact internal staff.
- **Enterprise applications** A common challenge with enterprise line-of-business applications is the need to deploy and manage client-side installations. To alleviate some of these problems, many organizations have created internal applications that are designed to be accessed through a Web browser. The applications can range from basic single-function sites to distributed enterprise-wide systems.
- **Internet applications** Web-based solutions are available for performing a wide array of computing tasks by using standards such as HTML. Users can access their e-mail and create documents, for example, without installing applications on their computers. Distributed organizations and teams can also take advantage of secure access to corporate applications by using the Internet while traveling or working from remote locations.
- **Extranet scenarios** Businesses commonly partner with other organizations to obtain services. An extranet scenario is one in which users from outside the organization are able to access information. Security is an important concern, but Web-based applications are a good choice because they provide a standard method by which a broad range of users can access the information they need.
- **Web hosting** Many companies have focused on offering the service of hosting Web sites for their customers. These hosting companies tend to run large numbers of Web sites on a single physical server. Ensuring security, performance, and reliability are key concerns.

Most organizations will deploy IIS in several different roles within the organization. It is important to note that requirements related to features and options will vary based on the specific needs of each deployment.

Exam Tip When learning about the many different features and options of the IIS platform, it often helps to think of scenarios in which those features can be helpful to meet technical or business requirements. When taking Exam 70-643, expect to see questions that require you to understand specific requirements and find the most appropriate option or feature to meet those requirements.

You'll learn more about the specific features and services that the IIS platform supports later in this lesson.

New Features in IIS

The IIS platform is one of the most popular Web servers in use for both public and private Web sites. IIS 7.0 in Windows Server 2008 includes numerous new features that provide increased performance and functionality in a broad range of areas. The major areas of improvement include:

- **Administration** One of the primary challenges of working with previous versions of IIS was dealing with a large number of property pages and dialog boxes. IIS 7.0 includes new administration tools that are designed to manage the many available options and settings more effectively. The user interface has been designed to be both powerful and accessible for both Web developers and systems administrators.
- **Security** By default, the Web Server (IIS) server role is enabled with only a basic set of functionality. Even the binary files for unused features are not available for access in the standard operating system locations. Systems administrators must enable additional services and features explicitly. This helps reduce the attack surface of IIS while also simplifying manageability. In addition, functionality for automatically detecting common hacking attempts is included with the product itself. (This feature was commonly enabled in the past by installing the URLScan utility.)
- **Diagnostics and troubleshooting** Because organizations depend on Web services as a mission-critical component of their infrastructure, it's important to detect and resolve any Web-based errors quickly. IIS 7.0 includes new features that make it easier to pinpoint problems and obtain the details necessary to address them.
- **Centralized configuration management** Many organizations support dozens or even hundreds of IIS installations. To meet scalability and performance requirements, it is often necessary to deploy numerous Web servers that essentially have the same configuration settings. In previous versions of IIS, it was difficult to manage these configurations without connecting to each of the servers. IIS 7.0 provides a simplified method by which administrators can share configuration information across server farms. Further,

a consistent set of user accounts (including globally unique identifiers [GUIDs] and permissions) are used for IIS security accounts. This means administrators can depend on specific account names and settings when scripting and automating common processes. IIS 7.0 also includes greatly improved command-line support.

- **Support for delegation** It is often necessary to divide Web server administration tasks for security or management reasons. IIS 7.0 provides the ability to implement granular security configuration permissions to support Web-hosting environments and enterprise-level configurations.
- **Backward compatibility** The vast majority of Web sites and applications that were created for previous versions of IIS will remain compatible with IIS 7.0. In addition, IIS 6.0 management tools are provided for those applications that depend on them.

Overall, IIS 7.0 has been designed to address the most common issues encountered with previous versions of IIS. There are also numerous additional improvements in IIS that you'll learn about as this chapter discusses the various features in depth.

MORE INFO IIS in Windows Vista

Microsoft first made the IIS 7.0 platform available in the Windows Vista operating system. Because the core architecture of IIS in Windows Vista is similar to that in Windows Server 2008, Web developers can use similar environments on both their development workstations and their production servers. It is important to note that there are some feature and licensing differences between the two platforms. For more information, see the Microsoft Internet Information Services Web site at *http://www.microsoft.com/iis/*.

MORE INFO Information from the IIS team

The IIS team at Microsoft has created a Web site that includes tutorials, technical articles, and other details about working with the IIS platform. This is a great resource for in-depth information on the many different features and components that are available. The site includes links to downloads and information about products that work with (or on) the IIS platform. Team members have their own blogs, too, which focus on their specific areas of expertise. The main page is located at *http://www.iis.net*.

Understanding IIS Components and Options

The IIS platform has been designed with a modular, component-based architecture. In its simplest configuration, the Web server component provides basic HTTP functionality. IIS includes many components and features that can be used to support different types of content and applications. Most deployments will need only a subset of these features. Therefore, administrators can choose to enable only those components that their Web applications require.

Although the modular approach requires systems administrators to enable explicitly the features that they require, this architecture provides numerous advantages:

- **Enhanced security** Each enabled service or feature potentially can increase the security attack surface on an IIS server. This is a significant concern for publicly accessible servers that might be the targets of malicious attacks of unauthorized access attempts. For example, a defect or vulnerability in a specific type of IIS extension might be used to perform unauthorized actions on the server. Administrators can reduce these risks greatly by enabling only those features and services that are required by their content and applications.
- **Improving performance** Installing and enabling unnecessary components can use up system resources on the server that is running IIS. By enabling only those features that are required specifically, server resources can be retained for use by other applications. The end result is better performance and scalability.
- **Customizing server configurations** As mentioned earlier in this lesson, organizations tend to use IIS in a wide variety of deployment scenarios. The security and functionality requirements can vary significantly, and a modular architecture allows systems administrators to customize each deployment based on its specific needs. For example, the authentication and security requirements of internal Web servers and Internet accessible servers often differ. Administrators can enable the required features for each type of server independently.

In this section, you'll learn about components and options that are related to the IIS platform.

Understanding the Application Server Role

One of the primary strengths of the Windows platform is its ability to support a wide range of different application development technologies. Modern applications often rely on extensive communications features. For example, a distributed application might need to create and manage transactions across several different sites and services using a distributed network. Building this type of functionality can be difficult and complicated. Applications developers can save significant time and effort by taking advantage of the features that are already available on their operating system platform.

Windows Server 2008 includes the Application Server role to provide support for a variety of different application development technologies. The Application Server role is based on .NET Framework 3.0 technology and includes support for other communications and presentation features. Although the Application Server role is not specifically dependent on the Web Server (IIS) role, distributed applications that are built using ASP.NET or Windows Communication Foundation (WCF) will require both roles.

Exam Tip The Application Server role provides additional functionality on top of ASP.NET support and other services that are available for the Web Server (IIS) role. In general, you should not need to install the Application Server role unless a specific Web application or Web service requires it. Basic ASP.NET applications, for example, will run without the Application Server role enabled on the server.

You can install the Application Server role by using the Add Roles Wizard in Server Manager. When you add the role, you will be given the option of determining which additional role services you plan to enable. (See Figure 2-1.) The specific features include:

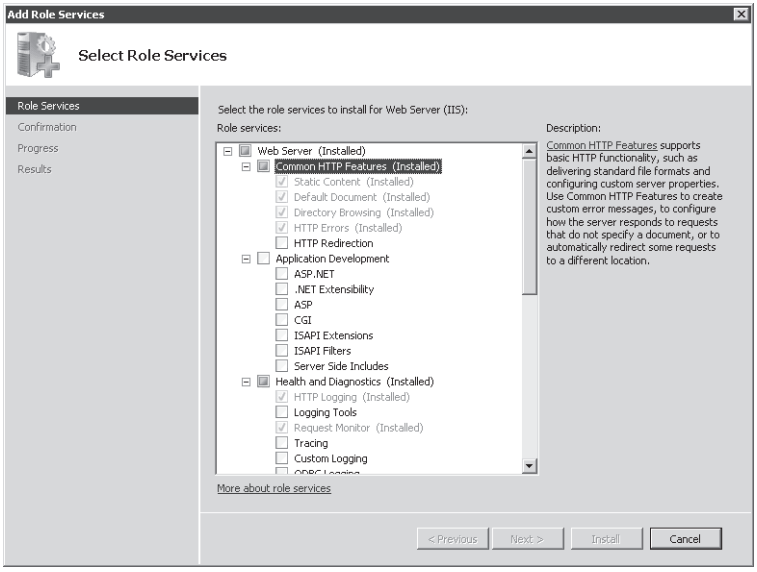

Figure 2-1 Viewing a list of available role services for the Application Server role

- **Application Server Foundation** This is a required feature of the Application Server role. It includes support for technology in the .NET Framework 3.0 platform. The primary technology components are the Windows Communication Foundation (WCF), Windows Presentation Foundation (WPF), and Windows Workflow Foundation (WF).
- **Web Server (IIS) Support** The Application Server role can be integrated with the Web Server (IIS) role to enable Web applications to access advanced features. When you select this option, the Add Roles Wizard will prompt you to install IIS automatically if it is not already installed.
- **COM+ Network Access** The Component Object Model (COM) standard provides applications developers with a method for accessing different pieces of application code. COM+ provides the ability to invoke (or access) application code remotely

across a network. Distributed applications, such as those that require multiple tiers of functionality, might require this feature.

- **TCP Port Sharing** A potential management challenge of working in distributed environments is that of supporting many different server applications on a single computer. Generally, each application requires its own TCP port for responding to inbound requests. The TCP Port Sharing feature enables multiple applications to share the same port to simplify server and firewall configuration.
- **Windows Process Activation Service Support** The Windows Process Activation Service (WAS) provides the ability to access application services over the network by using different types of protocols and services. This feature can be used by IIS itself to support additional protocols and communications methods.
- **Distributed Transactions** Applications that involve distributed transactions require multiple servers and applications to coordinate their activities before changes are made permanent. By using this section, you enable incoming and outgoing remote transactions and support the WS-Atomic Transactions standard for Web Services.

Generally, you should verify requirements with Web application developers to determine which Application Server components (if any) are required.

When done correctly, collecting and communicating Web server requirements can help ensure that systems administrators are aligned with the developers and users that they support. From an IT standpoint, IIS is one of those technology areas that can benefit from input and expertise from all areas of your organization. Be sure to do your homework before diving into the configuration process and you're much more likely to end up with the right IIS configuration.

Using Windows System Resource Manager

An important consideration for any server is to ensure that critical services are not interrupted when the system is under load. By default, most services in Windows Server 2008 run at an equal priority level. *Windows System Resource Manager (WSRM)* helps administrators assign priorities to various system processes such as IIS. Although WSRM is not a requirement for running IIS, on busy Web servers or servers that are providing many important services, enabling this feature can be helpful. For example, administrators can create Resource Allocation policies to define CPU and memory limitations to ensure that the system continues to respond well even when under heavy load. (See Figure 2-2.)

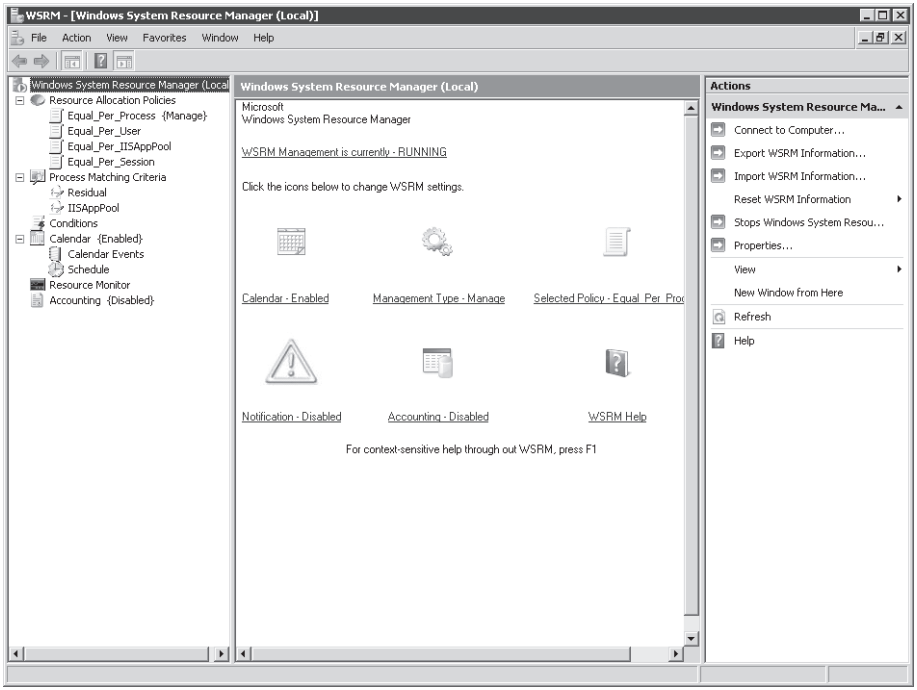

Figure 2-2 The Windows System Resource Manager console

You can add WSRM to a computer running Windows Server 2008 by using Server Manager. Right-click the Server Manager item and select Add Features to start the process. The Add Features Wizard includes an option to add WSRM. For more information about WSRM, In the Start menu Start Search box type "system resource" and then press Enter. The help file includes details on creating and managing resource settings.

Understanding IIS 7.0 Role Services

Role services define which specific features and options of the IIS platform are available for use on the local Web server. Once you have installed IIS 7.0 on a computer running Windows Server 2008, you can add additional components by using Server Manager. To do this, expand Roles, right-click Web Server (IIS), and select Add Role Services. You will see a dialog box like the one shown in Figure 2-3.

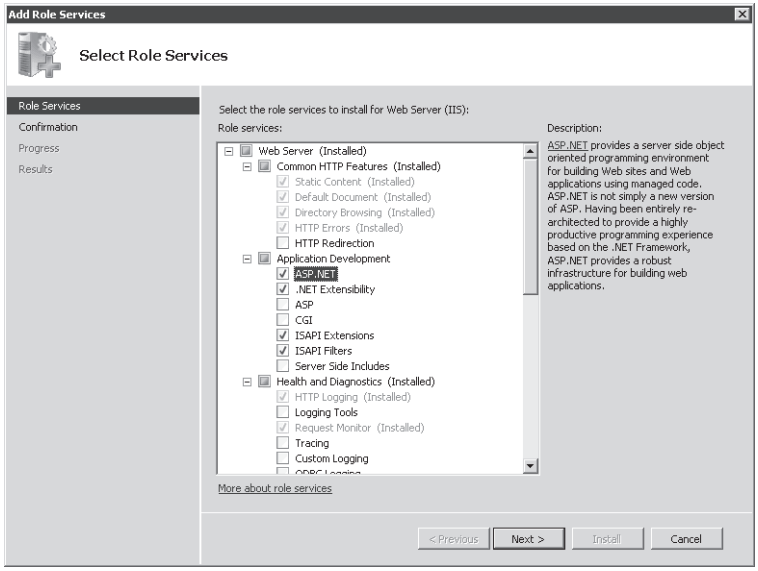

Figure 2-3 Managing Web Server role services in Server Manager

IIS role services are organized into several major areas:

- Common HTTP Features
- Application Development
- Health and Diagnostics
- Security
- Performance
- Management Tools
- FTP Publishing Service

The top level of the hierarchy is the Web Server itself. This item represents the core IIS services that are required by the below optional components that are listed below. Two other items, Management Tools and the FTP Publishing Service, can be installed independently of the Web Server. Each area contains features and options that are related. Several of the items depend on other role services. If you select an item without first selecting its dependencies, you will be given the option to add the required role services automatically. (See Figure 2-4.)

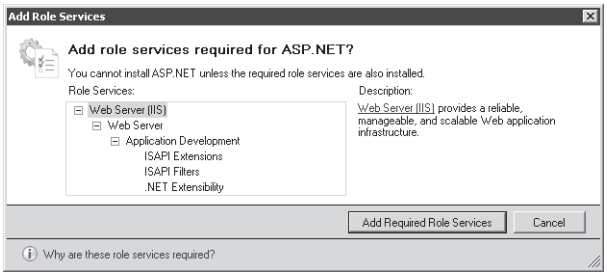

Figure 2-4 Including role dependencies when adding a role service

Exam Tip Note that adding a role service makes it available for use by your Web sites and applications. Additional configuration is sometimes required to take advantage of the service. For example, enabling certain authentication options will not make them automatically apply to all your Web sites. When taking Exam 70-643, keep in mind that adding a Web Server role service might be only one step in meeting the complete solution requirements.

Default IIS Role Services

As mentioned earlier, the default configuration includes a limited set of functionality. It is appropriate for installations that want to serve only limited static content and do not need advanced security or development features. In many cases, you will want to enable additional options.

Table 2-1 lists the role services that are included when you add the Web Server (IIS) server role to the computer.

Table 2-1 Default Role Services Included in the Web Server (IIS) Server Role

Group/Category	Feature(s)
Common HTTP Features	■ Static Content ■ Default Document ■ Directory Browsing ■ HTTP Errors ■ HTTP Redirection
Application Development Features	■ .NET Extensibility

Table 2-1 Default Role Services Included in the Web Server (IIS) Server Role

Group/Category	Feature(s)
Health and Diagnostics Features	■ HTTP Logging ■ Logging Tools ■ Request Monitor ■ Tracing
Security	■ Request Filtering
Performance Features	■ Static Content Compression
Management Tools	■ IIS Management Console
Windows Process Activation Service Features	■ Process Model

In the following sections, you'll learn more about the purpose of these and the many optional role services.

Common HTTP Features

The most important capability of the Web Server (IIS) role is to provide support for serving HTML Web pages using the HTTP protocol. All the components of the Common HTTP Features group are included in the default installation. They are:

- **Static Content** This functionality allows for serving static Web pages to clients, using HTTP. The most common content types are static HTML pages and images. Static content files are usually sent directly to users without any server-side processing.
- **Default Document** This feature allows IIS to return a specific file automatically for a Web site when one is not explicitly requested in the URL. For example, if a user attempts to connect to *http://www.contoso.com*, the Web server can be configured to return the default.htm file as a response.
- **Directory Browsing** IIS includes built-in functionality for providing basic directory listings to users. When enabled, directory browsing sends information about the files and folders on a Web site to the client's Web browser. (See Figure 2-5.). Because users will have the ability to access and download any files to which they have the appropriate permissions, this feature is usually disabled for public Web sites. If the default document feature is enabled and a default document is found, users will not see the directory browsing screen.

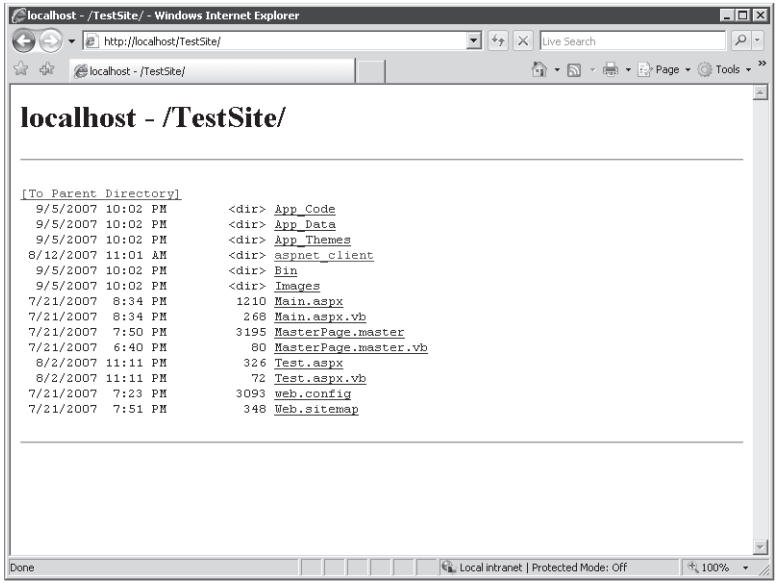

Figure 2-5 Using directory browsing to view the contents of a Web site

- **HTTP Errors** By default, most Web browsers are designed to present an error message automatically to users whenever a problem occurs. For example, if a page cannot be found or if the server is too busy, the Web browser will display this information to the user. To enhance the user experience, IIS can be configured to return custom error pages automatically when these problems occur. The content of the error pages can include contact information for the Web site's administrator or other details about resolving the problem.
- **HTTP Redirection** The HTTP protocol supports a method of redirecting a request from one site to another. The Web server can be configured to send an HTTP redirect request automatically to a Web user when a specific site is accessed. Site redirection is useful for situations in which a Web site has been relocated to a different URL or when multiple URLs are designed to access the same content.

Although all the Common HTTP Features are enabled by default, the specific behavior of each IIS Web site will be based on its content and configuration settings.

Application Development Features

Although some basic Web sites can meet their requirements by using only static content, it's far more common for production sites to require dynamic Web services and Web application support. IIS has been designed to support a broad array of different features and technologies to support these requirements. The list of Application Development role services includes:

- **ASP.NET** ASP.NET is the primary Microsoft Web server development platform. It is based on the .NET Framework and provides a powerful and flexible development framework for handling common Web site design tasks. Features include built-in support for managing access to databases, security and authorization methods, and reliability and scalability features.
- **.NET Extensibility** The Microsoft .NET Framework programming platform can be used to make modifications to IIS Web server functionality. This role service enables developers to access the IIS management namespaces and objects for building logic that interacts with Web server requests.
- **ASP** Active Server Pages (ASP) technology is the predecessor to the ASP.NET platform. ASP provided a simplified, script-based method of developing Web-based applications. ASP scripts run on the Web server and generate HTML content that is passed back to the user through IIS. Support for ASP is provided primarily for backward compatibility with applications that have not yet been moved to the ASP.NET platform.
- **CGI** The Common Gateway Interface (CGI) is a standard that defines how Web servers can pass information to programmatic scripts. It is required by some server-side components, especially those that have been written to run on multiple Web server platforms. Web development languages such as PHP: Hypertext Preprocessor (PHP) rely on CGI support within the Web server. IIS 7.0 includes features that can improve the performance of CGI processing significantly.
- **ISAPI extensions** IIS supports an extensibility standard known as the Internet Server Application Programming Interface (ISAPI). By building ISAPI extensions, Web developers can create their own content handlers that can interact with every aspect of the Web request pipeline. The ISAPI standard is designed to provide scalability for supporting many simultaneous requests.
- **ISAPI filters** ISAPI filters are custom code that developers can create to process specific Web server requests. The logic can receive Web request details and return the appropriate content based on server-side logic. IIS attempts to match Web requests with the most appropriate ISAPI filter for handling that type of content. Enabling this role service allows developers to add custom ISAPI filters to IIS.

- **Server-Side Includes** Web designers can often benefit from having the ability to embed certain common content on all their Web pages. Examples include a site header, navigation elements, and site footers. The Server Side Includes role service enables the Web server to include other pieces of content when generating a Web server request. For security reasons, this feature is disabled by default. However, sites that do not rely on other Web development technologies (such as ASP.NET) might require this capability.

Of these role services, only the .NET Extensibility item is installed by default. When planning to deploy production Web sites, determine which additional features should be enabled. This information is usually available from the Web application development team or organization.

Health and Diagnostics Features

Although basic Web server functionality can appear simple, there are numerous steps that must be performed during the processing of a typical Web request. Organizations that depend on their Web servers for access to critical information and systems need a method of isolating and troubleshooting any problems that might occur. Role services that are included in the Health and Diagnostics features section are designed to help administrators and developers collect and analyze information about Web requests.

A common challenge with monitoring Web sites is managing the volume of information that is generated. The process of recording in-depth details about all requests can add a significant level of performance overhead to production systems. To help address this issue, IIS 7.0 includes enhanced features for collecting details on specific requests and for configuring which information should be collected. The specific role services are:

- **HTTP logging** The most basic form of logging in IIS is to store HTTP request information within text files on the server's file system. HTTP logging enables this functionality, along with a set of default settings for logging requests. Details can be customized by accessing the properties of each Web site. The default location for log files is *%SystemDrive%*\Inetpub\Logs\LogFiles. Figure 2-6 shows a list of fields that can be included in the log files.

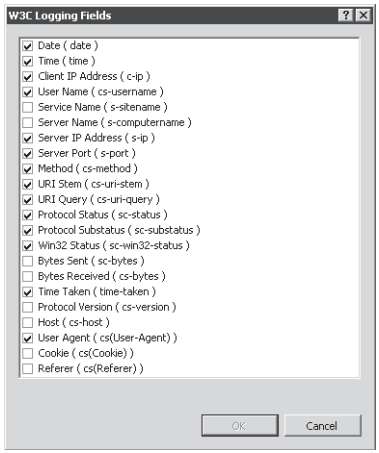

Figure 2-6 Configuring logging options

- **Logging Tools** Raw HTTP request logs are difficult to view and analyze manually. On busy Web servers, the files can get extremely large quickly. Because the content typically is organized with a single row per request, administrators might need to search through thousands of rows to get the information they need. The Logging Tools role service provides simple utilities for accessing and analyzing log files.
- **Request Monitor** A common difficulty with diagnosing performance-related issues on a Web server is that of trying to determine which activity is occurring currently. The Request Monitor feature enables administrators to see which requests are executing within the Web server process currently. This can help isolate the potential source of slowdowns or loss of service due to long-running requests or other issues.
- **Tracing** When an error or performance-related issue occurs on a Web server, it is useful to collect as much information as possible about the problem. Unfortunately, due to performance requirements, it's usually impractical to store details about all requests. Tracing functionality enables IIS to store detailed information for any failed requests. This feature works by keeping information about executing requests in memory just long enough to determine whether it was successful. If it was not, the results can be stored on the Web server for later analysis.
- **Custom Logging** The HTTP Logging feature provides a default text-based format for storing Web request information. Although this can meet the basic needs for most Web sites and services, organizations can also create their own COM-based modules, using the Custom Logging option. Developers will need to build the logging module and then register it with IIS for it to store data. This approach provides the greatest flexibility in determining which details are important to record.
- **ODBC Logging** Although storing data in a text file is an efficient method of logging requests, it makes the process of analyzing and reporting on Web server performance

difficult. The ODBC Logging role service enables applications to store Web request data in any format that is supported by an Open Database Connectivity (ODBC) connection. Examples include relational database servers such as Microsoft SQL Server and file-based formats such as Microsoft Excel. It is important to note, however, that logging to ODBC-based sources can cause significant processing and storage overhead, especially on busy Web servers.

By default, all these features are enabled except for Custom Logging and ODBC Logging. Web administrators often use log analyzer applications to process the text-based log files that store request information. Details can be used to isolate problems (such as erroneous links or missing content) as well as to analyze traffic and the popularity of specific Web pages.

Security Features

Maintaining security for Web sites, Web applications, and Web services is an important concern with all Web servers. Depending on the specific deployment and usage configuration, organizations can enable a wide variety of security mechanisms. The Security role services that are available for IIS include:

- Basic Authentication
- Windows Authentication
- Digest Authentication
- Client Certificate Mapping Authentication
- IIS Client Certificate Mapping Authentication
- URL Authorization
- Request Filtering
- IP and Domain Restrictions

Selecting and implementing these security mechanisms is covered in Chapter 3, "Managing Web Services Security."

Performance Features

Organizations often find that they receive a large volume of activity on their production Web servers, so it is fundamental for all types of Web servers to be able to service a large number of requests in a given amount of time. IIS includes numerous architectural features that help make the servicing of Web requests as efficient as possible. In addition, the Performance role services section includes two additional options:

- **Static Content Compression** The HTTP protocol provides a method by which unchanging Web pages can be compressed before they are sent to clients' Web browsers. The Web browser expands the information and renders the Web page. This method can save significant bandwidth with a minimal cost to CPU performance on the client and the

server. In addition, IIS has the ability to store frequently accessed static content in memory, further increasing performance and scalability. This feature is enabled by default and will work automatically as long as users' Web browsers support HTTP compression.

- **Dynamic Content Compression** Dynamic content usually results in different information being sent to different users. Because dynamic content often changes for each request that is made to the Web server, the amount of processing overhead for compressing the data can be significant. Dynamic content compression is disabled by default, but it can be added to help reduce bandwidth consumption for Web applications.

In general, bandwidth is more limited than is processing power on modern servers. Therefore, unless an organization has a specific reason to disable it, it is recommended that static content compression remain enabled.

Management Tools

The Management Tools section provides administrators with the ability to determine which programs will be available for working with IIS. By default, only the primary administration tool, the IIS Management Console, is installed along with the Web Server (IIS) role. This tool provides a graphical method of configuring and managing IIS Web services. You can choose to remove the IIS Management Console if you will be managing the server remotely or if your corporate security policy requires it.

The other available Management Tools options include IIS Management Scripts and Tools, which allows for command-line administration of IIS, and the Management Service, which enables you to administer IIS remotely using the IIS Management Console.

An important design goal for IIS 7.0 was to provide support for IIS 6.0–based Web applications. Although many applications can be moved directly to IIS 7.0, there are several backward-compatibility features that are included as role services:

- IIS 6 Management Compatibility
- IIS 6 Metabase Compatibility
- IIS 6 WMI Compatibility
- IIS 6 Scripting Tools
- IIS 6 Management Console

You'll learn more about these features and how you can use them in Lesson 2, "Configuring Internet Information Services."

Installing the Web Server (IIS) Role

Although there are numerous available features and options for the Web Server (IIS) role, installing the appropriate options is a simple task. Adding this role is the basis for providing Web server functionality. Components of IIS are also required by several other features and options that are part of Windows Server 2008. To begin the server role setup process, open Server Manager, right-click Roles, and select Add Role. On the Select Server Roles page, choose the Web Server (IIS) role and then click Next to begin the configuration process. (See Figure 2-7.)

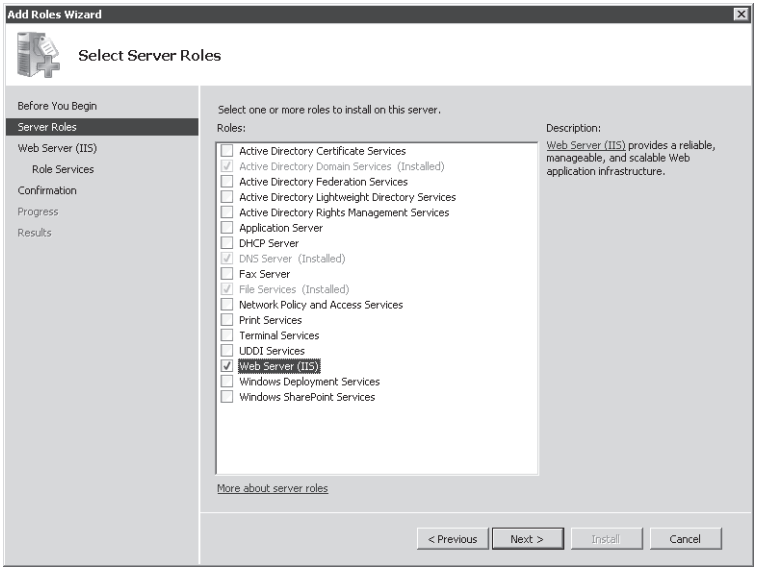

Figure 2-7 Selecting the Web Server (IIS) server role in the Add Roles Wizard

The Add Roles Wizard will evaluate the configuration of the local computer automatically and determine whether any additional role services are required. For example, if the Windows Process Activation Service has not yet been installed, you will be prompted to add it.

The Web Services (IIS) step provides some introductory information about IIS. The note also provides information about installing WSRM to ensure performance if the computer will be servicing multiple roles.

The Select Role Services page enables you to decide which components of IIS will be installed as part of the role setup process. (See Figure 2-8.) The default options provide a minimal set of features for the core Web server role. As described later in this section, you can also add or remove role services after the Web Server (IIS) role has been enabled. Because some role features depend on other features, you might be prompted to add those dependencies when selecting an item.

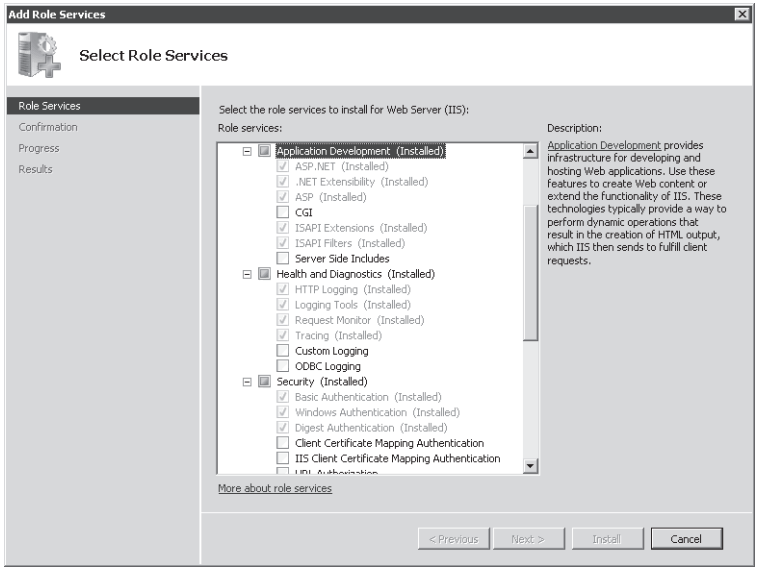

Figure 2-8 Selecting roles services for the Web Server (IIS) role

The Confirm Installation Selections page will provide you with a list of the configuration settings and role services you have chosen. Once you review the list and click Finish, the installation process will begin. Depending on which role services you've selected, the setup process might take significant time, require a reboot of the computer, or both. If a reboot is required, the Add Roles Wizard will resume from its previous ending point after you log on to the server again. Finally, on the Installation Results page (shown in Figure 2-9), you will see a confirmation of which features have been installed and any additional information that should be noted.

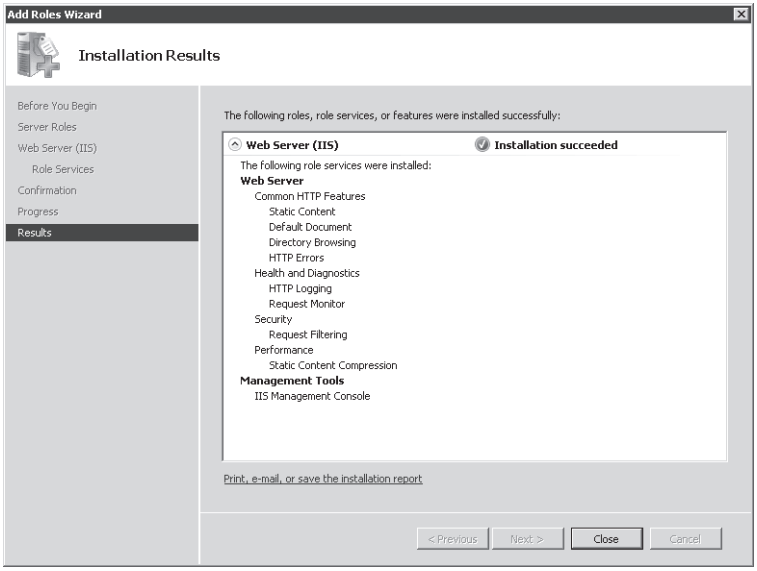

Figure 2-9 Viewing the installation results for adding the Web Server (IIS) server role

Verifying the IIS Installation by Using Server Manager

Once you have installed IIS, there are several ways in which you can verify that the Web Server processes are working properly. The first is by using the Server Manager tool. Expand the Roles section and then click Web Server (IIS) to view the relevant details. This page provides information on any event log items that need attention. In addition, it lists the services that have been installed, along with their current state. (See Figure 2-10.) The specific list of included items will vary based on which role services and dependencies you have installed. The World Wide Web Publishing Service (W3SVC) component is the main process responsible for responding to Web requests.

Server Manager also shows information about which role services have been installed for the Web Server. (See Figure 2-11.) You can use the Add Role Services and Remove Role Services links to make changes to the configuration.

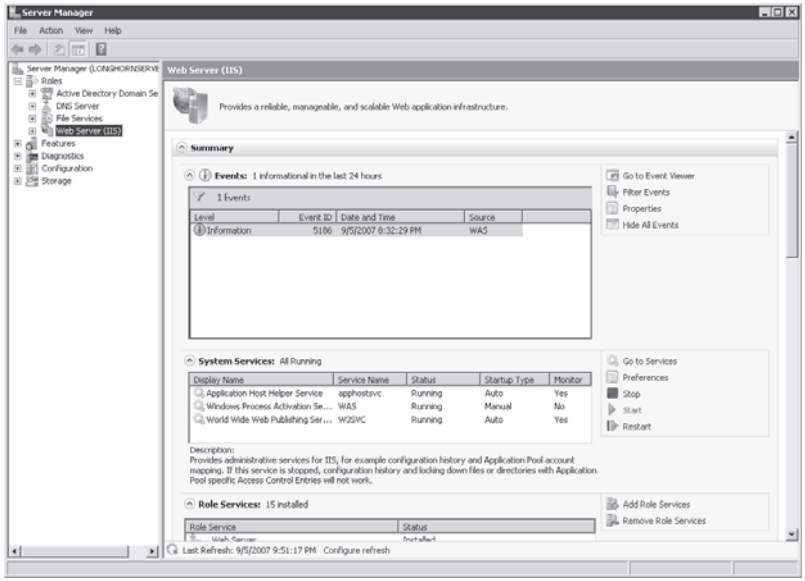

Figure 2-10 Viewing the status of the Web Server (IIS) role in Server Manager

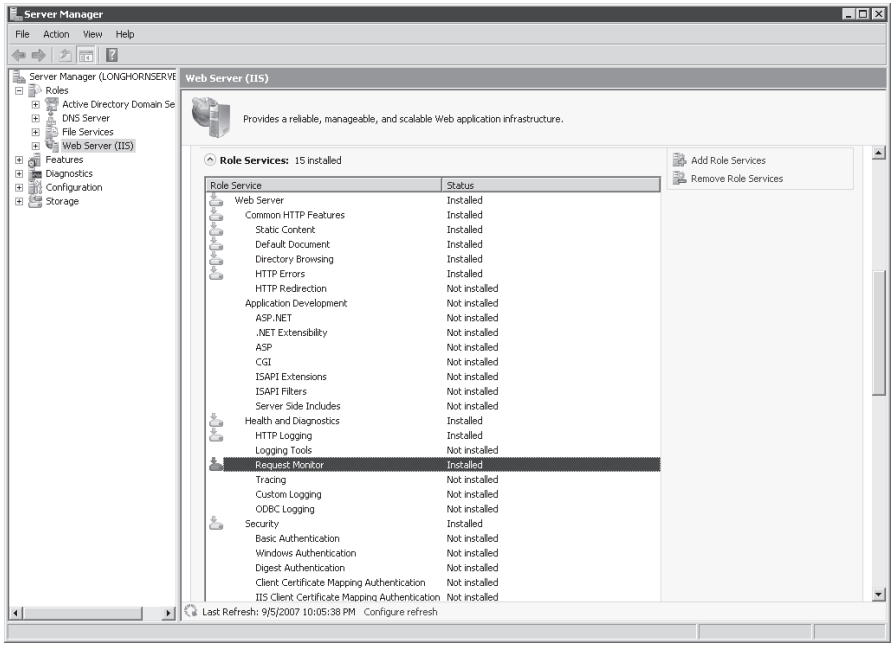

Figure 2-11 Viewing a list of installed role services in Server Manager

Finally, the Resources And Support section shows recommendations and other detailed information that can be helpful when you first set up IIS and the Web Server role on a computer. You will learn more about these options in Lesson 2. Links are also available to various online resources for learning more about IIS.

Verifying the IIS Installation by Using Internet Explorer

When you add the Web Server (IIS) role to a computer running Windows Server 2008, a default Web site that is configured to respond on HTTP port 80 is created automatically. The default location for this site is the *%SystemDrive%*\Inetpub\wwwroot folder. The default content includes only a simple static HTML page and an image file.

Because the purpose of IIS is to serve Web pages, a good way to verify that it is working properly is to launch a Web browser and connect to the local computer. You can use the built-in local alias by browsing to *http://localhost*, or you can use the local computer's fully qualified name (for example, *http://server1.contoso.com*). Using either method, you should see the default welcome page, as shown in Figure 2-12. When you click a language, the links will take you automatically to the *http://www.iis.net* Web site (assuming that the server has access to the Internet).

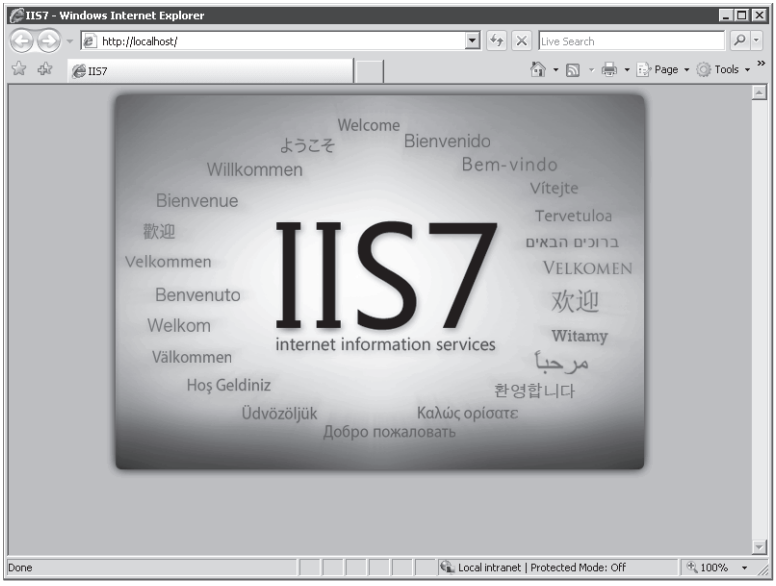

Figure 2-12 Viewing the default IIS Web site, using Internet Explorer

It is also a good idea to attempt to access the IIS Web site from a remote computer. Just open any Web browser and connect to the fully qualified address of the Web server. If you are

unable to connect, some of the likely problems are Domain Name System (DNS) name resolution issues or firewall configuration problems.

Managing Role Services

The modular architecture of IIS enables you to add or remove role services quickly and easily after the Web Server (IIS) role has been enabled on a computer running Windows Server 2008. The most common reasons for changing the role service configuration is to support a new type of Web application or Web service. You can also remove unnecessary services if they are no longer needed or the technical requirements have changed. Because the removal or addition of a role service affects the configuration of the entire server, make sure to consider the potential effects on all the Web sites on the server.

To do this, open Server Manager, expand Roles, right-click Web Server (IIS), and choose either Add Role Services or Remove Role Services. The dialog box will show which components are installed. (See Figure 2-13.) The checkmark means that an item (or an item and all its children, if there are any) have been installed. A cleared checkbox indicates that the item has not been installed. A dimmed box means that some of the role services components have been installed.

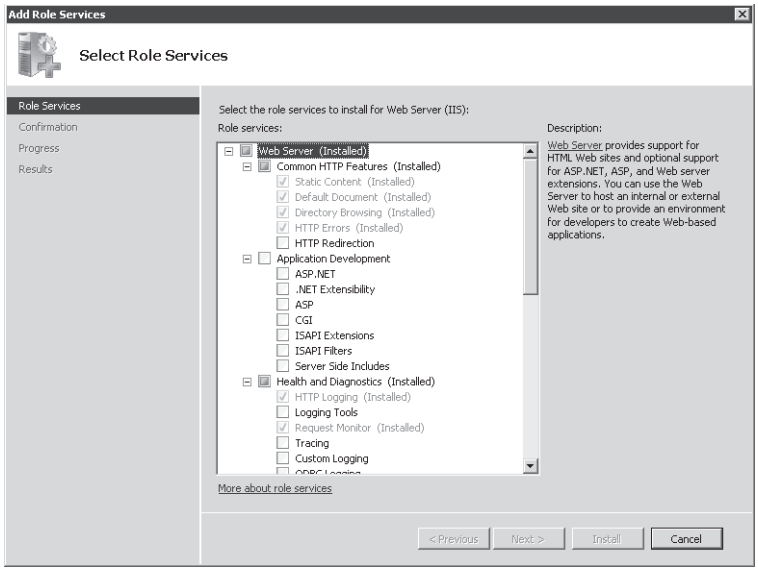

Figure 2-13 Using the Add Role Service Wizard for the Web Server (IIS) server role

When you add or remove role services, you'll receive a confirmation message, and then the process will continue. (See Figure 2-14.) If a reboot of the computer is required, the configuration process will resume automatically whenever you next log on to the computer.

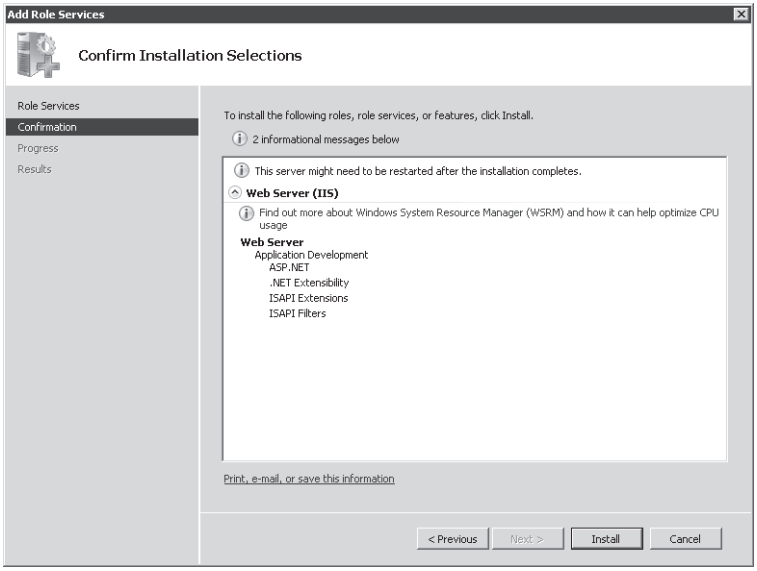

Figure 2-14 Adding a role service using Server Manager

Using Command-Line and Automated Installation Options

Organizations that rely on IIS often need to deploy many different installations of IIS. Although you can perform the process locally on each server, it is often more efficient to create scripts or commands for performing the necessary steps. There are several methods of performing automated and command-line–based installations.

The ServerManagerCmd.exe utility can be launched to install the Web Server (IIS) server role from the command line. For example, the command ServerManagerCmd.exe –install Web-Server will attempt to install the default Web server components. You can use the ServerManagerCmd.exe –query command to view which roles and features have been installed on the local computer. (See Figure 2-15.) This can be helpful when you want to collect complete configuration information quickly to determine whether changes are required to support a new Web application. For more information about using this command, type **ServerManagerCmd.exe -?** at a command prompt. You can also use this command to add or remove features such as WSRM.

Another option for performing a command-line installation of the Web Server (IIS) server role is to use the Windows Package Manager (PkgMgr.exe) utility. Windows Package Manager uses an XML file to store details about which features and options should be included in the IIS installation. For more information about using this utility, type **PkgMgr.exe -?** at a command prompt.

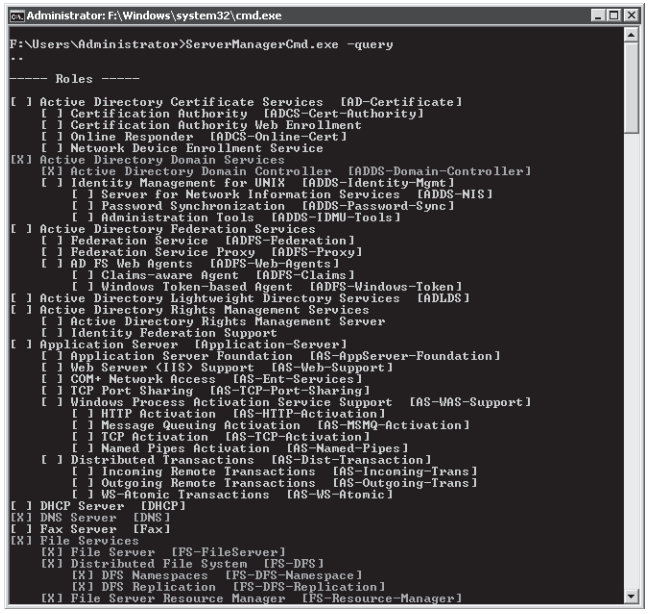

Figure 2-15 Viewing a list of installed role services and features, using ServerManagerCmd.exe

In Lesson 2, you will learn about how to use other commands to further configure IIS by using the command line or from within scripts.

Removing the Web Server (IIS) Role

If you no longer require an installation of Windows Server 2008 to serve as a Web server, you can remove IIS and all its related components by using the Remove Roles command in Server Manager. Keep in mind, however, that many different components and features of the operating system might require the Web Server to be installed. These dependent features will either be removed or the dependent functionality will be made available. Figure 2-16 shows the Confirm Removal Selections page.

Depending on which features were installed, it might be necessary to restart the computer during the removal process. If that is necessary, the process will automatically resume whenever a user next logs on to the computer.

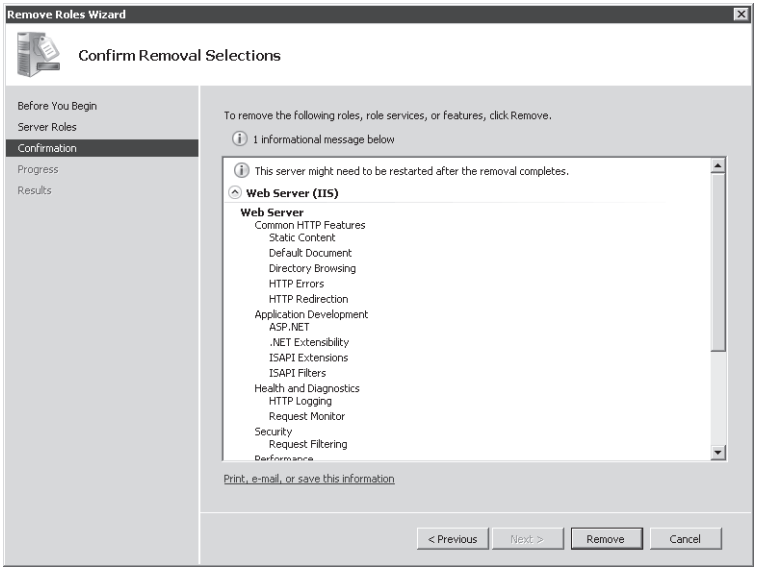

Figure 2-16 Confirming the removal of the Web Server (IIS) role

Removing the Web Server (IIS) role will remove all the binary files and role services that are associated with the Web server. The basic server configuration, including the list of Web sites and their settings, will be retained if you choose to reinstall the Web server role. Actual Web site content will not be deleted automatically. If you are planning to remove Web services permanently from the server, manually delete any remaining Web pages and data that are no longer required.

Quick Check

1. What are two methods by which you can verify a successful installation of the Web Server (IIS) role?
2. When can you add role services to the Web Server (IIS) server role?

Quick Check Answers

1. You can use Server Manager to verify that the proper services have been installed and started, and you can use Internet Explorer or another Web browser to verify that the default Web site is responding.
2. You can add the role services when you initially add the server role, or you can add them after the Web Server (IIS) role has been enabled.

Practice: Installing and Verifying the Web Server (IIS) Role

In these practice exercises, you will walk through the steps of installing the Web Server (IIS) server role on the server2.contoso.com server. The steps assume that you have not yet added this role to the computer. You must complete Practice 1 before performing the steps in Practice 2.

▶ Exercise 1: Installing the Web Server Role

In this practice exercise, you will perform the steps required to add the Web Server (IIS) server role. You will install the service with only the basic role services that are enabled by default.

1. Log on to server2.contoso.com, using an account that is a member of the local Administrators group.
2. Open Server Manager. Right-click Roles and select Add Roles to open the Add Roles Wizard. Click Next on the Before You Begin page if it is displayed.
3. On the Select Server Roles page, select the Web Server (IIS) server role. If any required dependencies are detected, choose to add them automatically. Click Next.
4. On the Web Server (IIS) page, read the basic introductory information about IIS. Note that you can use the Additional Information links to learn more about IIS and related components. Click Next.

   On the Select Role Services page, the default selections will include those components that are part of the basic Web Server (IIS) role. Note that you can obtain more information about each item in the list by selecting it and reading the text on the right side of the page. Links to additional information in the help file are available for most items. For the purpose of this exercise, keep only the default options selected and then click Next.
5. On the Confirm Installation Selections page, verify the role service selections that will be included. Optionally, you can choose to print, e-mail, or save the information to keep a record of which components were installed. When you are ready to begin the installation process, click Install.
6. When the installation process has completed, verify the installed roles and services on the Installation Results page. To complete the process, click Close.
7. When finished, close Server Manager.

▶ Exercise 2: Verifying the IIS Installation

In this practice exercise, you will verify the installation of the Web Server (IIS) role that you added to server2.contoso.com in Practice 1. Specifically, you will use both Server Manager and Internet Explorer to ensure that IIS is working properly.

1. Log on to server2.contoso.com, using an account that is a member of the local Administrators group.
2. Open Server Manager. Expand Roles and then click Web Server (IIS).

   You will see a summary of information about the Web Server role. The Events section will display any important messages that are related to the Web Server (IIS) server role.

3. In the System Services section, verify that the World Wide Web Publishing Service (W3SVC) is started. You will also see the Application Host Helper Service (apphostsvc) and the Windows Process Activation Service (WAS). If either of these services is stopped, click it and choose to start it.
4. In the Role Services section, view a list of the installed items and verify that all the default options have been installed. (The list of default role services is provided in the text of Lesson 1, "Installing the Web Server Role.")
5. Close Server Manager and open Internet Explorer. In the Address box, type **http://localhost** and then press Enter. You should see the default IIS welcome page.
6. In the Internet Explorer Address box, type the URL **http://server2.contoso.com** and press Enter. You should again see the IIS welcome page. Close Internet Explorer.
7. When finished, close Server Manager.

Lesson Summary

- The Web Server (IIS) role is designed to provide access to Web site content, using the HTTP protocol.
- The Application Server role provides support to applications that require features of the .NET Framework 3.0, COM+, and distributed transactions.
- Windows System Resource Manager (WSRM) can be used to assign resource allocation rules to various workloads and services such as IIS.
- IIS 7.0 role services include features for application development, health and diagnostics, security, performance, and management.
- You can use Server Manager to add the Web Server (IIS) server role and to manage role services.
- You can verify the installation of IIS by using Server Manager or by browsing to the default Web site, using Internet Explorer.
- ServerManagerCmd.exe and PkgMgr.exe can be used to perform automated, command line installations of the Web Server (IIS) role.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 1, "Installing the Web Server Role." The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the "Answers" section at the end of the book.

---

1. You are a systems administrator who is attempting to troubleshoot a problem with accessing a Web site on a computer running Windows Server 2008. In the past, users have been able to access the Web site by using *http://hr.contoso.com*. However, when they attempt to access the site now, they receive the error message "Internet Explorer Cannot Display The Web page." Which of the following steps should you take to resolve the error?
   - **A.** Using Server Manager, add the HTTP Errors server role.
   - **B.** Using Server Manager, verify that the World Wide Web Publishing Service has been started.
   - **C.** Verify the configuration of the users' Web browsers.
   - **D.** Using Server Manager, add the HTTP Logging server role.
   - **E.** Using Server Manager, verify that the IIS Admin Service has been started.

Lesson 2: Configuring Internet Information Services

After you have installed the Web Server (IIS) role, you will likely need to create and manage Web sites and enable specific features that are required by your applications. The details of these tasks will be based on the type of Web services you require and the way in which IIS will be used. Considerations include migrating Web sites from previous versions of IIS and managing multiple sites and applications on the same server. Fortunately, IIS includes several useful management tools and methods for simplifying administration. In this lesson, you'll learn about how to manage Web sites and server settings for the Web Server (IIS) role in Windows Server 2008.

MORE INFO Securing IIS

One of the most important considerations for production Web servers is that of managing security settings and permissions. This lesson focuses on configuring Web applications and features other than security. For more information about authentication and authorization approaches, see Chapter 3.

After this lesson, you will be able to:

- Use the IIS Manager utility to connect to and manage server settings for the Web Server role.
- Create and configure settings for Web sites, including site bindings.
- Create and manage new Web applications within Web sites.
- Describe the purpose of application pools and manage application pool settings for Web sites and Web applications.
- Create and manage virtual directories.
- Use AppCmd.exe to perform common IIS Web server administration tasks.
- Describe how IIS 7.0 manages configuration settings stored in the ApplicationHost.config and Web.config files.
- Provide support for migrating applications from IIS 6.0.

Estimated lesson time: 60 minutes

Working with IIS Management Tools

As you learned in Lesson 1, IIS includes many features and options that can be enabled to meet technical and business requirements. The *Internet Information Services (IIS) Manager* utility is the primary tool that you will use to configure and manage Web sites and their related settings. It is automatically installed when you add the Web Server (IIS) server role to a computer running Windows Server 2008 using the default options. You can launch it by selecting Internet Information Services (IIS) Manager from the Administrative Tools program group.. Figure 2-17 shows the user interface.

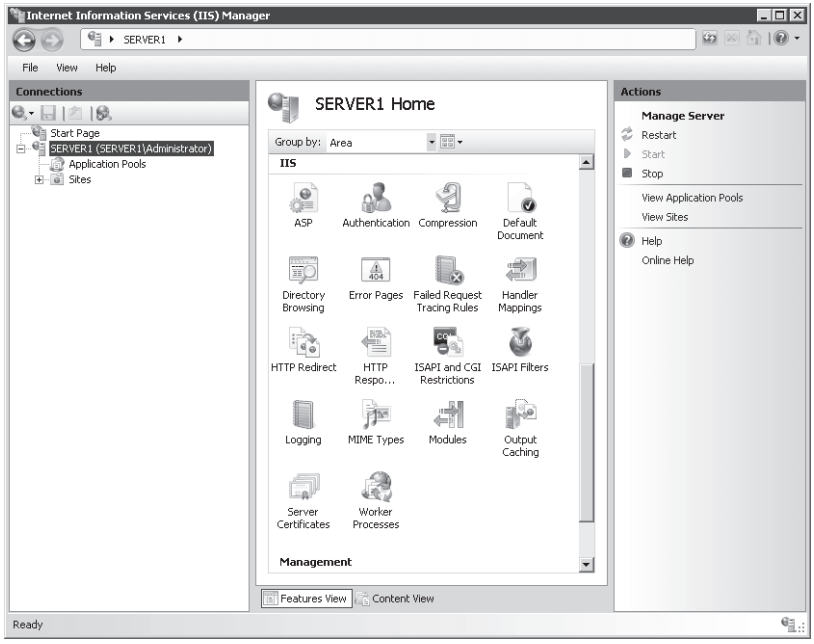

Figure 2-17 Using the IIS Manager console to connect to the local server

By default, IIS Manager will connect to the local server. This will enable you to make configuration changes to the server and other settings for this computer. IIS Manager has been designed to provide a vast array of information, using simple and consistent user interface features. The left pane shows information about the server to which you are connected. You can expand these branches to view information about Web sites and other objects that are hosted on that server. Some items contain additional commands that are available by right-clicking the object name.

Using the Features Views

The center pane of the display provides details and options that are related to the selected item in the left pane. Two main views can be selected at the bottom of the screen. Features View shows a list of all the available settings that can be configured for the selected item. The specific list of items will vary based on which role servers you have added to the server's configuration. The Group By drop-down list enables you to specify how you want the various items to be displayed. The options are:

- **No Grouping** All items are displayed alphabetically in a single list.
- **Category** Items are grouped based on their functional areas (for example, Performance and Security).
- **Area** Items are groups based on the configuration areas that they will affect.

Figure 2-18 shows the items that are displayed when the server item is selected in the left pane and when the Category grouping is selected. In addition to these options, you can display the items by using Details, Icons, Tiles, or List options. The overall layout is similar to that of Windows Explorer. It is designed to organize and display a large number of settings in a way that is easy for systems administrators to understand and manage.

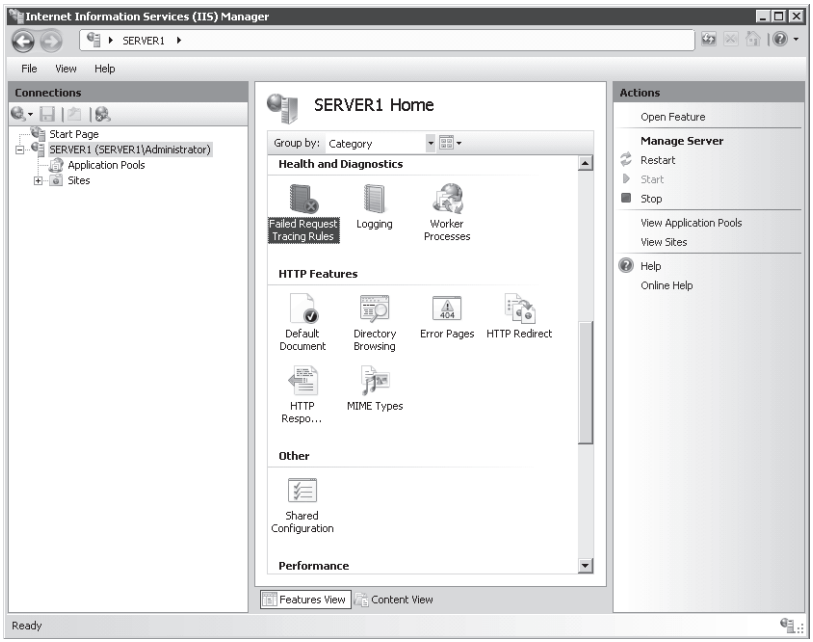

Figure 2-18 Viewing IIS Manager configuration items grouped by category

Double-clicking on specific features will load a separate options page that enables you to modify those settings.

Exam Tip Learning about the many features and options that are part of the IIS platform can be daunting, especially if you're not already familiar with Web development and management. Often, a picture can be worth a thousand words (and can help you remember available options and settings when you're taking Exam 70-643). For that reason, there are plenty of screen shots in this lesson. There's no substitute for doing, so a good way to prepare for the exam is simply to access the various properties pages for the many features and role services that are available. Having seen these options can be helpful when deciding how best to meet specific requirements, both on the exam and in the real world.

Using the Content View

Content View is designed to show the files and folders that are part of a Web site. It displays details in a Windows Explorer format and offers the ability to filter and group the list of files. (See Figure 2-19.) Content View is most useful when you are managing site content rather than site settings. It is also similar to default display in the management tools from previous versions of IIS.

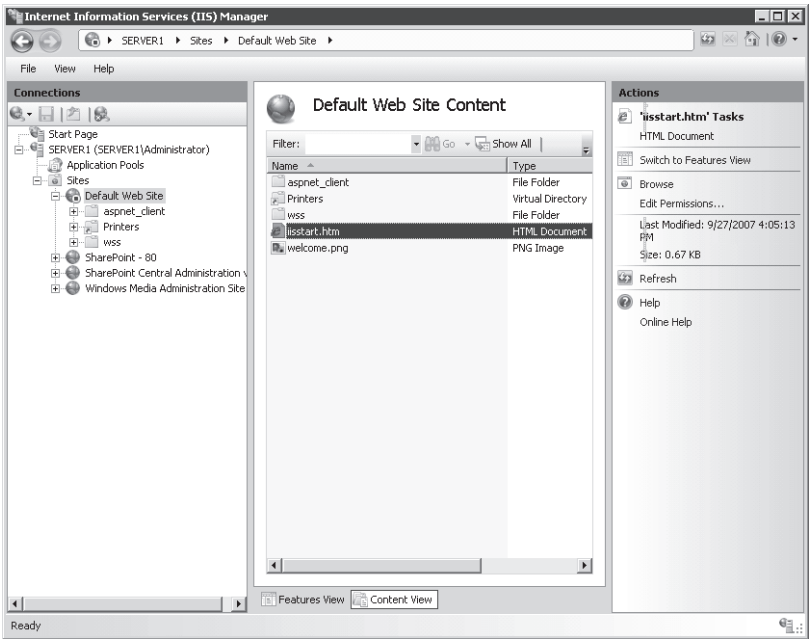

Figure 2-19 Using Content View in IIS Manager

MORE INFO Transitioning from IIS 6.0

If you're moving to IIS 7.0 after having worked with IIS 6.0, rest assured that all the functionality that you're used to seeing is still here. Roughly speaking, the Features View is a replacement for the properties pages that were available for configuring an IIS 6.0 Web server. Content View shows the information about the files and folders within each of the selected Web sites and directories in a way that is similar to the right-side pane in IIS 6.0. The goal in IIS 7.0 is to organize the presentation of a wide range of options without overwhelming systems administrators.

Using the Actions Pane

The right side of the IIS Manager screen displays the Actions pane. The specific commands that appear here are context-sensitive. For example, when you select a Web site, you will see actions for browsing to the Web site and for stopping, starting, or restarting the Web site. (See

Figure 2-20.) Furthermore, when you are changing settings for specific features, you will generally find Accept and Cancel links within the Actions pane.

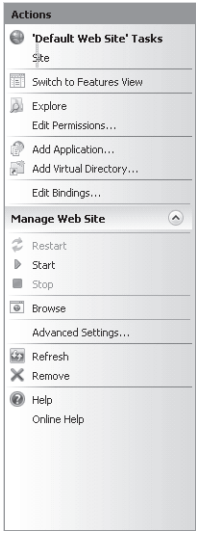

Figure 2-20 Viewing commands for managing a Web site in the IIS Manager Actions pane

Creating and Configuring Web Sites

Although some Web servers might be responsible primarily for hosting only a single Web site, it is much more common for a single IIS server to host many different Web services and applications. Before you learn about how to administer IIS, it is important to understand how the different Web server components and objects fit together.

Understanding Sites and Site Bindings

Web sites are the top-level containers that provide access to Web content. Every Web site must map to a physical path on the server. Generally, this path will contain the root folder for all content that will be available to users who access the site.

The configuration of the Web site specifies which protocols, ports, and other settings will be used to connect to the Web server. This information is collectively known as a *site binding*. Each site can have multiple bindings, based on the needs of the server. The details that can be specified in a site binding include:

- **Type** This option specifies the protocol that will used to connect to the Web server. The two default options are HTTP and HTTPS.

MORE INFO Supporting other protocols

One of the benefits of the WAS is that it enables IIS 7.0 to create sites that respond to protocols other than HTTP and HTTPS. For the purpose of taking the exam (and the content in this chapter), you will learn primarily about working with the two most common Web server protocols. When supporting distributed applications, such as those that use the WCF, keep in mind that IIS sites can support direct TCP connections and other methods of communications.

- **IP Address** The list of IPv4 or IPv6 address(es) on which the server will respond. If the server is configured with more than one IP address, different Web sites can be configured to respond to each of them. In addition to selecting a specific IP address, administrators can also choose the (All Unassigned) option to allow the Web site to respond to a request on any interface that doesn't have an explicit port and protocol binding.
- **Port** Specifies the TCP port on which the server will listen and respond. The default port for HTTP connections is port 80. Users who need to access Web sites on alternative ports must specify the port number in their URL. For example, the URL address *http:// Server1.contoso.com:5937* will attempt to connect to the Web server named Server1.contoso.com by using the HTTP protocol on TCP port 5937. The standard range for TCP ports is between 1 and 65535. Generally, many of the port numbers under 1024 are reserved for use by specific well-known applications, although there is no technical reason that they cannot be used for hosting a Web site.
- **Host Name** This text setting allows multiple Web sites to share the same protocol type, IP address, and port number while still allowing users to connect to different Web sites. The method works by interpreting the host header information stored in an HTTP request. Site administrators can configure their DNS settings to allow multiple domain names to point to the same IP address. The domain name information is then used by the Web server to determine to which Web site the user is attempting to connect and to generate the response from the appropriate site.

It is important to remember that the combination of site binding settings must be unique for every Web site hosted on an installation of IIS. For example, no two Web sites can respond using the same protocol, IP address, port, and host name setting. Although it is possible to create multiple sites with the same site bindings, IIS will allow only a single one of these sites to be started at a time.

Managing the Default Web Site

Initially, the Web Server (IIS) role includes a site called Default Web Site. The site is configured to respond to requests, using HTTP (port 80) and HTTPS (port 443). To view a list of the bindings, right-click the Default Web Site in IIS Manager (see Figure 2-21) and select Edit Bindings. (You can also use the Bindings link in the Actions pane to open the same dialog box.)

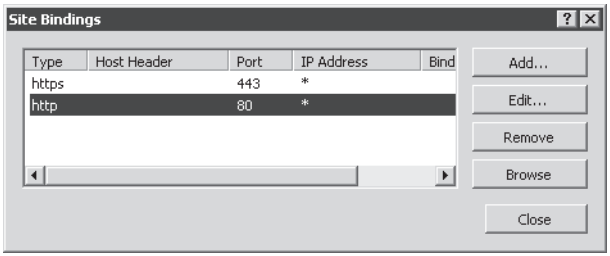

Figure 2-21 Viewing the site bindings for the Default Web Site

When you launch a Web browser and connect to a URL such as *http://server2.contoso.com*, IIS receives the request on HTTP port 80 and returns the content from the appropriate Web site.

To add a new site binding for the Default Web Site, click the Add button in the Site Bindings dialog box. As shown in Figure 2-22, you can specify the protocol type, IP address, and port information along with an optional host name. If you attempt to add a site binding that is already in use, you will be reminded that you must configure a unique binding.

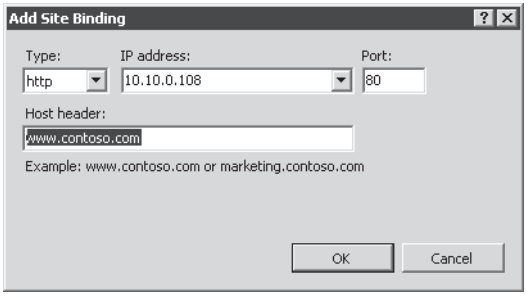

Figure 2-22 Adding a new site binding to the Default Web Site

Adding Web Sites

Start the process of adding a new Web site to IIS by right-clicking the Sites container in IIS Manager and selecting Add Web Site. Figure 2-23 shows the options that are available for the new site.

In addition to specifying the default protocol binding for the site, you will need to provide the site name. This setting is simply a logical name that will not be seen directly by users of the site. By default, IIS Manager will create a new *application pool* with the same name you provide for the Web site. You can also select an existing application pool by clicking the Select button. You will learn more about application pools and their purpose later in this lesson.

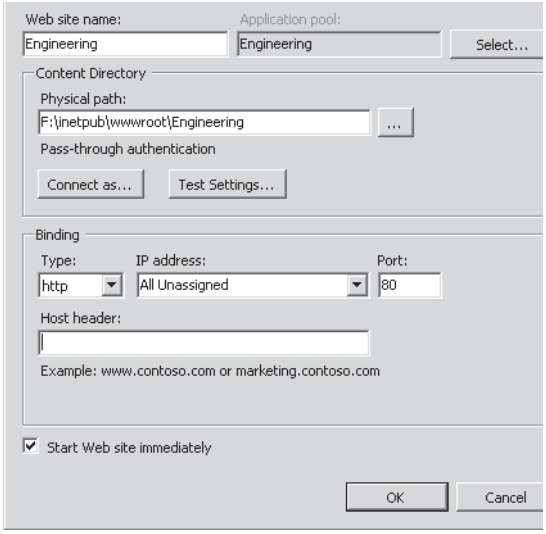

Figure 2-23 Adding a new Web site, using IIS Manager

The Content Directory section enables you to provide the full physical path to the folder that will be the root of the Web site. The default root location for IIS Web content is *%SystemDrive%* \Inetpub\wwwroot. The initial files for the default Web site are located in this folder. You should create a new folder (either within this path or in another one) to store the content of the new Web site. The Connect As button enables you to specify the security credentials that will be used by IIS to access the content. The default setting is to use Pass-Through Authentication, which means that the security context of the requesting Web user will be used. You will learn more about securing Web site content in Chapter 3.

The final check box enables you to specify whether you want the site to be started immediately after you click the OK button. Again, you will be given a warning if the Web site binding information is already in use. (See Figure 2-24.)

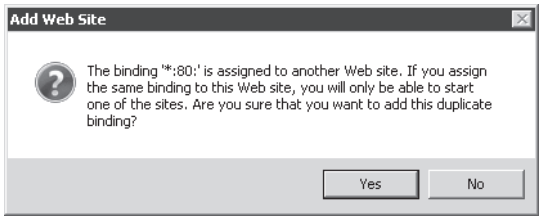

Figure 2-24 Attempting to create a new Web site, using duplicate binding information

Once you click OK to add the Web site, it will appear within the left pane of IIS Manager. Web sites can be started and stopped individually by selecting them and using the commands in the Actions pane or by right-clicking and selecting the Manage Web Site menu. Other details,

such as site bindings, can also be modified at any time. This enables you to create, reconfigure, and stop sites individually without affecting other sites on the same server. In addition to the basic site-related settings, there are some configuration settings that are defined at the site level.

Configuring Web Site Limits

Web Site Limits settings place maximum limitations on the amount of bandwidth and the number of connections that can be supported by the Web site. These settings enable systems administrators to ensure that one or more sites on the server do not use excessive network bandwidth or consume too many resources. To configure Web site limits, select the appropriate Web site and click the Limits command in the Actions pane. Figure 2-25 shows the default settings for a new Web site.

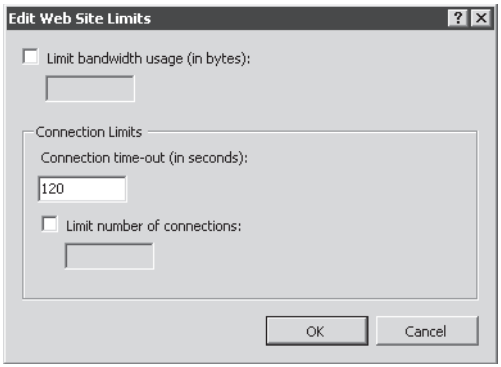

Figure 2-25 Configuring bandwidth usage and user connection limits for a Web site

The Limit Bandwidth Usage option (which is initially disabled) enables you to enter the maximum number of bytes per second that the Web server will support. If this limit is exceeded, the Web server will throttle responses by adding a time delay.

The Connection Limits section refers to the maximum number of user connections that can be active on the site. Each user connection is timed-out automatically if a new request is not received within the specified number of seconds. (The default is 120 seconds, or two minutes.) In addition, you can configure the maximum number of connections allowed for the site. If this number is exceeded, users that attempt to make a new connection will receive an error message stating that the server is too busy to respond.

Configuring Site Logging Settings

Another site-level setting is Logging. You can access these properties by selecting the appropriate Web site and, in the Features View, double-clicking Logging. Figure 2-26 shows the default options for logging.

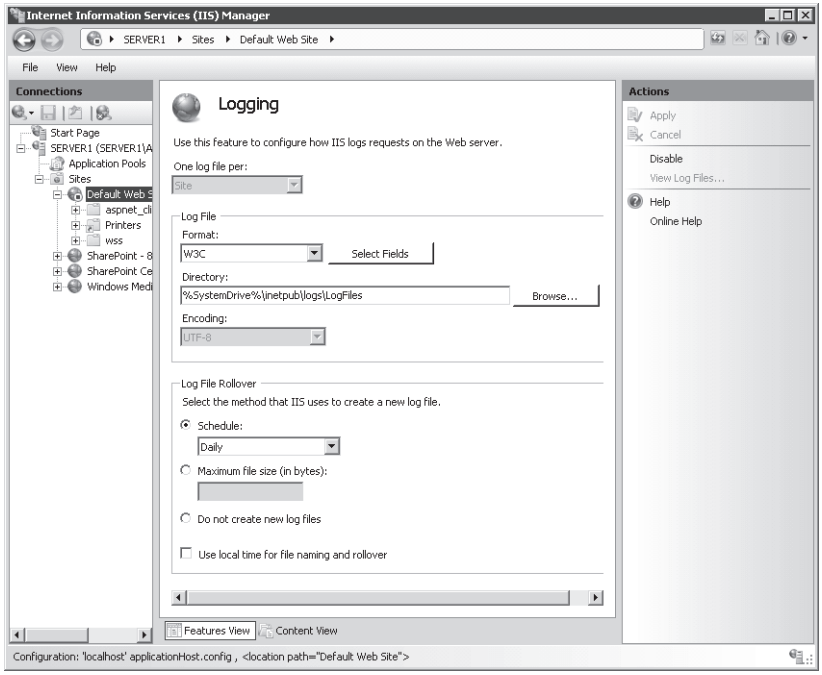

Figure 2-26 Configuring logging settings for a Web site

The specific options that are available will be based on which role services were installed for the Web server. By default, each new site is configured to store text-based log files within the *%SystemDrive%*\Inetpub\Logs\LogFiles path on the local server. Each Web site will be assigned its own folder, and each folder will contain one or more log files. You can choose from different log file formats, but the default is the W3C format, which is a standard that can be used to compare log information from different Web server platforms. The Select Fields button enables you to determine which information is stored in the log file. The default field settings are designed to provide a good balance between performance and useful information. Adding fields can affect Web server performance adversely and increase log file size, so add the information that you plan to use in alter analysis only.

On busy Web servers, log files will grow quickly. Because the log files are text-based, it can often be difficult to manage and analyze large files. The Log File Rollover section enables you to specify when IIS will create a new log file. By default, a new log file will be created daily. You can choose a different time interval, or you can specify the maximum size of each log file. There is also an option to use only a single log file. Although it is possible to obtain information by opening the log files in a text viewer such as Notepad, it is much more common to use log analysis utilities to parse the results.

Understanding Web Applications

It is common in many Web server usage scenarios for a single site to provide access to different types of content. Web applications are created within Web sites to point to the physical location of a set of content files. For example, an online news site might include two different Web applications: one for registered users and one for nonregistered users. Each Web application can point to a separate physical folder on the computer so IIS can determine how to process the requests. Web applications can also use other methods to ensure that the same content (such as news stories) is available to both sites.

Creating Web Applications

You can easily create new Web applications by using IIS Manager. Right-click the Web site within which you want to create a Web application and then select Add Application. Figure 2-27 shows the available options. The first setting option is the alias that will be used for the site. This is the name that users will type as part of their URL to connect to the content. For example, if a new Web application with the alias Engineering is created within the default Web site, visitors will use a URL such as *http://server1.contoso.com/Engineering* to access the content. You will learn about application pool setting later in this lesson.

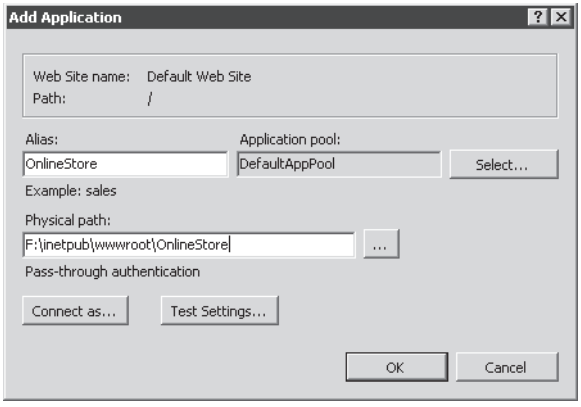

Figure 2-27 Adding a new Web application to a Web site

The Physical Path option enables you to specify the folder in which the content for the Web application will be stored. Generally, the file system location should be unique and unshared with other Web applications. As with the process of creating a site, you will be able to keep the default setting of Pass-Through Authentication or click the Connect As button to specify a username and password to use. The Test Settings button enables you to verify the connection details that you have entered (if any). The Test Connection dialog box as shown in Figure 2-28, details that if you keep the default setting, IIS Manager will be unable to verify the authorization permissions. (You will learn more about authentication and authorization in

Chapter 3.) This is because the specific user context is not defined until a user attempts to access the content.

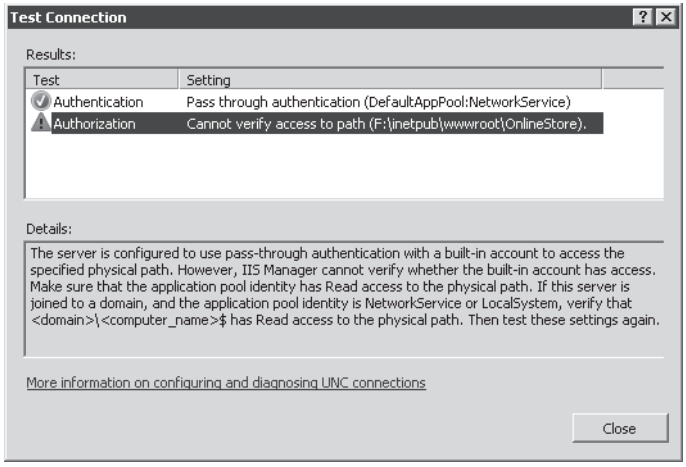

Figure 2-28 Testing physical path connection settings when creating a new Web application

To finish the creation of the Web application, click OK. You will now see a new Web application under the site object in IIS Manager. You can now also modify other settings for the Web application by using the Features View.

Managing Web Application Settings

By default, many of the settings for a new Web application will be inherited automatically from the Web site in which it was created. This enables you to use the same default settings easily for each new site. In most cases, you can also override the settings at the Web application level based on specific needs of the application. To do this, double-click on any of the items in the Features view and make the corresponding changes at the Web application level. Most of these settings will override those that are assigned for the parent site.

Working with Application Pools

One of the primary concerns with managing Web servers is the potential for one Web site or application to affect operations of others on the same computer negatively. Issues such as memory leaks or application bugs potentially can cause a loss of service or reduced performance for many different Web applications. Application pools are designed to isolate different

sites from each other so that failures and other problems can be contained. Within each application pool, worker processes are actually responsible for completing Web requests. Each application pool contains its own set of worker processes, so it is impossible for problems in one pool to affect processes in another. Application pools can also be started and stopped independently.

By default, IIS includes the Classic .NET AppPool and DefaultAppPool application pools, along with an application pool that has the same name as the application itself. Classic .NET AppPool is used to support applications that require Microsoft .NET Framework 2.0, using classic Managed Pipeline Mode (a mode that enables .NET code to use methods of intercepting and responding to requests that are being processed by IIS). DefaultAppPool, as its name implies, is used to support the default Web site. It also supports Microsoft .NET Framework 2.0, but it uses the new Integrated Managed Pipeline Mode. You will learn more about pipeline modes later in this lesson.

Earlier in this lesson, you saw options to select or create new application pools when adding sites and Web applications. By default, IIS Manager will create a new application pool when you create a new Web site. The application pool will have the same name as the site. This is the recommended approach because it allows the processes within each Web site to run independently of others. When you create a new Web application, you will have the option of selecting from any of the available application pools.

Creating Application Pools

IIS Manager includes an Application Pools object that enables you to manage application pools on the Web server. The default display will show all the application pools that currently exist on the server, along with their current status and settings. (See Figure 2-29.)

To create a new application pool, right-click the Application Pools object and select Add Application Pool. Figure 2-30 shows the available options. The name option will be used by systems administrators to identify the purpose of the application pool. If you are creating this object to support a specific Web site, include identifying information in the name. The .NET Framework version options will be based on which versions are available on the local computer. By default, the .NET Framework 2.0 and No Managed Code options are offered. The latter option specifies that .NET functionality will not be available for Web applications that are part of the pool.

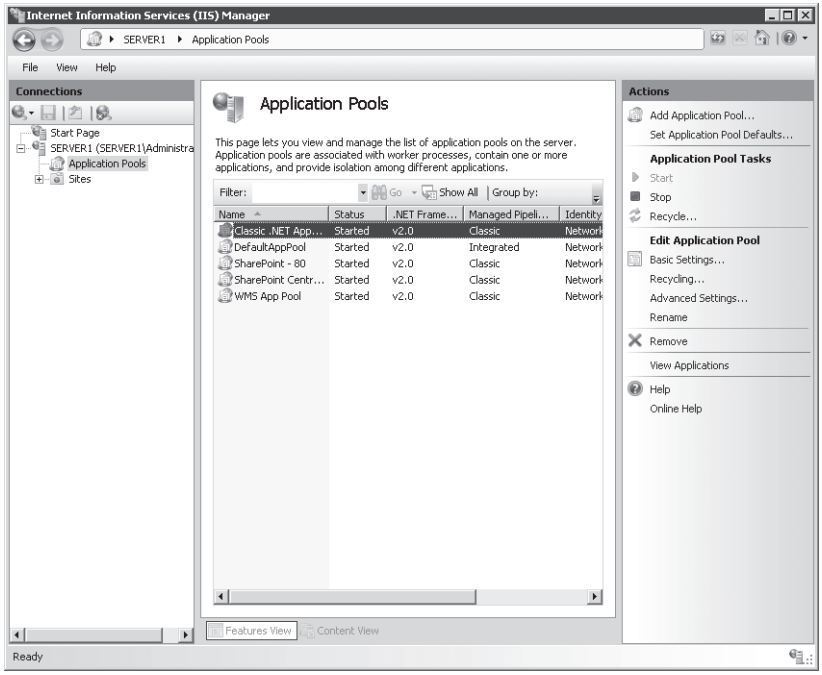

Figure 2-29 Managing application pools in IIS Manager

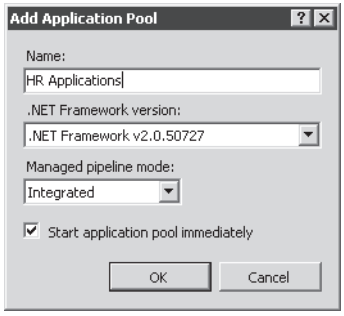

Figure 2-30 Creating a new application pool

Managed Pipeline Mode specifies the method that will be supported for code that needs to intercept and modify Web request processing. Finally, you can choose whether you want to start the application pool immediately.

Managing Application Pools

Each application pool present on a Web server can be started and stopped independently. Stopping an application pool will prevent requests from being processed by any applications that are a part of that pool. Users that attempt to access content from these sites will receive an

error message stating HTTP Error 503, "Service Unavailable." It is a good idea to verify which applications are using an application pool before you stop it. You can do this by right-clicking one of the application pool items in IIS Manager and selecting View Applications.

Configuring Recycling Settings

An alternative to stopping an application pool is to recycle it using the *Recycle* command in the Actions pane. This command instructs IIS to retire any current worker process automatically after it has executed existing requests. The benefit is that users will not see a disruption to service on their computer, but the worker process will be replaced by a new one as quickly as possible. Recycling application pools is generally done when issues such as memory leaks or resource usage tend to increase significantly over time. Often, the root cause of this problem is a defect or other problem in the application code. The ideal solution is to correct the underlying application problem. However, it is possible at least to address the symptoms by using the *Recycle* command.

In some cases, you might automatically recycle worker processes based on resource usage or at specific times. You can access these options by clicking the Recycling command under Edit Application Pool in the Actions pane. (See Figure 2-31.)

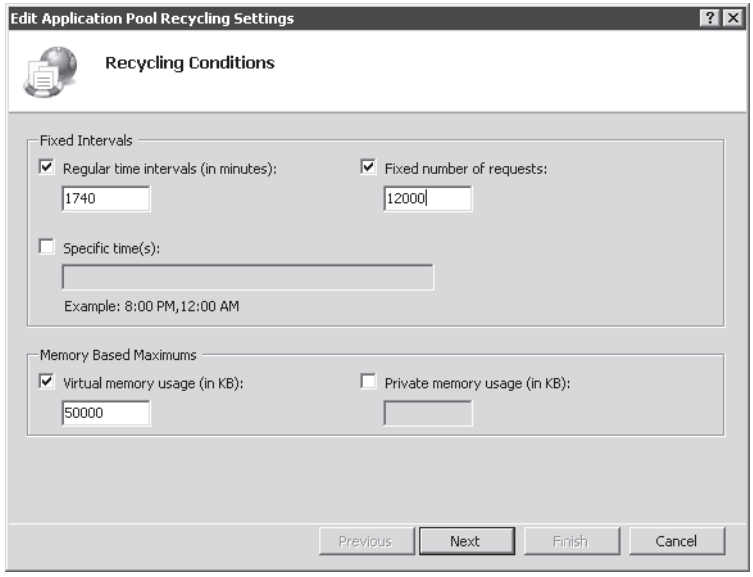

Figure 2-31 Configuring Application Pool recycling settings

The primary options for recycling settings are either Fixed Intervals (which are based on specific times or after a fixed number of requests is processed), or Memory Based Maximums. The most appropriate options will be based on the specific problems you are trying to troubleshoot

or avoid. In general, recycling application pools too quickly can reduce performance. However, if a Web application has serious problems, it is preferable to address them through recycling worker processes before users see slowdowns or errors on the Web site.

Keeping track of application pool recycle events is also an important part of ensuring that your Web server and its applications are running as expected. For example, if you set the maximum memory settings, you will likely want to know how often the application pool has been recycled. Figure 2-32 shows the Recycling Events To Log step that enables you to define which events are recorded. To view the Recycling Events To Log page click Next.

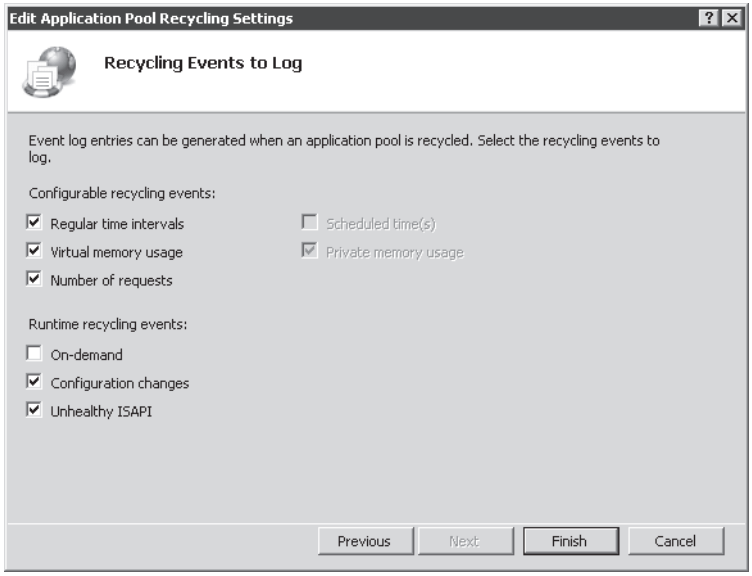

Figure 2-32 Choosing which recycling events should be logged

Configuring Advanced Application Pool Settings

In addition to the basic configuration settings and recycling options for an application pool, systems administrators can configure additional details to control the behavior of worker processes. To access these settings, select an application pool in IIS Manager and click the Advanced Settings link in the Actions pane. (See Figure 2-33.)

The options allow for setting detailed parameters related to CPU and memory resource usage. In general, you should not change these parameters manually unless you are reasonably sure of their intended effects. Some modifications can result in reducing processing speed for the applications that are part of the pool. Others can result in reserving or using too many system resources for a particular pool.

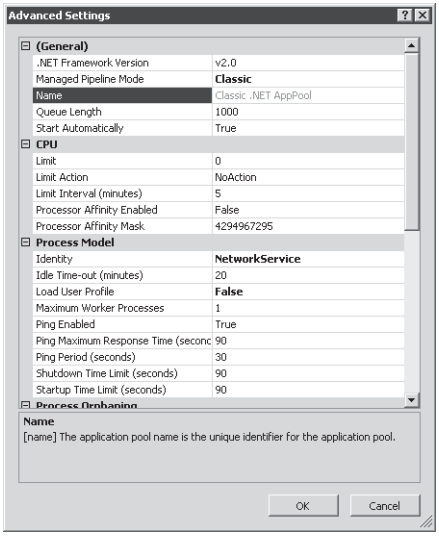

Figure 2-33 Configuring Advanced Settings for an application pool

Working with Virtual Directories

A common requirement within Web sites is to include content from folders that are located outside of the Web site's primary folder structure. For example, multiple Web sites that share the same set of images might need to access a pointer to a single path from which they can all access files. Virtual directories are designed to provide this capability. Virtual directories can be created at either the level of a Web site or within a specific Web application. They include an alias name (which will be used in the requesting URL) and point to a physical file system location path.

Creating a Virtual Directory

The process of creating a virtual directory is similar to that of creating a Web application. In IIS Manager, right-click the appropriate parent Web site or Web application and then select Add Virtual Directory. Figure 2-34 shows the available options.

You will be able to provide an alias for the virtual directory (such as Images), along with security credentials and the physical path to the virtual directory. When a request is received for this alias, IIS will look in the appropriate file system location automatically for the requested content.

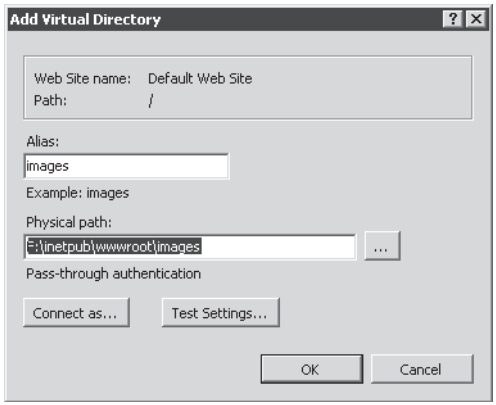

Figure 2-34 Adding a new virtual directory within a Web site

Comparing Virtual Directories and Web Applications

Although the settings for a virtual directory are similar to those of a Web application, there are some differences in their usage. Web Applications are generally designed to support executable Web code such as applications built using ASP.NET. They run within an isolated process space, using WAS. The reliance on WAS also enables Web applications to respond using protocols other than HTTP and HTTPS (assuming that other protocols have been installed and configured on the local server).

Virtual directories, on the other hand, are primarily used to point to static content that is stored in an alternate file system location. Both Web applications and virtual directories form a portion of the complete URL that is used to access a Web site. They can also both be nested to provide access to multiple levels of site content. The more appropriate choice will be based on the requirements of the Web application that you plan to support.

MORE INFO Keeping the configuration simple

Web applications and virtual directories offer a lot of power and flexibility for both Web server administrators and Web developers. In general, try to keep your configurations as simple and intuitive as possible. For example, although both types of objects can be nested within each other, complex nesting can be confusing (especially if some of the objects share the same names). Overall, keep management of the Web site in mind when creating and designing the site structure.

Using Command-Line Management

Performing simple administrator tasks on a few IIS servers is a relatively simple process, using the IIS Manager console. However, when you want to commit the same changes on many different servers, or you want to automate the configuration process by using scripting, command-line utilities can make these tasks more efficient. IIS includes an executable command,

AppCmd.exe, which provides a simple way for systems administrators to perform common operational tasks. The actual parameters are designed to map to the structure of IIS Web sites, Web applications, and virtual directories.

The AppCmd.exe file is located within the *%SystemRoot%*\System32\Inetsrv folder. You can get initial help for the utility by running the command with the -/? switch. (See Figure 2-35.) You can use the same switch to get additional details about other commands. The general syntax for the command is:

```
AppCmd.exe Command Object "ObjectName" /parameter:value
```

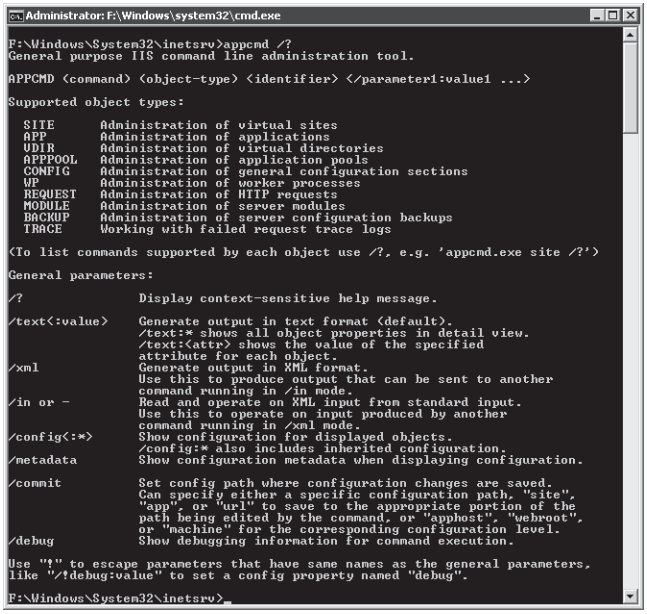

Figure 2-35 Viewing help for the *AppCmd.exe* utility

Understanding Command Options

AppCmd has been designed to use a simple set of six commands for performing tasks on objects. The list of commands includes:

- **List** Returns information about the specified object.
- **Add** Creates a new object of the type that is specified. Details can be added, using parameters and values.
- **Delete** Deletes the specified object (such as a Web site or Web application).
- **Set** Changes settings for the object, as specified by the parameters and values.
- **Start / Stop** Available for objects that support these actions (such as a Web site).

If you want to perform multiple operations (either from a script file or from the command line), you will need to call AppCmd.exe for each operation. This helps keep the syntax of the statements simple and easy to read.

Understanding Objects

In a standard AppCmd statement, you will need to provide an object type and the name of the object that you plan to use. The types of objects supported by AppCmd.exe include:

- App (Web Application)
- AppPool (Application Pool)
- Backup (Server configuration backups)
- Config (Server configuration information)
- Module
- Request
- Site (Web Site)
- Trace
- VDir (Virtual Directory)
- WP (Worker Process)

You can get more information about the parameters and values that apply to an object by typing -? after the command. Figure 2-36 shows an example.

```
Appcmd site -?
```

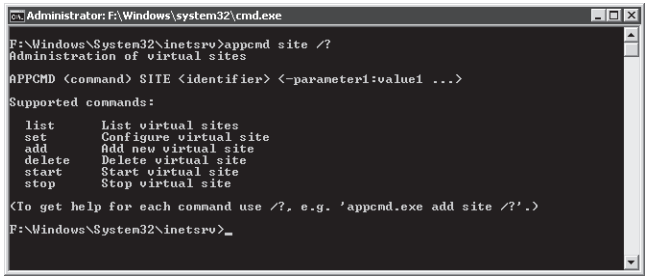

Figure 2-36 Getting help for an *AppCmd.exe* site object

Examples of Commands

The process of listing, creating, and managing IIS configuration settings by using AppCmd is generally fairly simple. Table 2-2 provides some examples of common commands and their purpose.

Table 2-2 Sample Commands for AppCmd.exe and Their Purpose

Command	Purpose
`AppCmd list site`	Returns a list of Web sites on the local server
`AppCmd add site /name:TestSite01`	Adds a new Web site called TestSite01
`AppCmd add vdir /app.name:"Default Web Site/" /path:/Images /physicalPath:"C:\Inetpub\wwwroot\images"`	Adds a new virtual directory with the alias Images and points to the specified physical file system location
`AppCmd list request`	Returns a list of currently running Web server requests
`AppCmd list config`	Returns the entire contents of the current Web server configuration in XML format

Exam Tip When preparing for Exam 70-643, it's not necessary to memorize every command-line option and parameter for utilities such as AppCmd.exe. Instead, focus on the basic syntax and the types of operations that can be performed. There's no better way to become familiar with the commands than by actually performing actions such as creating sites and changing configuration settings. This will help you identify (and rule out) answer choices when you're taking the exam. Generally, if you know what you're trying to accomplish, you should be able to identify the correct command-line option.

Using Windows PowerShell

In addition to using the AppCmd utility, Web server administrators can use the command shell and scripting language, Microsoft Windows PowerShell. Windows Server 2008 includes this functionality by default, and you can launch it by searching for Powershell in the Start menu. Windows PowerShell enables you to write and create powerful scripts for performing a wide array of operations. Although a complete description of how to use PowerShell is beyond the scope of this book (and Exam 70-643), you can find more information about using it to manage IIS by searching for Powershell at *http://www.iis.net*. The Microsoft TechNet Scripting with Windows PowerShell Web site offers tutorials and examples for creating new scripts at *http://www.microsoft.com/technet/scriptcenter/hubs/msh.mspx*.

Automation Using .NET Framework

Many Web developers already have a significant amount of knowledge about working with the .NET Framework. Therefore, it can be helpful for them to manage IIS, using standard .NET code. IIS 7.0 provides two .NET namespaces that can be used to manage IIS configuration settings programmatically. They are:

- *Microsoft.Web.Administration* This namespace provides objects and methods that are useful for managing and changing Web server settings. It is focused primarily on performing configuration changes for an IIS Web server.
- *Microsoft.Web.Management* Although the default IIS Manager user interface has been designed to provide simple access to the majority of commonly used functionality, some environments might want to create their own management extensions for performing specific tasks. The *Microsoft.Web.Management* namespace includes objects and methods that enable developers to extend the user interface functionality of IIS management tools. These additions can then be configured to run in a standalone environment, or they can be integrated with the built-in IIS Manager utility for easy access.

Understanding how to write applications, using the .NET Framework is beyond the scope of Exam 70-643, but it can be helpful to know that these options are available for automating configuration and management tasks. Additional information about the namespaces mentioned here and others can be found at the following URL *http://msdn2.microsoft.com/en-us/library/aa388745.aspx*

Managing Web Server Configuration Files

Although making configuration settings on one or a few servers is easiest using graphical tools, systems administrators often need to configure many different Web servers. In addition to using IIS Manager and related tools for configuring settings, you can also configure your Web server by using XML configuration files. In addition, by storing settings in a single file, you can back up and restore settings to other IIS installations easily. In this section, you'll learn about where Web server and Web site settings are stored.

Understanding ApplicationHost.config

All the configuration settings that have been defined for the local IIS Web server are stored in an XML text file named *ApplicationHost.config*. The default file system location for these files is *%SystemDrive%*\Inetpub\History. Within this base folder is a series of folders, each of which contains a copy of the ApplicationHost.config file. The ApplicationHost Helper Service (a default component that is included when you install the Web Server [IIS] role) automatically makes periodic backups of the configuration of the local Web server. This process automatically creates a new folder and a copy of the ApplicationHost.config file. The schema subfolder

contains a file that is used to describe and interpret the specific settings that can be used in the configuration files.

An ApplicationHost.config file can be opened and modified, using a standard text editor (such as Windows Notepad) or by using an XML-aware application (such as Visual Studio). The contents are arranged in a hierarchy that defines the various settings and options that can be configured within IIS. (See Figure 2-37.) Before you make changes directly to a configuration file, be sure to make a backup copy of it. It is fairly easy to introduce changes that can cause errors in IIS.

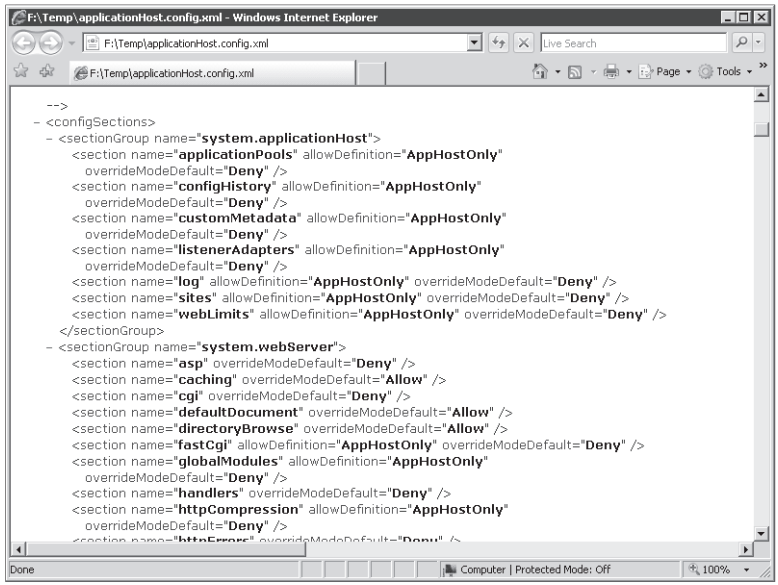

Figure 2-37 Viewing a portion of the ApplicationHost.config file, using Internet Explorer

Restoring the ApplicationHost.config File

In the event that you need to revert the configuration of IIS to an earlier state by using the automatic backup files, you can copy over the working config file manually. The active version of the ApplicationHost.config file is in the *%SystemRoot%*\System32\Inetsrv\Config folder. To roll back the configuration of IIS, find the ApplicationHost.config version you want to use and then copy it over the current file. Note that for the changes to be reflected, it might be necessary to restart the Web server and IIS Manager. It is also highly recommended that you copy the current configuration file to a backup location in the event you need to refer to it later.

Understanding Web.config Files

A common problem related to managing Web applications and Web sites is that of retaining settings as sites are moved between servers. In previous versions of IIS, it was often necessary to re-create settings manually to ensure that the site would run properly. IIS 7.0 uses a hierarchical approach to create and manage configuration settings. In addition to the server-level settings that are defined in the ApplicationHost.config file, systems administrators and Web developers can include other settings in *Web.config* files.

Web.config files can be located within the root folder for a Web site or Web application. These files can contain settings that override the default server-level settings that are included in the ApplicationHost.config file. The format of the files and options is similar. By default, a new Web.config file is created automatically whenever you add a new Web site or a new Web application. The default settings are inherited from the server-level settings unless you specifically change them.

Overall, the hierarchy for configuration files is:

1. Host (ApplicationHost.config)
2. Site (Web.config)
3. Application (Web.config)

Settings in lower-level files can override settings defined in the parent. A useful benefit of this approach is that the configuration information is included automatically when you choose to copy an entire folder of Web content to another server or to another location on the same server.

Exam Tip When making changes to IIS and Web application configurations, consider which portions of the site structure the modifications should affect. If the goal is to modify all Web sites, consider making the change in the server-level ApplicationHost.config file. Otherwise, making site-level or application-level changes will likely be more appropriate.

Migrating Web Sites and Web Applications

The presence of Web.config files within Web application and Web site folders helps make the process of migrating Web sites to different servers or physical locations simpler. For most applications, all that is required is for all the files within the appropriate folders to be moved or copied to the new location. Then, within IIS Manager, you can re-create any additional Web sites, Web applications, and virtual directories that are required. It is important, however, to test any migrated Web application thoroughly. In some cases, incompatibilities or other issues between server-level and application-level settings can have unintended consequences. Overall however, the process of moving and copying Web sites is usually fairly simple and straightforward.

Backing Up and Restoring Configuration Data Using AppCmd.exe

An important aspect of Web server administration is ensuring that the configuration of the server is protected against data loss. Because IIS configuration settings are stored automatically in the *%SystemDrive%*\Inetpub\History folder, ensure that this folder is included in file system backup policies. In addition, it's important to back up Web sites and Web applications to ensure that they can be restored quickly in the case of a failure. Often, however, you'll need to create your own configuration backups manually. For example, if you want to transfer configuration data to another IIS installation, or if you want to protect against unwanted changes, it is a good idea to make an on-demand configuration backup.

You can use AppCmd.exe to create a backup of the configuration of IIS and store it to a text file. The utility offers simple capabilities for creating a backup and for restoring from it. The standard command for adding a new backup is:

```
AppCmd add backup "BackupName"
```

BackupName specifies the name of the file that you want to create. You can leave off the name, and an automatic filename that includes a timestamp will be generated. The file will be created in the location in which AppCmd.exe was run, but you can always move or copy the file manually to another location.

You can restore the configuration information from the backup, using the following command:

```
AppCmd restore backup "BackupName"
```

This process restores the configuration of the IIS Web server to the settings that were included in the backup file. If you want to view a list of backups that have been made, you can use the following command:

```
AppCmd list backups
```

You will see a list of all the backup files you have created. Figure 2-38 shows an example of all these backup-related commands at work.

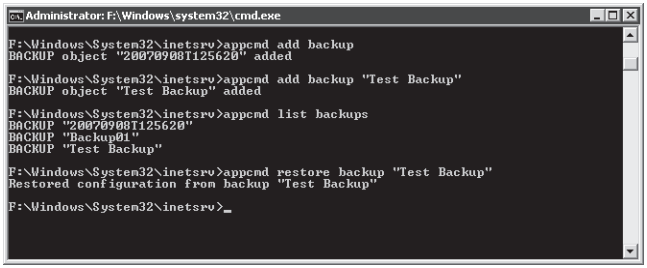

Figure 2-38 Performing IIS configuration backup and restore operations, using *AppCmd.exe*

Using Centralized Configuration for Server Farms

As organizations place a greater reliance on their Web sites and Web-based applications, the ability to improve performance, scalability, and reliability are important goals. With relation to Web servers, a common configuration is known as a *Web server farm*. In this approach, many different Web servers are configured to provide access to the same content. Generally, they have the same configuration settings and either store local copies of Web sites and applications or access them from a shared location.

From a systems administration standpoint, managing large groups of Web servers can be challenging. When configuration changes are required, they often have to be committed manually to many computers. Even with the use of automation or scripting, it is possible to overlook one or a few servers. To support the server farm usage scenario better, IIS 7.0 enables you to share centrally stored configuration data with multiple Web servers.

The first step in the process of creating a shared configuration is to export the configuration of a single IIS server. Generally, you will configure this server with all the settings that you want to use on the other servers. Then, using IIS Manager, click the server name and double-click Shared Configuration in the Features View. To generate an export, click the Export Configuration command in the Actions pane. (See Figure 2-39.) You will be able to provide a path into which the configuration files will be stored. To protect sensitive information in the configuration files, you must type and confirm an encryption key password. This password will be required to view the settings in the file. You can also use the Connect As option to provide security credentials if you are planning to store the configuration in a network location.

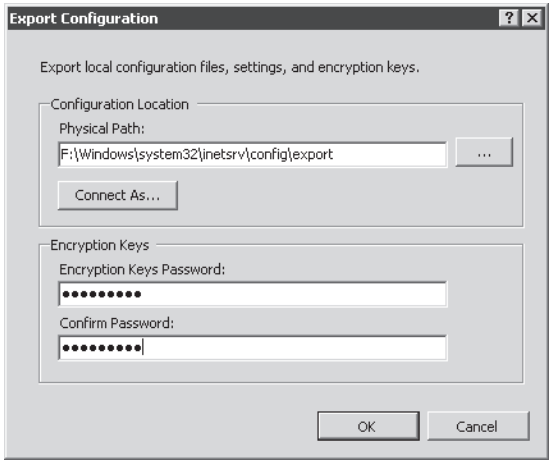

Figure 2-39 Exporting configuration information, using IIS

The second step of the process is to place the shared configuration file in a location that is accessible to all the Web servers. Usually, the best choice is a shared network folder on a reliable server. Once you know the path to the files, you can use the Shared Configuration feature to enter the details. First, select the Enable Shared Configuration check box. (See Figure 2-40.) This will enable you to specify the Physical Path setting. You can use a local file system location or a Universal Naming Convention (UNC)-based network path (for example, \\Server1\WebConfig). The User Name and Password fields enable you to enter the security credentials that will be used by IIS to connect to the physical path you have specified.

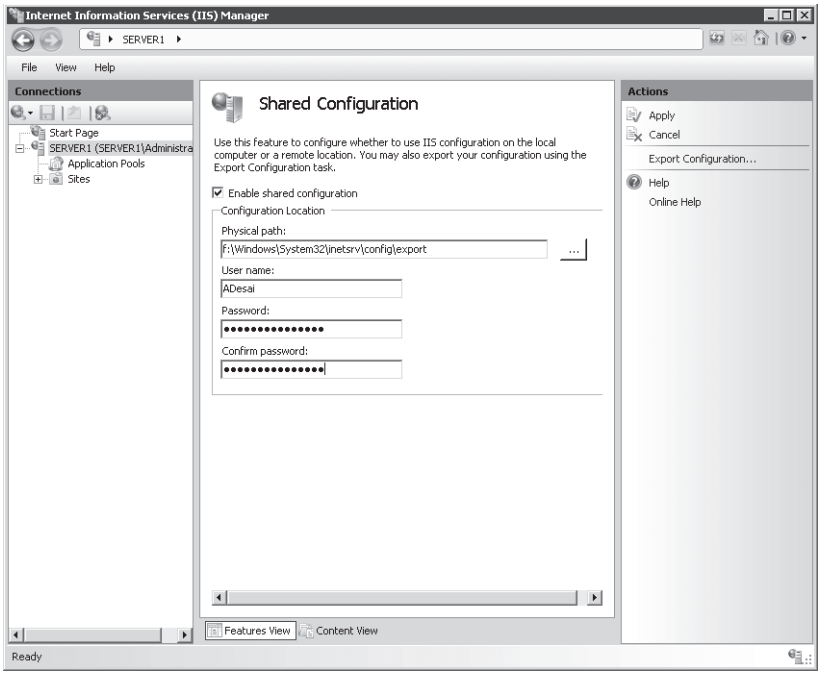

Figure 2-40 Enabling Shared Configuration for an IIS Web server

To save the settings, click the Apply command in the Actions pane. You will be prompted to enter the encryption key password for the configuration files. Once the configuration import is complete, you will be notified that you must restart IIS Manager for it to recognize the configuration changes. You can disable the shared configuration settings later by deselecting the Enable Shared Configuration check box. This returns the Web server to using locally defined configuration settings.

MORE INFO Creating production server farms

The ability to share settings easily among Web servers is helpful for setting up IIS-based Web server farms. However, sharing configuration data is only one part of an overall Web server farm configuration. Other considerations include deploying and synchronizing content updates, handling session state, managing security, implementing load-balancing, and responding to fail-over events. Rest assured, there are many good ways of addressing these challenges. However, always be sure to involve Web developers and systems administrators when designing a scale-out strategy.

Migrating From IIS 6.0

A large number of Web developers have depended on previous versions of IIS to support their Web applications and Web sites. IIS 6.0, the version included with Windows Server 2003, provided several enhancement features and capabilities over previous versions. IIS 7.0 provides even more improvements in functionality, performance, reliability, and management capabilities. However, with these new improvements, preserving backward compatibility with existing applications built for IIS 6.0 was an important goal.

For Web sites and Web applications that rely primarily on static content, the migration process to IIS 7.0 should be fairly easy. Generally, all that is required is for the content to be moved and any associated site-level or application-level settings to be re-created. However, there are additional options and considerations for other types of applications, such as those that were built using ASP.NET or that rely on IIS 6.0 architectural features. In this section, you'll learn about how to migrate Web applications to IIS 7.0.

Upgrading from Windows Server 2003 and IIS 6.0

One approach to moving Web applications to Windows Server 2008 is to perform an in-place upgrade of a computer running Windows Server 2003. The upgrade process automatically makes decisions that will help preserve compatibility with older applications. For example, the majority of role services that are optional with a standard Web Server (IIS) role installation are included automatically. Furthermore, IIS 6.0 management tools and features are available for use. Following an upgrade to Windows Server 2008 and IIS 7.0, verify which installed components are required and remove those that are not. And, as with any migration, thoroughly test the functionality of your Web sites before redeploying them into production.

Another option for upgrading to IIS 7.0 is to migrate Web sites manually by copying the relevant content to a new Windows Server 2008 installation. In this approach, the existing content is transferred to a new server, and Web sites and Web applications must be reconfigured.

Installing IIS 6 Management Compatibility

Some Web sites and Web applications might include application code that relies on the architecture of IIS 6.0 for handling requests. Examples include Web applications that need access to the IIS 6.0 configuration database and compatibility with earlier scripting methods. In addition, some applications might require access to an earlier version of the management console.

By default, backward-compatibility features are not installed automatically for new Web server installations in Windows Server 2008. To provide backward-compatibility, you can use Server Manager to add role services to the Web Server (IIS) role. The available options are:

- **IIS 6 Management Compatibility** This compatibility feature provides support for two scripting and administration features that were included with IIS 6.0: Admin Base Object (ABO) and Active Directory Services Interface (ADSI). Web applications that relied on these technologies will need these features to operate correctly. Additionally, the IIS 6 Management Compatibility role service is required to enable other IIS 6.0 compatibility options.
- **IIS 6 Metabase Compatibility** IIS 6.0 used a configuration database known as the metabase for storing server settings and other details. In IIS 7.0, this has been replaced by XML-based configuration files such as ApplicationHost.config and Web.config files. IIS 6.0 Web applications could use the ability to query the metabase to manage IIS settings. To support these applications, you must enable the IIS 6 Metabase Compatibility role service.
- **IIS 6 WMI Compatibility** Windows Management Instrumentation (WMI) is a programming interface that enables application code to query and manage IIS settings, using scripts or WMI-capable tools. This role service adds compatibility that enables IIS 6.0 WMI–based commands to apply to IIS 7.0 Web servers.
- **IIS 6 Scripting Tools** Web developers and systems administrators can transition IIS 6.0 management scripts to IIS 7.0 by enabling this role service. The IIS 6 Scripting Tools option adds support for using ActiveX Data Objects (ADO) and ADSI.
- **IIS 6 Management Console** For systems administrators who want to manage IIS 6.0 installations remotely, it is possible to install IIS 6 Management Console on Windows Server 2008. This console is capable only of connecting to IIS 6.0 servers, however, and cannot connect to a Windows Server 2008 Web server.

Overall, these tools and features can help ensure that previous versions of applications that relied on IIS 6.0 will continue to function in Windows Server 2008.

Understanding ASP.NET Integration Modes

IIS 7.0 provides enhancements for the ASP.NET development platform. In previous versions of IIS, ASP.NET processing was performed through an ISAPI code module. Although this approach worked well, there were some important limitations. In IIS 7.0, ASP.NET integration

has been enhanced by more closely incorporating the process of ASP.NET Web pages with the Web server request pipeline. This new architecture offers several benefits, including greater control over request processing and the ability to use ASP.NET features for types of content other than dynamic Web pages.

All ASP.NET applications can take advantage of the new .NET Integrated Mode pipeline when they are running on IIS 7.0. However, applications that relied on IIS 6.0 architecture for intercepting and modifying requests will need support for the Classic pipeline mode. You can configure the processing mode by changing application pool settings or modifying the configuration of existing application pools. (Both topics were covered earlier in this lesson.)

Quick Check

1. How can you avoid potential performance or resource-related problems for multiple Web sites that are running on the same IIS Web server?
2. How can you back up the configuration of the IIS Web server before you make changes to the configuration?

Quick Check Answers

1. By configuring each Web site to run in a separate application pool, you can minimize the risks of problems with one application conflicting with another.
2. The AppCmd.exe utility provides commands for creating and restoring backups of the IIS configuration.

Practice: Configuring and Managing IIS Settings

In these practice exercises, you will create Web sites and Web applications on Server2.contoso.com and test the backup and recovery process for configuration settings. The steps in the exercise assume that you have already installed the Web Server (IIS) role on this computer, using the default role services. (For more information about adding the role, see Lesson 1, Exercise 1.) The steps in Exercise 2 require you to complete the steps in Exercise 1 because the new Web site you created will be used for testing the backup and restore processes.

▶ Exercise 1: Creating Web Sites and Web Applications

In this exercise, you will use IIS Manager to create a new Web site on the local server. Because the default Web site is already configured to use the standard HTTP and HTTPS ports, you will specify alternate site binding information. You will also create a new Web application that includes a test Web page to ensure that the server is responding properly.

1. Log on to Server2.contoso.com with local administrative credentials.

2. Before you create a new Web site, you will create content folders within the file system. Using Windows Explorer, navigate to the *%SystemDrive%*\Inetpub\wwwroot path on the computer's system drive.
3. Within the Wwwroot folder, create a new folder called **Contoso**. Within the Contoso folder, create another new folder called **WebApp01**.

   You will use these folders as the physical paths for the Web site and Web application you will create in later steps.
4. Copy the Iisstart.htm and Welcome.png files from the Wwwroot folder to the Contoso folder. Rename the Iisstart.htm file to Default.htm.
5. Within the *%SystemDrive%*\Inetpub\wwwroot\Contoso\WebApp01 folder, create a new text file named Default.htm. Within the text file, enter the following text and then save the file:

```
<html>
<title>Web Application 01</title>
<body>
<h1>Welcome to Web Application 01.<h1>
</body>
</html>
```

6. Launch Internet Information Services (IIS) Manager from the Administrative Tools program group..
7. If prompted to connect to a server, choose to connect to the local computer.
8. Expand the local computer object and the Sites container to view a list of existing Web sites.

   You will see the default Web site that was installed when the Web Server (IIS) role was added to the computer.
9. To create a new Web site, right-click the Sites container and click Add Web Site.

   This will open the Add Web Site dialog box.
10. For the name of the new Web site, type **Contoso Test Site**. Note that, by default, a new application pool of the same name is created and selected automatically. For this practice exercise, you will use this new application pool; however you can choose an existing pool by clicking the Select button.
11. For Physical Path, browse to the *%SystemDrive%*\Inetpub\wwwroot\Contoso folder that you created earlier. Accept the default security setting of Pass-Through Authentication and then click Test Settings. Note that IIS is able to verify authentication but not authorization because this information will not be known until a user attempts to access the site. Click Close to return to the Add Web Site dialog box.
12. In the Binding section, choose the following settings:
    - Protocol: **HTTP**
    - IP Address: **All Unassigned**

- Port: **8000**
- Host Name: (blank)

13. Verify that the Start Web Site Immediately option is selected and then click OK to create and start the new Web site automatically.
14. Click the newly created Contoso Test Site object in the left pane of IIS Manager. Note that the Actions pane provides commands for working with the Web site. To verify that the site is configured properly, click the *Browse *:8000 (http)* command. This will launch Internet Explorer automatically and connect to *http:://Server2:8000.contoso.com*. You should see the default IIS start page content in the Web browser. When finished, close Internet Explorer.
15. To create a new Web application, right-click the Contoso Test Site item in IIS Manager and select Add Application.
16. For the Alias of the application, type **TestApp**. For the physical path, type or browse to the %SystemDrive%**\Inetpub\wwwroot\Contoso\WebApp01** physical path. Notice that the DefaultAppPool option is selected for the application pool.
17. Click the Select button to change the application pool to Contoso Test Site. Leave all other settings at their defaults and then click OK to create the new Web application.
18. In the left pane of IIS Manager, you will see a new Web application called TestApp within the Contoso Test Site object. To verify the content of this application, select the TestApp item and then click the Content View button at the bottom of the center pane in IIS Manager. You will see the default.htm file that you created earlier.
19. To test the Web application, click the Browse button in the Manage Application section of the Actions pane. This will launch Internet Explorer and connect to *http://Server2:8000.contoso.com/TestApp/default.htm*. The title bar for the Web browser will read Web Application 01, and the text will display the welcome message you specified in the HTML file. When finished, close Internet Explorer.
20. Close IIS Manager.

▶ Exercise 2: Backing Up and Restoring the IIS Configuration

In this exercise, you will walk through the steps required to make a backup of the IIS configuration, using the AppCmd.exe utility. You will then delete the Contoso Web Site object that you created in Exercise 1, using IIS Manager. To restore the Web site configuration, you will again use the AppCmd.exe utility.

1. Log on to Server1.contoso.com with local administrative credentials..
2. Open a new command prompt window by clicking Start and then Run Type **cmd**.
3. Change the current working directory to the location of AppCmd.exe by typing **cd** %SystemRoot%**\Windows\System32\Inetsrv**.

4. To create a new backup of the IIS configuration, type the following command at the command prompt:

   ```
   AppCmd add backup "IISBackup01"
   ```

5. To verify that the backup has been created, type the following command:

   ```
   AppCmd list backups
   ```

6. You should see the IISBackup01 item in the list. (If you have made other backups of the configuration, they will also appear in the list.)
7. Leave the command prompt window open and then launch the Internet Information Services (IIS) Manager console

   In the next step, you will remove a Web site from the configuration of IIS.
8. Connect to the local server and expand the Sites object. Right-click the Contoso Test Site object and select Remove. When prompted, select Yes to confirm the removal. Note that the site and its Web application have been deleted.
9. Return to the command prompt window and type the following command to restore the IIS configuration from the backup you created earlier:

   ```
   AppCmd restore backup "IISBackup01"
   ```

10. When the command finishes, close the command prompt and return to the IIS Manager console.
11. To refresh the display, right-click the Sites object and choose Refresh.

    You will now see the Contoso Test Site object. Note that removing the Web site did not delete any of the content that was stored in the file system, so the site should be available for use. In some cases, it might be necessary to close the IIS Manager console and reload it after the restore process has been performed.
12. When finished, close IIS Manager.

Lesson Summary

- IIS Manager provides an integrated graphical user interface for managing IIS-related settings, features, and Web content.
- Web sites have associated site bindings that specify the protocol, IP address, port, and host headers to which a site will respond.
- Systems administrators can configure bandwidth limitations, user connection limits, and logging settings for each Web site.
- Create Web applications within IIS to manage a set of related Web application code.

- Application pools provide independence and isolation for multiple Web sites and Web applications that are running on the same IIS installation.
- Systems administrators and Web developers can use AppCmd.exe to perform common IIS management tasks from the command line.
- IIS server configuration settings are stored in the ApplicationHost.config file. These settings can be overridden by Web.config files that are located in content folders.
- Windows Server 2008 provides numerous backward-compatibility features for supporting applications built for IIS 6.0 and for managing IIS 6.0 servers.

Lesson Review

You can use the following questions to test your knowledge of the information in Lesson 2, "Configuring Internet Information Services." The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the "Answers" section at the end of the book.

---

1. You are a systems administrator responsible for managing a Windows Server 2008 Web server. Currently, there are no Web sites configured on the server. You need to configure the server to host two different Web applications: EngineeringApp and SalesApp. Both Web applications must be accessible by using HTTP port 80 without the use of host headers. Also, you must protect against problems in one Web application affecting the performance or reliability of the other Web application. Which two steps should you take to meet these requirements?
   - **A.** Create a single Web site that contains both Web applications.
   - **B.** Create two Web sites, one for each Web application.
   - **C.** Assign both Web applications to the same application pool.
   - **D.** Assign each Web application to its own application pool.
2. You are a systems administrator responsible for managing a Windows Server 2008 Web server. You have not created any manual backups of the IIS configuration. Recently, a Web developer reported that he had accidentally removed two Web sites from the IIS configuration. Both Web sites contained several Web applications. You have verified that the two sites do not appear when you open the IIS Manager console and expand the Sites object. You have also verified that the content for the two Web sites is still present in the C:\WebSites folder. You have also ensured that no other changes have been made to the IIS configuration by interviewing other members of the Web development team.

Which of the following steps should you take to restore the two missing Web sites with their associated settings as quickly as possible?

A. Manually re-create the two Web sites and then re-create the associated Web applications.

B. Manually modify the Web server's ApplicationHost.config file and add the Web site and Web application settings.

C. Restore the IIS configuration, using the AppCmd utility.

D. Copy an earlier version of the ApplicationHost.config file from the *%SystemDrive%* \Inetpub\wwwroot\History folder over the current active version of Application-Host.config.

Chapter Review

To further practice and reinforce the skills you learned in this chapter, you can perform the following tasks:

- Review the chapter summary.
- Review the list of key terms introduced in this chapter.
- Complete the case scenarios. These scenarios set up real-world situations involving the topics of this chapter and ask you to create a solution.
- Complete the suggested practices.
- Take a practice test.

Chapter Summary

- The Web Server (IIS) role in Windows Server 2008 is designed to support Web sites and Web applications.
- The Web Server (IIS) role offers numerous role services related to security, performance, diagnostics, and backward compatibility.
- The IIS Manager console is the primary method for creating and managing Web sites, Web applications, application pools, and virtual directories.
- IIS can be managed, using the AppCmd.exe command-line utility, Windows PowerShell, and the.NET Framework.
- Windows Server 2008 provides several methods for maintaining backward compatibility with applications built for previous versions of IIS.

Key Terms

Do you know what these key terms mean? You can check your answers by looking up the terms in the glossary at the end of the book.

- AppCmd.exe
- application pools (IIS)
- ApplicationHost.config file
- ASP.NET
- Hypertext Transfer Protocol (HTTP)
- Hypertext Transfer Protocol Secure (HTTPS)
- IIS Manager
- Internet Information Services (IIS)
- Secure Sockets Layer (SSL)

- site bindings
- Transport Layer Security (TLS)
- Web Server (IIS) server role
- Web server farms
- Web.config files
- Windows PowerShell
- Windows System Resource Manager (WSRM)

Case Scenarios

In these case scenarios, you will apply the information you have learned about Web sites and Web applications to meet business and technical requirements.

Case Scenario 1: IIS Web Server Administration

You are a systems administrator responsible for managing five different Web servers for your organization. Each of the Web servers supports multiple Web applications. Your general requirements include ensuring reliability and performance for all Web applications. In addition, you must simplify administration tasks for the servers. The organization requires that no more than four hours of configuration or site content changes can be lost in the event of a hardware failure. A Web developer has stated that she needs to make multiple changes to the IIS settings on one test Web server.

1. How can you simplify the configuration of all the servers, assuming that the settings must be the same for all of them?
2. Which content should you include in the backup process?
3. What are two ways in which you can roll back the server configuration on a test server if an accidental or unwanted modification is made?

Case Scenario 2: Managing Multiple Web Sites

You are a server administrator responsible for managing fifteen Web sites on a single Windows Server 2008 Web server. For security, reliability, and performance reasons, you need to prevent problems in one Web application from causing issues with others. In addition, several different Web applications must be configured to respond on HTTP port 80 and HTTP port 443, using the same public IP address. One of the ASP.NET Web applications was originally designed for IIS 6.0 and takes advantage of advanced request processing features.

1. How can you minimize the risks associated with Web application defects affecting other Web applications on the same server?

2. What configuration settings will enable you to meet the default HTTP and HTTPS connection requirements?
3. What are some methods by which you can support the IIS 6.0 Web application on Windows Server 2008?

Suggested Practices

To help you successfully master the exam objectives presented in this chapter, complete the following tasks.

Managing Web Applications

Perform the following exercises to practice the process of creating and managing Web applications, using IIS Manager and command-line utilities.

- **Practice 1: Deploying and Configuring Web Applications** Web applications often have numerous requirements and features that must be enabled to function properly in IIS. If possible, download sample Web applications and deploy them in IIS, using various settings for application pools and other options. A good starting point for downloading applications based on ASP.NET is the Microsoft ASP.NET Starter Kit site at *http://www.asp.net/downloads/starter-kits/*. In addition, if your organization has any existing Web sites or applications, attempt to install them in a test environment.
- **Practice 2: Managing IIS from the Command Line** Once you are familiar with the concepts of using IIS Manager to create and manage Web sites, try performing the same actions by using the command line. Use the AppCmd.exe utility to perform operations such as:
  - Creating a new Web site, including unique site binding parameters.
  - Creating multiple Web applications within the new Web site.
  - Adding virtual directories that point to file system locations outside of the folder for the default site or Web application.
  - Backing up and restoring the IIS configuration.
  - Deleting the test sites and other objects you have created.

If you need to create many sites on several Web servers, you can also combine multiple commands in a batch file to automate the process.

Take a Practice Test

The practice tests on this book's companion CD offer many options. For example, you can test yourself on just one exam objective, or you can test yourself on all the 70-643 certification exam content. You can set up the test so that it closely simulates the experience of taking a certification exam, or you can set it up in study mode so that you can look at the correct answers and explanations after you answer each question.

MORE INFO Practice tests

For details about all the practice test options available, see the "How to Use the Practice Tests" section in this book's introduction.

Chapter 7

Installing and Configuring Terminal Services

Terminal Services is a technology that enables remote users to establish and interact with a desktop session on a computer running Microsoft Windows Server 2008. Because of the many features, tools, and functions associated with Terminal Services, there's a fair amount to learn about this topic for both real-world administration and the 70-643 exam.

This chapter will focus first on deploying the core Terminal Services component and then on configuring all options available in the Terminal Services Configuration console, the main Terminal Services configuration tool. In the next chapter, the discussion will move on to configuring clients and additional server components.

Exam objectives in this chapter:

- Configuring Terminal Services
- Configuring Terminal Services server options
- Configuring Terminal Services licensing
- Configuring Terminal Services load balancing

Lessons in this chapter:

Before You Begin

To complete the lessons in this chapter, you must have

- A computer running Windows Server 2008 named Server1 that is a domain controller in a domain named Contoso.com.
- A computer running Windows Server 2008 named Server2 that is a member server in the Contoso.com domain.
- A Server Core installation of Windows Server 2008 named Core1 that is a member server in the Contoso.com domain.

Real World

JC Mackin

The most important thing to know about Terminal Services in Windows Server 2008 is that it includes some radically new and important features beyond those offered in Remote Desktop or in any previous version of Windows Server. The RemoteApp feature, for starters, enables you to run a remote program on another computer as if it were installed locally. Another feature, TS Web Access, provides a Web page from which you can launch these same remote applications, and TS Gateway gives your organization an attractive alternative to virtual private networks (VPNs) by allowing authorized users to connect from the Internet to any desired desktop on your internal network.

In the past, such functionality was made available through third-party applications only. Now that these powerful features are built into Windows Server 2008, more organizations will start to deploy them and take advantage of them. As a Windows support technician, you might have dismissed Terminal Services in the past as a feature that you didn't really have to understand too well, but the role of Terminal Services is now certain to grow.

Terminal Services is moving closer to the core of essential, real-world support skills that you absolutely must know and understand. Given this, it's time to start looking very closely at this feature if you haven't already.

Lesson 1: Deploying a Terminal Server

This lesson provides an introduction to Terminal Services by describing the basic features of this server role, the installation method of its main component, and the preparatory steps necessary to perform in advance of server deployment.

After this lesson, you will be able to:

- Understand the basic features and function of Terminal Services.
- Compare and contrast Terminal Services with the built-in Remote Desktop feature of Windows.
- Install the Terminal Services role on a full installation and a server core installation of Windows Server 2008.
- Describe client licensing options for a terminal server.
- Prepare a terminal server for deployment.

Estimated lesson time: 40 minutes

Terminal Services Basics

Terminal Services enables many users to establish simultaneous interactive desktops or application sessions on a remote computer running Windows Server 2008. During a Terminal Services session, Terminal Services clients offload virtually the entire processing load for that session to the terminal server. The functionality offered by Terminal Services thus enables an organization to distribute the resources of a central server among many users or clients. For example, Terminal Services is often used to make a single installation of an application available to many users throughout an organization

Comparing Terminal Services and Remote Desktop

Microsoft Windows XP, Windows Vista, Windows Server 2003, and Windows Server 2008 all include a feature called Remote Desktop, which, like Terminal Services, enables users to establish an interactive desktop session on a remote computer. Remote Desktop and Terminal Services are in fact closely related. First, both technologies use the same client software, named Remote Desktop Connection (also called *Terminal Services Client*). This client software is built into all versions of Windows since Windows XP but can be installed on virtually any Windows-based or non-Windows–based computer. From the remote user's perspective, then, the procedure of connecting to a terminal server is identical to connecting to a remote desktop. Second, the server component of both features is also essentially the same. Both Terminal Services and Remote Desktop rely on the same service, called the Terminal Services service. Finally, both Remote Desktop and Terminal Services establish sessions by means of the same protocol, called Remote Desktop Protocol (RDP), and through the same TCP port, 3389. Despite these similarities, the differences between Remote Desktop and Terminal Services are significant in

that Terminal Services offers much greater scalability and a number of important additional features. For example, only two remote users simultaneously can be connected to a computer running Windows Server 2008 on which Remote Desktop is enabled, but no such limitation exists for a server on which Terminal Services has been configured. Terminal Services in Windows Server 2008 also includes the following additional features beyond those available in Remote Desktop:

- **RemoteApp** In Windows Server 2008, the RemoteApp component of Terminal Services enables you to deploy an application remotely to users as if the application were running on the end user's local computer. Instead of providing the entire desktop of the remote terminal server within a resizable window, RemoteApp enables a remote application to be integrated with the user's own desktop. The application deployed through Terminal Services thus runs in its own resizable window with its own entry in the taskbar.
- **TS Web Access** TS Web Access enables you to make applications hosted on a remote terminal server available to users through a Web browser. When TS Web Access is configured, users visit a Web site (either from the Internet or from the organization's intranet) and view a list of all the applications available through RemoteApp. To start one of the listed applications, users simply click the program icon on the Web page.
- **Network Load Balancing** By using Network Load Balancing (NLB), you can deploy a number of terminal servers in a farm that, from the perspective of remote users, emulates a single server. To enhance the functionality of NLB, you can also use the TS Session Broker role service. The TS Session Broker component ensures that clients connecting to the load-balanced server farm can reconnect to disconnected sessions.
- **TS Gateway** TS Gateway enables authorized users on the Internet to connect to internal remote desktops and terminal servers located on a corporate network. TS Gateway provides security for these connections by tunneling each RDP session inside an encrypted Hypertext Transfer Protocol Secure (HTTPS) session. By providing authorized users broad access to internal computers over an encrypted connection, TS Gateway can eliminate the need for a VPN in many cases.

Advantages of Remote Desktop

The main advantage of Remote Desktop, compared to Terminal Services, is that its functionality is built into Windows Server 2008 fully and does not require the purchase of any Terminal Services client access licenses (TS CALs). If you don't purchase any TS CALs for Terminal Services, the feature will stop working after 120 days. After this period, you will be able to use only the built-in Remote Desktop feature of Windows Server 2008.

Another advantage of Remote Desktop, compared to Terminal Services, is that the feature is very easy to implement. Whereas enabling Terminal Services requires installing and configuring a new server role, enabling Remote Desktop requires you to select only a single option in the System Properties dialog box.

NOTE Remote Desktop vs. Remote Desktop for Administration

In Windows Server 2003 and Windows Server 2008, the built-in Remote Desktop feature is often referred to as Remote Desktop for Administration (RDA). The difference between RDA and the Remote Desktop feature in Windows XP and Windows Vista is that RDA allows three simultaneous logon sessions to the RDA-enabled server: two remote sessions and one console session. Windows XP and Windows Vista, however, do not allow concurrent sessions for Remote Desktop. Only one Remote Desktop user can connect at a time and, when a remote user does connect, any locally logged-in user must first be logged off.

Exam Tip In Windows Server 2008, the Remote Desktop feature typically is used for remote administration, and Terminal Services is used to host applications. However, the main difference between these two features is scale, and the purposes of their implementations do overlap. You can use the Remote Desktop feature to host a seldom-used application just as you can administer a server remotely on which Terminal Services has been installed. Remember also that the core client and server components of these technologies are shared, so do not be surprised if at times you hear the terms used interchangeably.

Enabling Remote Desktop

By default, Windows Server 2008 does not accept connections from any Remote Desktop clients. To enable the Remote Desktop feature in Windows Server 2008, use the Remote tab of the System Properties dialog box. To access this tab, you can open System located in Control Panel and then click the Remote Settings link, or you can type **control sysdm.cpl** in the Start menu Search box and then, after System Properties dialog box opens, click the Remote tab.

On the Remote tab, if you want to require a high standard of security from RDP connections, select the option to require Network Level Authentication (NLA), as shown in Figure 7-1. This selection will enable connections only from Remote Desktop Connection clients running on Windows Vista or later. Alternatively, you can select the option to allow connections from computers running any version of Remote Desktop.

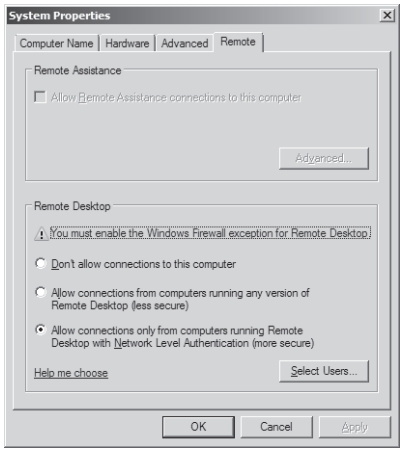

Figure 7-1 Enabling the Remote Desktop feature on Windows Server 2008

In Windows Server 2008, when you use the System Properties dialog box to allow Remote Desktop connections, a Windows Firewall exception for RDP traffic is created automatically. Therefore, you do not have to create the exception manually to allow connections from Remote Desktop clients.

NOTE What is Network Level Authentication?

NLA is a feature of Remote Desktop Protocol 6.0 that ensures that user authentication occurs before a Remote Desktop connection is fully established between two computers. With earlier versions of RDP, a user could enter a username and password for authentication only after a Log On To Windows dialog from the remote computer appeared in the Remote Desktop session. Because every attempt to authenticate a session demanded relatively significant resources from the server, this behavior in earlier versions of RDP made Remote Desktop–enabled and Terminal Services–enabled computers susceptible to denial-of-service attacks.

Also important to know is that, by default, Remote Desktop Connection 6.0 (also known as Terminal Services Client 6.0) does not support NLA on computers running Windows XP. However, this version of the Remote Desktop client can be made to support NLA on Windows XP SP2 if you download and install the Terminal Services Client 6.0 update for Windows XP (KB925876), available on the Microsoft Web site.

Enabling Remote Desktop on a Server Core Installation

A Server Core installation of Windows Server 2008 does not support the full Terminal Services role. However, you can enable the Remote Desktop feature on a Server Core installation by using the Server Core Registry Editor script, Scregedit.wsf. Scregedit.wsf provides a simplified way of configuring the most commonly used features in a Server Core installation of Windows Server 2008.

IMPORTANT Where can you find Scregedit.wsf?

Scregedit.wsf is located in the *%SystemRoot%*\System32 folder of every Server Core installation.

To use the Scregedit.wsf script to enable Remote Desktop, use *Cscript.exe* to invoke the script and then pass the /AR switch a value of 0, which allows Remote Desktop connections, Passing the /AR switch a value of 1 disables Remote Desktop connections. An example is shown here:

```
Cscript.exe C:\Windows\System32\Scregedit.wsf /AR 0
```

By default, enabling Remote Desktop on the Server Core installation in this way configures the server to accept Remote Desktop connections only from clients running Windows Vista or later. To enable the server to accept Remote Desktop connections from earlier versions of RDP, you need to relax the security requirements of the server by using the Scregedit.wsf script with the /CS switch and a value of 0, as shown:

```
Cscript.exe C:\Windows\System32\Scregedit.wsf /CS 0
```

Creating a Firewall Exception in Server Core

Unlike the full installation of Windows Server 2008, a Server Core installation does not create a Windows Firewall exception automatically for you when you enable Remote Desktop. Therefore, you need to create an exception for TCP port 3389 before any remote clients will be able to establish an RDP session with the server. To open this firewall port, type the following command at the prompt on the Server Core server:

```
netsh firewall set service type=remotedesktop mode=enable
```

NOTE Connecting to a Server Core through Remote Desktop

When you connect to a Server Core installation by means of Remote Desktop, you receive the same interface that you would receive as if you were seated locally at the server. A Remote Desktop connection to a Server Core–enabled computer, in other words, does not provide you with access to any additional graphical tools to manage the server.

Exam Tip For the 70-643 exam, you need to know how to enable Remote Desktop on a Server Core installation of Windows Server 2008 and how to allow connections from RDP clients earlier than RDP 6.0. Also, do not be surprised if the exam refers to this process as "enabling Terminal Services or enabling Terminal Services for remote administration."

Installing Terminal Services

Unlike Remote Desktop, the full implementation of Terminal Services requires you to add the Terminal Services server role. As with any server role, the simplest way to install the feature on a full installation of Windows Server 2008 is to click Add Roles in Server Manager.

Clicking Add Roles launches the Add Roles Wizard. On the Select Server Roles page, select the Terminal Services check box, as shown in Figure 7-2.

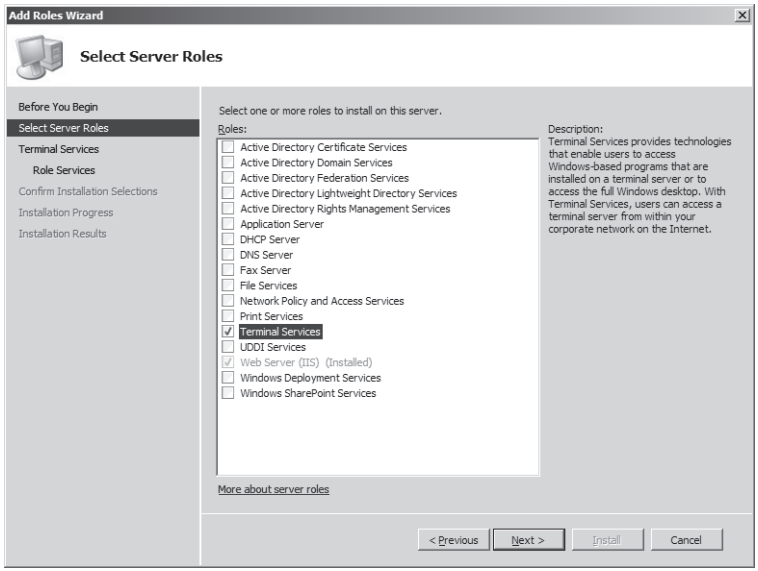

Figure 7-2 Adding the Terminal Services role

Selecting Role Services

On the Select Role Services page of the Add Roles Wizard, you can select any of the following five role services associated with the Terminal Services role.

- **Terminal Server** This is the only required role service for the Terminal Services role. This role service provides the basic functionality of Terminal Services, including the RemoteApp feature.
- **TS Licensing** You need to install this role service only if you have purchased TS CALs and can activate a license server. If you have not purchased any TS CALs, Terminal Services has a 120 day grace period.
- **TS Session Broker** Install and configure this role service when you plan to implement Terminal Services in an NLB server farm. This feature enhances the functionality of the server farm by ensuring that clients are able to reconnect to disconnected sessions.

- **TS Gateway** Install this role service if you want to make a number of internal terminal servers accessible to authorized external clients beyond a firewall or Network Address Translation (NAT) device.
- **TS Web Access** Install this role service if you want to make applications deployed through Terminal Services available to clients through a Web page.

The Select Role Services page is shown in Figure 7-3.

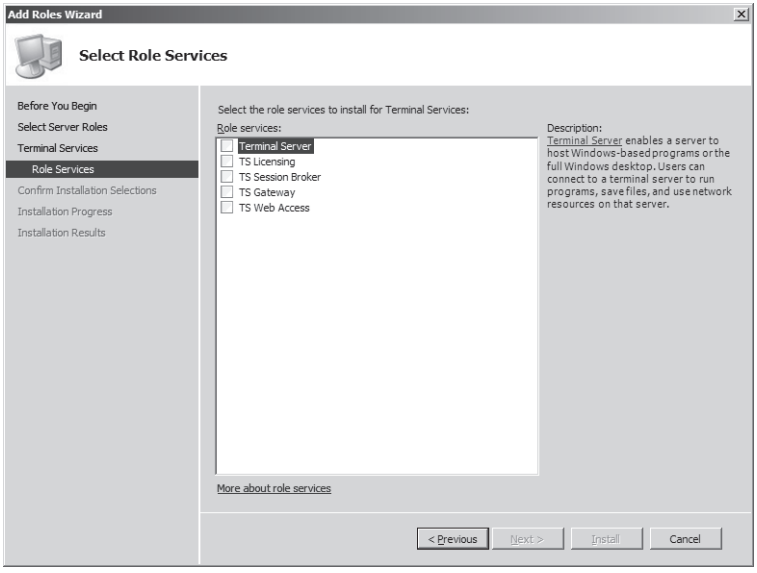

Figure 7-3 Adding the Terminal Services role services

In the description that follows of the Add Roles Wizard pages, it is assumed that you are installing only the Terminal Services role service from the Select Role Services page.

Uninstalling Applications

After you select the Terminal Services role service, the Add Roles Wizard reminds you that any applications that you want to deploy to users through Terminal Services should be installed after you add the Terminal Services role. If you have already installed any applications you want to deploy, you should uninstall and reinstall them. This reminder is shown in Figure 7-4.

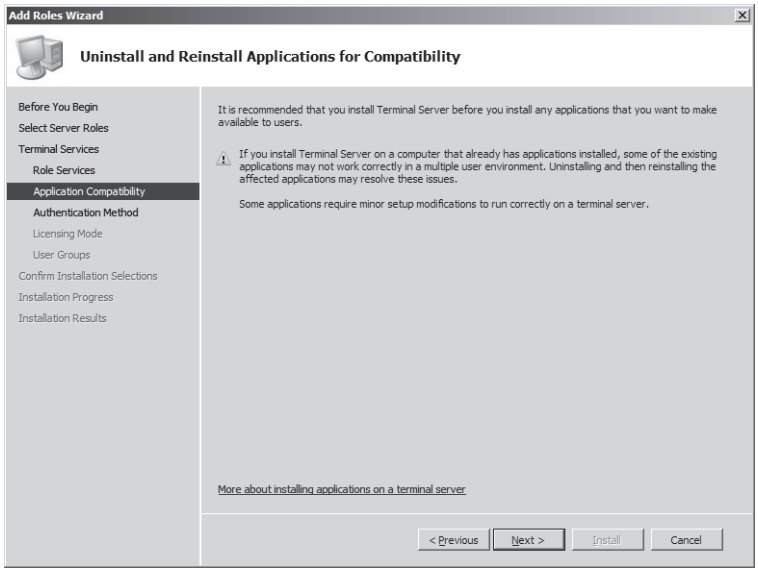

Figure 7-4 Reminder to reinstall TS applications

Specifying NLA Settings

Next, you have to specify whether the terminal server will accept connections only from clients that can perform NLA. When you select this requirement, Remote Desktop connections will be blocked from computers with operating systems earlier than Windows Vista. See Figure 7-5.

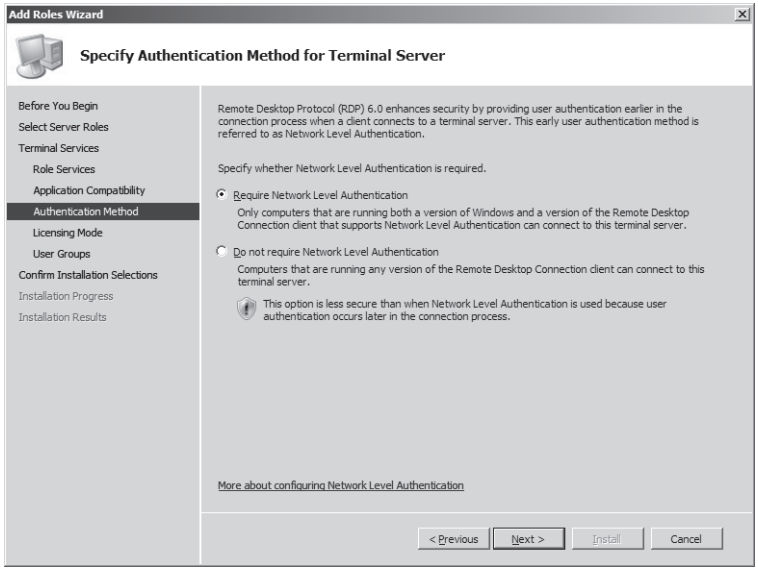

Figure 7-5 Setting NLA/client version requirements

Specifying Client Access License Types

The Add Roles Wizard then gives you the option to specify the TS CAL types you have purchased. Two types of CALs for Terminal Services are available:

- TS Per Device CALs

  TS Per Device CALs are permanent CALs assigned to any computer or device that connects to Terminal Services more than once. When the Per Device licensing mode is used and a client computer or device connects to a terminal server for the first time, the client computer or device is issued a temporary license by default. When a client computer or device connects to a terminal server for the second time, if the license server is activated and if enough TS Per Device CALs are available, the license server issues the client computer or device a permanent TS Per Device CAL.

- TS Per User CALs

  TS Per User CALs give users the right to access Terminal Services from any number of devices. TS Per User CALs are not assigned to specific users. If you opt for per user licensing, you simply need to make sure that you have purchased enough licenses for all the users in your organization.

In deciding which of these two CALs to purchase for your organization, consider several factors. The first factor to consider is the number of devices and users in your organization. In general, it's financially preferable to choose per-device CALs if you anticipate having fewer devices than users over the life of the terminal server and to choose per-user licensing if you anticipate fewer users than devices. Another factor to consider is how often your users travel and connect from different computers. Per-user licensing is often preferable when a small number of users tend to connect from many different sites, such as from customer networks.

If you have not yet decided which TS CALs to purchase, you can select the option Configure Later, as shown in Figure 7-6. You have 120 days to purchase TS CALs and to install these licenses on a locally activated license server. After this grace period, Terminal Services stops functioning.

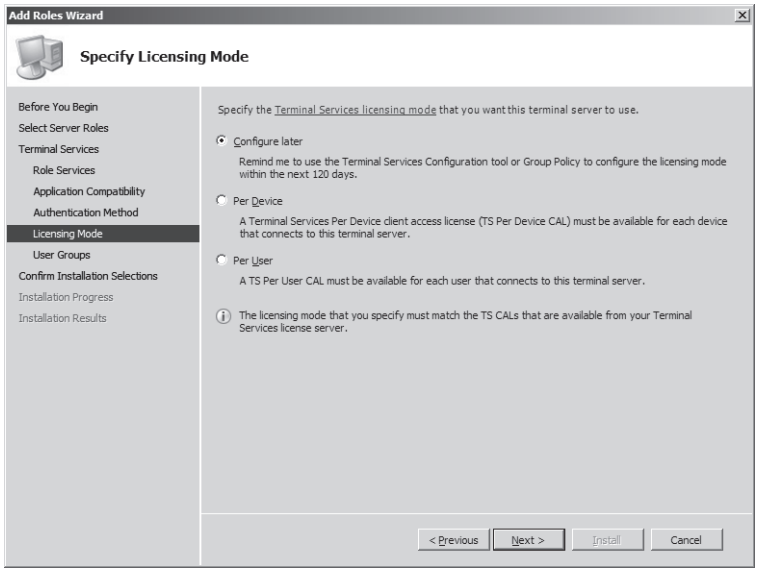

Figure 7-6 Specifying a licensing mode

Exam Tip For the 70-643 exam, you definitely need to know the difference between the client access license modes.

Authorizing Users

The last configuration step is to choose the users and groups you want to allow access through Terminal Services. The Remote Desktop Users built-in local group automatically is granted the user right to connect to the local machine through Terminal Services, and the Add Roles Wizard here simply provides a fast way of adding accounts to this Remote Desktop Users group. By default, local administrators are already members of the Remote Desktop Users group, as shown in Figure 7-7.

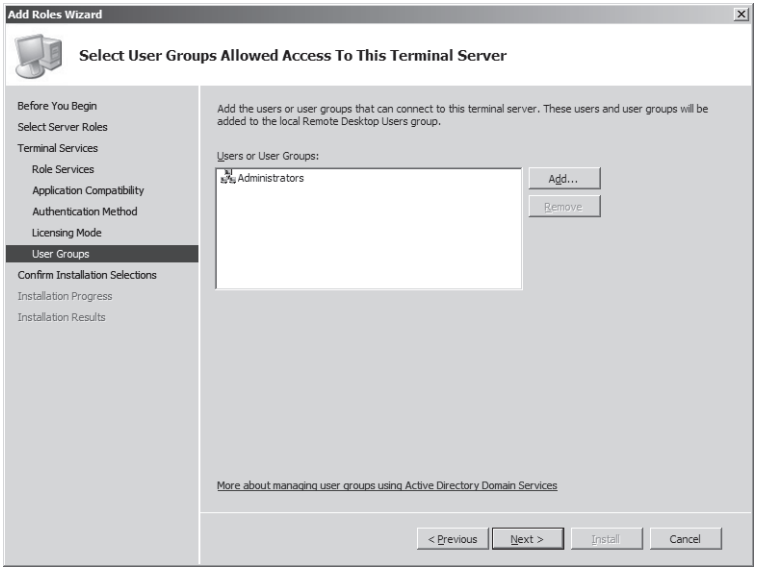

Figure 7-7 Authorizing users for Terminal Services

After this last step, you simply need to confirm your selections and begin the Terminal Services installation, as shown in Figure 7-8.

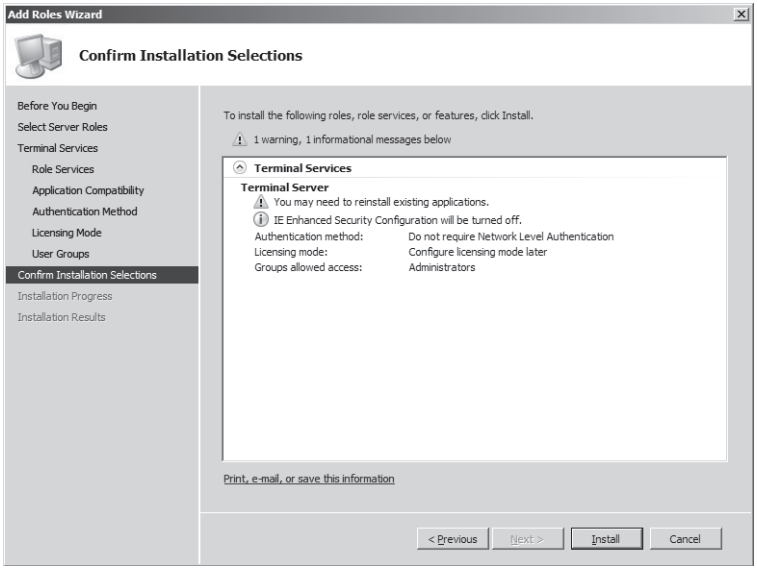

Figure 7-8 Confirming Terminal Services installation options

Staging the Terminal Server

Staging a server refers to the process of preparing it in advance of deployment. In the case of a terminal server, staging a computer includes installing and configuring all of the components on the server that you want to make available to Terminal Services clients. At a minimum, this process includes installing appropriate server features and applications.

Installing Windows Server 2008 Built-in Features

Besides enabling you to add server roles, Server Manager enables you to install any of 36 Windows Server 2008 features. Features are smaller Windows components that enable specific functionality in the operating system. To prepare a terminal server for deployment, you need to know which of these Windows Server 2008 features you want to make available to clients connecting to the terminal server.

Because the only features made available to remote users are those that you install on the terminal server, you need to review client needs and the functionality offered by each feature. For example, if you want Windows Media® Player to be made available to clients connecting to Terminal Services, you have to install the Desktop Experience feature on the computer running Terminal Services.

To install a feature, click Add Features in Server Manager to launch the Add Features Wizard. Figure 7-9 shows a partial list of the features made available by the Add Features Wizard.

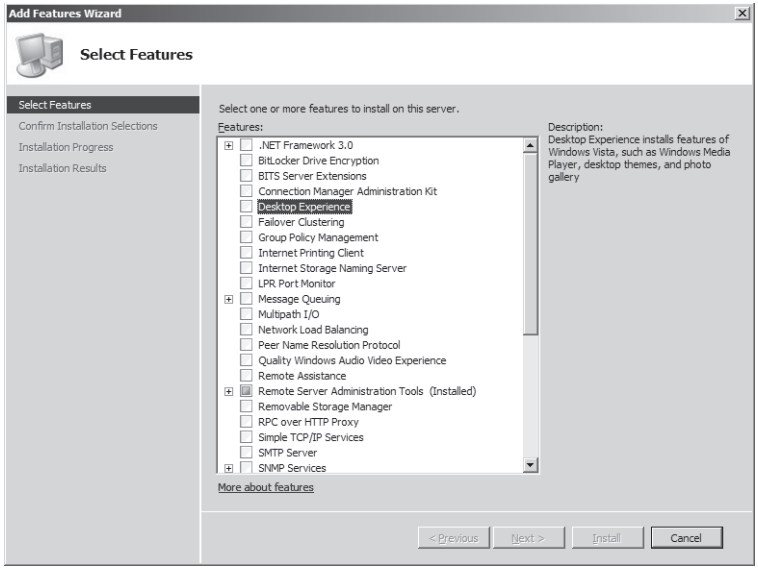

Figure 7-9 The Add Features Wizard

A list follows of some example Windows Server 2008 features that you might need to make available to Terminal Services clients. Successful deployment of Terminal Services requires you to understand these features and to review them during the server staging process.

- **Desktop Experience** This feature installs Windows Media Player 11, desktop themes, and the photo gallery.
- **Quality Windows Audio Video Experience** This feature enables high-quality performance for streaming media over IP networks.
- **Network Load Balancing** The NLB feature enables you to join a server to an NLB cluster or NLB server farm.
- **Windows Server Backup Features** You can install the Windows Server Backup Features to allow administrators to perform backups as part of remote maintenance of the computer running the terminal server.
- **Windows PowerShell** Windows PowerShell is a command-line environment and administrative scripting language built into Windows Server 2008. You can install the PowerShell feature to enable remote administration of the computer running Terminal Services, by using PowerShell.
- **Group Policy Management** Group Policy Management is a console that facilitates administration of Group Policy. You can install this feature if you anticipate that administrators will use the server to manage Group Policy remotely.
- **Windows System Resource Manager** Windows System Resource Manager (WSRM) enables you to manage the resources of a server so that the workload is spread equitably among roles. For more information about WSRM, see the section titled "Allocating Server Resources" later in this lesson.

Exam Tip Be sure to know these server features for the 70-643 exam.

Installing Terminal Services Applications

Terminal Services is often used to deploy a single installation of an application to many users. Deploying an application in this way is frequently the best option for data-entry programs designed to run on a single server or for those tied to a locally installed database. However, you might also want to deploy an application through Terminal Services to reduce associated licensing fees, to offload processing from the client's computer, or simply to facilitate user productivity within a Terminal Services session.

Before deciding to install any applications on the terminal server, be sure to verify that the applications in question can in fact support simultaneous users.

Quick Check

1. Which server feature should you install on a terminal server if you want users to be able to play audio and video in Terminal Services sessions?
2. How many simultaneous remote user sessions, not including the console session, can be hosted on a computer running Windows Server 2008 configured with the Remote Desktop for Administration feature?

Quick Check Answers

1. Desktop Experience
2. Two

Practice: Installing a Terminal Server

In this practice, you will install Terminal Services on a full installation of Windows Server 2008 and then enable the Remote Desktop feature on a server core installation.

▶ Exercise 1: Install the Terminal Services Role

In this exercise, you will install the Terminal Services server role on Server2.

1. Log on to Contoso.com from Server2 as a domain administrator.
2. In Server Manager, select the Roles node in the console tree and then click Add Roles in the details pane.
3. If the Before You Begin page is displayed, click Next.
4. On the Select Server Roles page of the Add Roles Wizard, select the Terminal Services check box and then click Next.
5. On the Terminal Services page, read all the text on the page and then click Next.
6. On the Select Role Services page, select the Terminal Server check box and then click Next.
7. On the Uninstall And Reinstall Applications For Compatibility page, read all the text on the page and then click Next.
8. On the Specify Authentication Method For Terminal Services page, read all the text on the page, select Require Network Level Authentication, and then click Next.
9. On the Specify Licensing Mode page, read all of the text on the page, leave the default selection of Configure Later, and then click Next.
10. On the Select User Groups Allowed Access To This Terminal Server page, read all the text on the page and then click Next.
11. On the Confirm Installation Selections page, read all the text on the page and then click Install.

12. After the installation is complete, read all the text on the Installation Results page and then click Close.
13. In the Add Roles Wizard dialog box, click Yes to restart the server.
14. After the server reboots, log back on to Contoso.com from Server2 as a domain administrator.

    After several moments, the Resume Configuration Wizard appears.
15. When the Installation Results page appears, click Close.

▶ Exercise 2: Test the Terminal Services Connection

In this exercise, you will test the Terminal Services configuration on Server2 by connecting to it from a Remote Desktop Connection on Server1.

1. Log on to Contoso.com from Server1 as a domain administrator.
2. In the Search Search box on the Start Menu, type **mstsc** and then press Enter.

   The Remote Desktop Connection window opens.
3. In the Computer text box of the Remote Desktop Connection windows, type **server2.contoso.com** and then press Enter.

   The Windows Security window opens.
4. In the Windows Security window, enter the credentials of a domain administrator. Be sure to enter the username in the form **contoso** .

   After several moments, a Remote Desktop connection is established to Server2. Within the desktop of Server1, the remote Server2 desktop is designated with a yellowish banner labeled "server2.contoso.com."
5. Using the Start button within the Remote Desktop session to Server2, log off the Remote Desktop connection.

   The Remote Desktop window closes.

▶ Exercise 3: Enable Remote Desktop on a Server Core Installation of Windows Server 2008

In this exercise, you will enable Remote Desktop on the Core1 computer and then test the configuration.

NOTE Server1 and Server2

Although Server1 is needed for this exercise, Server2 is not. If you are using virtual machines and do not have ample RAM to support all three computers, you can shut down Server2 before beginning this exercise.

---

6. Log on to Contoso.com from Core1 as a domain administrator.
7. At the command prompt, type the following command: **cd C:\Windows\System32**
8. At the command prompt, type the following command: **cscript scregedit.wsf /AR /v**

This command shows the current status of the fDenyTSConnections registry setting. When set to 1, the local computer is configured to deny incoming Remote Desktop connections.

9. Type the following command: **cscript scregedit.wsf /AR 0**
10. To verify the setting change, type the following command: **cscript scregedit.wsf /AR /v**

    The output from the command reveals that the fDenyTSConnections registry setting is now set to 0.
11. To ensure that the server will accept connections from RDP clients earlier than 6.0, or from clients native to Windows XP and earlier, type the following command: **cscript scregedit.wsf /CS 0**
12. To verify the setting, type the following command: **cscript scregedit.wsf /CS /v**
13. The output from the command reveals that the RDP-Tcp UserAuthentication setting is now set to 0. This setting enables connections from earlier versions of Remote Desktop.
14. Type the following command: **netsh firewall set service type=remotedesktop mode=enable**

    This command creates a firewall exception for Remote Desktop on Core1.
15. Log on to Contoso.com from Server1 as a domain administrator.
16. Click the Start button, type **mstsc**, and then press Enter.

    The Remote Desktop Connection window opens.
17. In the Computer text box, type Core1 and then click Connect.
18. In the Windows Security window, enter the username and password of a domain administrator. Be sure to enter the name in this format: contoso*username*.
19. In the Windows Security window, click OK.

    After a few moments, a Remote Desktop connection to Core1 is established. The Remote Desktop connection shows the same Server Core desktop on that you can see when you log on to Core1 locally.
20. On Server1, within the Remote Desktop session to Core1, type **logoff** at the command prompt.

    On Server1, the Remote Desktop session closes.
21. On Core1, type **shutdown /p** at the command prompt to shut down the computer.

Lesson Summary

- Terminal Services enables users to establish and interact with a desktop session on a remote computer running Terminal Services.
- Terminal Services shares its core functionality with that of Remote Desktop. The biggest difference between these two features is that Remote Desktop allows only two simultaneous connections (on Windows Server 2008), whereas Terminal Services has no such limits.

- In Windows Server 2008, Terminal Services includes many new and important features such as TS Gateway, RemoteApp, and TS Web Access.
- To install Terminal Services on a computer running Windows Server 2008, add the Terminal Services server role.
- Terminal Services requires client access licenses (CALs) either for all connecting users or for all connecting devices. If you do not purchase and install Terminal Services CALs, the feature will stop working after 120 days.

Lesson Review

The following questions are intended to reinforce key information presented in this lesson. The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the "Answers" section at the end of the book.

---

1. Which commands must you use to enable Remote Desktop on a Server Core installation of Windows Server 2008 and to enable the server to accept connections from clients configured with RDP versions prior to 6.0? (Choose two.)
   - **A.** cscript scregedit.wsf /AR 0
   - **B.** cscript scregedit.wsf /AR 1
   - **C.** cscript scregedit.wsf /CS 0
   - **D.** cscript scregedit.wsf /CS 1
2. You are one of 75 consultants employed by an IT services company named Contoso.com. As part of your job, you and other team members provide network support for over 150 businesses in your city. Your company is about to implement a business process in which consultants must connect to an application server on the Contoso.com network while working at customer premises. When connected to the application server, consultants provide critical information about each assignment in the field. To connect to the Contoso.com application server, consultants are expected to use Remote Desktop Connection on customer computers running Windows XP or Windows Vista. You have been asked to determine whether your company needs to purchase client access licenses (CALs) for Terminal Services. Which of the following options best suits the needs of your organization?

A. Use Remote Desktop for Administration on the application server and purchase per-user CALs.

B. Use Remote Desktop for Administration on the application server but do not purchase any CALs.

C. Install Terminal Services on the application server and purchase per-device CALs.

D. Install Terminal Services on the application server and purchase per-user CALs.

Lesson 2: Configuring Terminal Services

The Terminal Services Configuration console is the main tool used to configure the Terminal Services role. The server options available in this tool primarily affect the user's environment when connecting to the main terminal server component (role service). Other options available in this tool, however, relate to server licensing and load balancing features. After describing all the options and features configurable in the Terminal Services Configuration console, this lesson then describes supplementary configuration options available in Group Policy for one feature in particular: printer redirection.

After this lesson, you will be able to:

- Configure terminal server options.
- Configure Terminal Services load balancing.
- Install and configure a Terminal Services license server.

Estimated lesson time: 50 minutes

Introducing the Terminal Services Configuration Console

The Terminal Services Configuration (TSC) console is designed to control settings that affect all users connecting to the terminal server or all users connecting through certain connection types. For instance, you can use the TSC console to set the encryption level of all Terminal Services sessions, to configure the graphical resolution of sessions, or to restrict all users to one session.

The TSC console is shown in Figure 7-10.

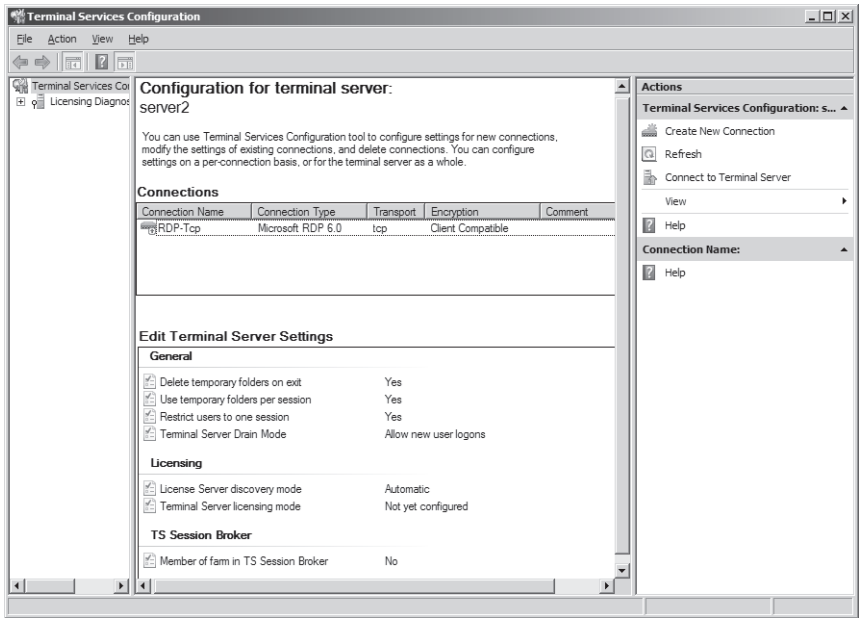

Figure 7-10 The Terminal Services Configuration console

The TSC console provides two general areas for configuration: the connection (RDP-Tcp) properties dialog box and the Edit Terminal Server Settings area. The following sections describe the options available through each of these configuration areas.

Configuring Connection (RDP-Tcp) Properties

Connection properties are used to customize the behavior of all Terminal Services sessions initiated through certain specific transport protocols (such as RDP over TCP) or through specific network adapters on the terminal server. By default, only one connection (named RDP-Tcp) is available for configuration; the properties configured for this connection apply to RDP sessions through all local network adapters. Beyond this default connection, you can also create new connections that apply to a third-party transport protocol (such as one enabled by Citrix MetaFrame) or that apply to any one particular adapter.

For environments using only the built-in functionality offered by Windows Server 2008, the RDP-Tcp connection normally will serve as the only connection, and the RDP-Tcp Properties dialog box provides key configuration options for the entire server.

To open the properties of the RDP-Tcp connection, in the TSC console Connections area, right-click RDP-Tcp and then click Properties. This procedure opens the RDP-Tcp Properties dialog box, as shown in Figure 7-11.

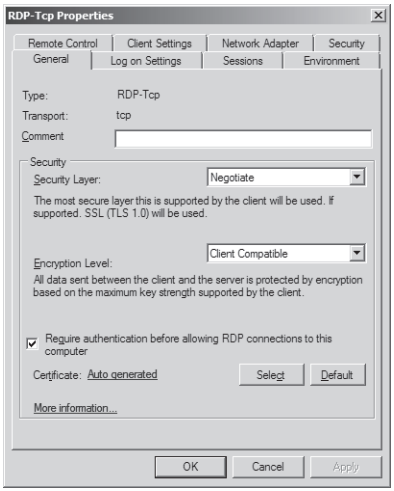

Figure 7-11 RDP-Tcp Properties General tab

The following section explains the configurable options available through each of the eight tabs.

Exam Tip Learn all the settings on the eight RDP-Tcp properties tabs. This is a key area that is tested on the 70-643 exam.

General Tab

The General tab enables you to control security level, encryption level, and NLA settings.

Security All RDP connections are automatically encrypted. Security settings determine the type of encryption used for these Terminal Services connections. Three options for this setting are available: RDP Security Layer, SSL (TLS 1.0), and Negotiate.

- The RDP Security Layer option limits encryption to the native encryption built into Remote Desktop Protocol. The advantages of this option are that it requires no additional configuration and that it offers a high standard of performance. The disadvantage of this option is that it does not provide terminal server authentication for all client types. Although RDP 6.0 can provide server authentication for clients running on Windows Vista and later, Terminal Services clients running Windows XP and earlier do not support server authentication. If you want to allow RDP clients running on Windows XP to authenticate the terminal server before establishing a connection, you have to configure SSL encryption.
- The SSL (TSL 1.0) option offers two advantages over RDP encryption. First, it offers stronger encryption. Second, it offers the possibility of server authentication for RDP client

versions 5.2 and later. SSL is, therefore, a good option if you need to support terminal server authentication for Windows XP clients. However, this option does have some drawbacks. To begin with, SSL requires a computer certificate for both encryption and authentication. By default, only a self-signed certificate is used, which is equivalent to no authentication. To improve security, you must obtain a valid computer certificate from a trusted certification authority (CA), and you must store this certificate in the computer account certificate store on the terminal server. Another disadvantage of SSL is that its high encryption results in slower performance compared to that of other RDP connections.

MORE INFO More Info

More information on server certificates is covered in Chapter 3, "Managing Web Server Security."

- When you choose the Negotiate option, the terminal server will use SSL security only when supported by both the client and the server. Otherwise, native RDP encryption is used. Negotiate is also the default selection.

Encryption Level The Encryption Level setting on the General tab enables you to define the strength of the encryption algorithm used in RDP connections. The default selection is Client Compatible, which chooses the maximum key strength supported by the client computer. The other available options are FIPS Compliant (highest), High, and Low.

Network Level Authentication When the Allow Connections Only From Computers Running Remote Desktop With Network Level Authentication setting is enabled, only clients that support NLA will be allowed to connect to the terminal server.

To determine whether a computer is running a version of the Remote Desktop Connection (RDC) client that supports NLA, start the RDC client, click the icon in the upper-left corner of the Remote Desktop Connection dialog box, and then click About. Look for the phrase Network Level Authentication Supported in the About Remote Desktop Connection dialog box, shown in Figure 7-12.

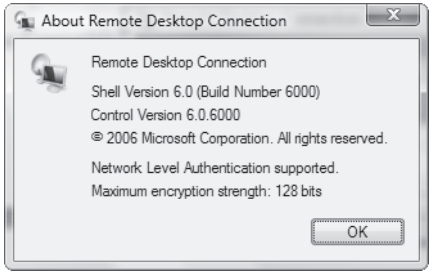

Figure 7-12 Verifying NLA support

Logon Settings Tab

The Logon Settings tab, shown in Figure 7-13, enables you to configure all Terminal Services clients to use a single pre-defined username and password. Sharing credentials in this way enables users to connect to the terminal server without having to supply any credentials. Choosing this option might be suitable for testing environments or for public terminals.

When you select the Always Prompt For Password option, the user must always supply at least a password (if not the username) before connecting.

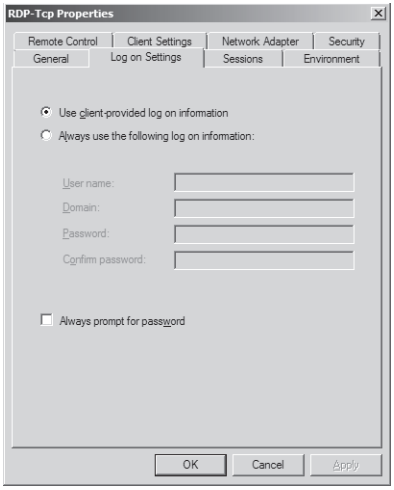

Figure 7-13 Configuring Terminal Services logon settings

Sessions Tab

You can use the Sessions tab to control session timeout settings for the terminal server. Specifically, this tab enables you to choose timeout settings for disconnected sessions, set time limits for active and idle sessions, and define the behavior for disconnections and session limits.

NOTE Terminal Services connections vs. sessions

Strictly speaking, what is the difference between a Terminal Services connection and session? A Terminal Services connection is merely an open Remote Desktop Connection window displaying a desktop on a remote computer. A Terminal Services session, however, is a continuous period during which a user is logged on to a remote computer. If you were to close a Remote Desktop Connection window without logging off from a remote computer, the connection would end, but (provided that the server settings allow it) the session would continue. If you then reconnected to the remote server, you would find the same session in progress with the open programs and files exactly as you had left them.

By default, these settings are defined not in the RDP-Tcp Properties dialog box but in each user's domain account properties. To override these user-defined settings, you can click the Override User Settings check box, as shown in Figure 7-14, and then choose options for the following policies:

- **End a Disconnected Session** This setting determines when (if ever) a user is automatically logged off from a disconnected session.
- **Active Session Limit** This setting determines how long a user can stay active within a Terminal Services session before automatically being disconnected.
- **Idle Session Limit** This setting determines how long a user can leave an inactive connection open to a Terminal Services session before automatically being disconnected.
- **When Session Limit Is Reached Or Connection Is Broken** This setting determines whether a user is logged off automatically when a connection is broken (manually or automatically).

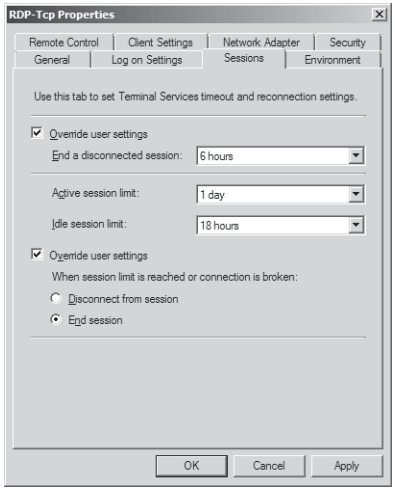

Figure 7-14 Terminal Service timeout and reconnection settings

Environment Tab

This tab enables you to control whether initial programs defined in a user's profile should be allowed to run automatically at the start of a Terminal Services session. It also enables you to specify a program to start for all users connecting to the local terminal server through RDP.

The Environment tab is shown in Figure 7-15.

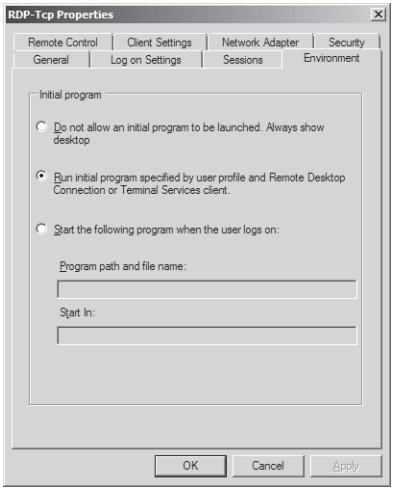

Figure 7-15 Initial program settings

Remote Control Tab

The remote control feature of Terminal Services enables an administrator to see or interact with another user's Terminal Services session. By default, the properties that define the behavior of this feature are set on a per-user basis in each user account's properties dialog box. (These properties define how an administrator can view or control that user's Terminal Services sessions.) The Remote Control tab enables you to control the settings of this feature on a per-server basis instead.

The default settings of a user account, allows an administrator to interact with another user's Terminal Services session only if the user provides consent. However, you can use the Environment tab of the RDP-Tcp Properties dialog box to allow administrators to interact with (or merely to view) all user sessions with or without consent. You can also prevent administrators from viewing or interacting with other users' session's altogether.

IMPORTANT

You can only use the Remote Control feature from within an RDP session. If an administrator is logged on to a terminal server locally, the feature is disabled.

The Remote Control tab is shown in Figure 7-16.

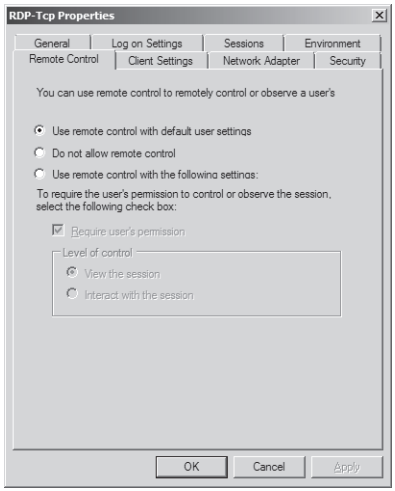

Figure 7-16 Remote control settings

Client Settings Tab

The Client Settings tab provides settings that control how features of the remote terminal server are redirected to the local computer.

The Client Settings tab is shown in Figure 7-17.

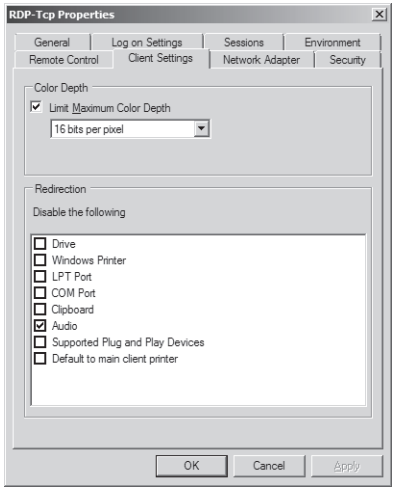

Figure 7-17 The Client Settings tab

In the Color Depth area of the tab, you can define the amount of color detail sent from the Terminal Server to the client. The default setting is 16 bits per pixel, but you can adjust this higher

or lower. In general, when you require more bit depth for RDP connections, appearance is improved at the expense of performance.

In the Redirection – Disable The Following area of the tab, you can determine which features on the remote server should not be redirected to the local computer. The advantage of disabling redirection is improved performance, but this improvement comes at the expense of the advantages offered by each particular feature that you choose to disable.

- **Drive** When you select this option, the drives local to the client cannot be included in the Terminal Services connection. (To include the drives, this check box must be cleared, and the Drives option must be selected on the Local Resources tab of the Remote Desktop Connection client.)
- **Windows Printer** When you select this option, printers local to the client cannot be accessed in the Terminal Services connection. However, a user can still connect to the client printer at the command prompt by using LPT port mapping or COM port mapping.
- **LPT Port** Selecting this option prevents users from mapping a connection to an LPT printer.
- **COM Port** Selecting this option blocks a connection from the Terminal Services session to COM devices on the client computer.
- **Clipboard** This option, when selected, prevents users from cutting or copying data from a Remote Desktop (Terminal Services) session and then pasting that data into the local session on the client computer. Over slow connections, disabling clipboard redirection can prevent screen freezes.
- **Audio** When enabled, this option prevents the transmission of audio data from the remote desktop to the local client computer. This is the only option that is selected by default.
- **Supported Plug and Play Devices** This option, when selected, prevents Plug and Play devices local to the client from being redirected to a Terminal Services session.
- **Default to Main Client Printer** When you select this option, the default printer assigned to the Terminal Services client is prevented from serving as the default printer for the Terminal Services session.

Network Adapter Tab

This tab enables you to restrict the default RDP-Tcp connection to listen for RDP connection attempts on only one particular network adapter. The tab also enables you to set a limit on the number of connections allowed by the terminal server. By default, no limit is set, as shown in Figure 7-18.

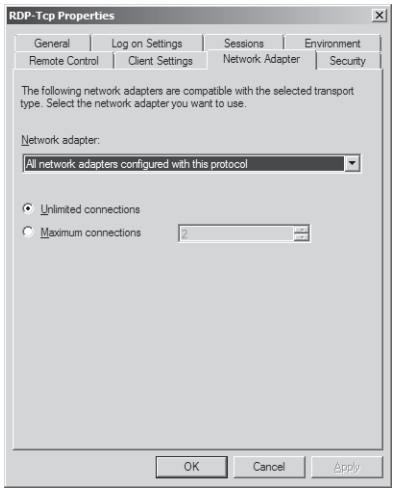

Figure 7-18 The Network Adapter tab

Security Tab

This tab enables you to set user permissions for all RDP connections to the terminal server. It is recommended that you do not use this tab to configure user access to Terminal Services; for that, use the Remote Desktop Users group instead. You can use this tab to determine which users should have administrative control (Full Control) of Terminal Services.

The Security tab is shown in Figure 7-19.

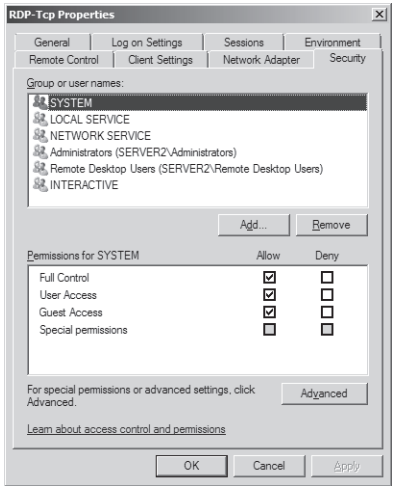

Figure 7-19 The RDP-TCP Properties Security tab

Configuring Terminal Services Server Properties

Besides the RDP-Tcp Properties tabs, the TSC console offers a second important set of Terminal Services configuration options, available through the Edit Settings area. These settings apply to the entire terminal server only; unlike RDP-Tcp or other connection settings, they cannot be configured to apply merely to one transport protocol or to one particular network adapter.

The Edit Settings area provides a summary of seven terminal server options organized under three categories: General, Licensing, and TS Session Broker. To change these server options, double-click on any one of them. This procedure opens a Properties dialog box whose three tabs are also named General, Licensing, and TS Session Broker.

The options available in these three tabs are explained as follows.

General Tab

The General tab enables you to configure the following features related to user logon sessions:

- **Delete Temporary Folders On Exit** When this option is enabled, as it is by default, all temporary data is deleted when a user logs off from a Terminal Services session. Deleting temporary data in this way decreases performance but improves privacy because it prevents users from potentially accessing another user's data.

  This setting functions only when the next option, Use Temporary Folders Per Session, is also enabled.
- **Use Temporary Folders Per S ession** Enabled by default, this option ensures that a new folder to store temporary data is created for each user session. When this option is disabled, temporary data is shared among all active sessions. Sharing temporary data among users can improve performance at the expense of user privacy.
- **Restrict Each User To a Single Session** This option is enabled by default. When enabled, it allows only one logon session to the terminal server per user. For instance, if you are logged on to a server locally with the built-in Administrator account, you cannot log on to the same computer through a Remote Desktop connection by using the same Administrator account until you first log off the server locally.

  By ensuring that you log off one session before beginning another, this default setting prevents possible data loss in the user profile. It also prevents stranded user sessions and, therefore, conserves server resources.
- **User Logon Mode** User logon mode is a Terminal Services feature that you can enable to prevent new users from logging on to the terminal server, usually in advance of a reboot. The Allow All Connections option is the default setting. To prevent users from connecting to the terminal server indefinitely, you can select the Allow Reconnections, But Prevent New Logons option. To prevent users from connecting to the server only until you reboot the server, you can select the Allow reconnection, But Prevent New

Logons Until The Server Is Restarted option. Note that user logon mode does not terminate current sessions. If you need to reboot a server, you might need to end these sessions manually.

The General tab is shown in Figure 7-20.

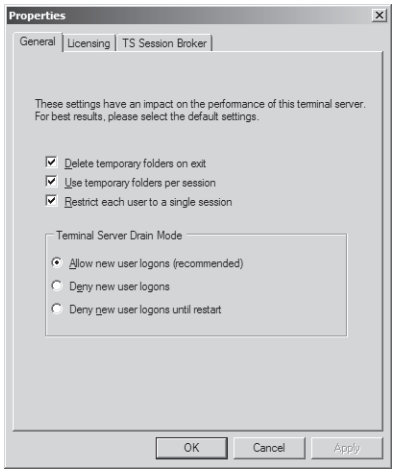

Figure 7-20 Edit Setings Properties dialog

Exam Tip Because Terminal Server User Logon Mode is a new option in Windows Server 2008, expect to see a question on it on the 70-643 exam.

Licensing Tab

The Licensing tab, shown in Figure 7-21, enables you to configure two features related to terminal server licensing: Use The Specified License Server and Automatically Discover A License Server.

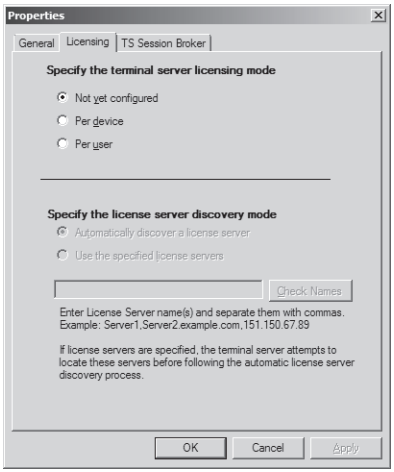

Figure 7-21 Server Options Licensing tab

- **Use The Specified License Servers** You can specify the licensing mode of the terminal server during the installation of the Terminal Services server role, but during this installation process, you can also select the option to configure the licensing mode later. To set or reset the licensing mode after installation, select the server properties Licensing tab and then choose the Per Device or Per User option in the Specify The License Server Disocvery Mode area.
- **Automatically Discover a License Server** The license server discovery mode is the method by which a terminal server contacts a license server to obtain TS CALs. By default, the discovery mode is set to Automatically Discover A License Server. In the automatic license server discovery process, a terminal server attempts to contact any license servers published in the Active Directory directory service or installed on domain controllers in the local domain. As an alternative to automatic discovery mode, you can specify the license server manually by selecting the Use The Specified License Servers option and by then typing a license server name or address in the associated text box.

TS Session Broker Settings Tab

The TS Session Broker Settings tab, shown in Figure 7-22, is used to configure settings for a member server in a TS Session Broker farm. TS Session Broker can be used to load balance among the servers in a farm by directing new user sessions to the server in the farm with the fewest sessions. TS Session Broker is also used to ensure that users can reconnect automatically to disconnected sessions on the appropriate farm member server.

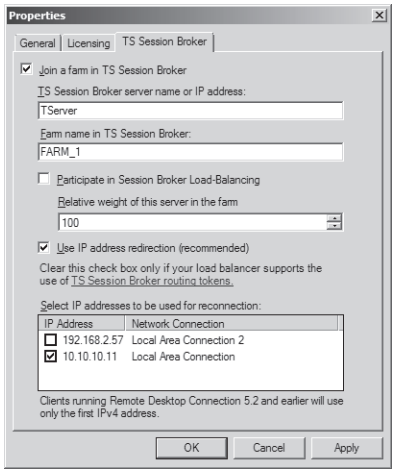

Figure 7-22 Configuring Terminal Services load balancing

To configure a terminal server farm, the first step is to install the TS Session Broker role service on the server that you want to use to track user sessions for the farm. Then, you need to add the terminal servers in the farm to the Session Directory Computers local group on the TS Session Broker server. Finally, you have to configure the terminal servers to join the farm by configuring the following desired options on this tab:

- **Join A Farm In TS Session Broker** Select this check box to add the local server to a farm and to make the remaining options available for configuration.
- **TS Session Broker Server Name Or IP Address** In this text box, type the name or IP address of the server on which you have installed the TS Session Broker role service.
- **Farm Name Is TS Session Broker** In this text box, you must type the name of the farm that will be shared by all farm members. This name also represents the Domain Name System (DNS) name that clients will use to connect to the terminal server farm. (For this reason, in the appropriate DNS server, be sure to add multiple DNS records that correspond to this farm name and that specify the IP address of each farm member.)
- **Participate In Session Broker Load-Balancing** Select this check box to configure the local server to participate in the load balancing feature enabled by TS Session Broker.
- **Relative Weight Of This Server In The Farm** You can use this setting to give powerful servers a larger proportion of user sessions than less powerful servers. For example, if you assign a powerful server a weight of 200 and a less powerful server a weight of 100, the first server will receive twice the number of sessions as the second server.
- **Use IP Address Redirection (Recommended)** Session Broker can use two methods to redirect a client to a disconnected session: IP address redirection and routing token redirection. IP address redirection is enabled by default and is suitable in most scenarios. This redirection method works when the clients can connect to each terminal server in

the farm directly. Clear this check box only if your terminal services clients cannot connect to all terminal servers in the farm and when your network load balancing solution supports the use of TS Session Broker routing tokens.

- **Select IP addresses to be used for reconnection** Use this section to select the IP address that you want to enable for use in the terminal server farm.

Exam Tip For both the 70-643 exam and the real world, remember that you need to add each farm member to the Session Directory Computers local group on the TS Session Broker server.

IMPORTANT

To distribute the initial connections to the server farm, Session Broker load balancing must rely on a network load balancing solution such as DNS round-robin, Network Load Balancing, or a hardware load balancer.

Configuring Terminal Services Printer Redirection

Printer redirection is a feature that enables the client's printers to be used as printers for a Terminal Services session. Although you can modify basic options regarding printer redirection easily in the Client Settings tab of the RDP-Tcp Properties dialog box, important additional options concerning this feature are made available in Group Policy.

You can disable or customize the behavior of printer redirection by using Group Policy and the Group Policy Management console. To find printer redirection configuration options in Group Policy, open a Group Policy object (GPO) and navigate to Computer Configuration \Administrative Templates\Windows Components\Terminal Services\Terminal Server\Printer Redirection. Within the Printer Redirection folder, you can configure the following five policy settings:

- **Do Not Set Default Client Printer To Be Default Printer In A Session** By default, Terminal Services automatically designates the client default printer as the default printer in a Terminal Services session. You can use this policy setting to override this behavior. If you enable this policy setting, the default printer will be designated as the printer specified on the remote computer. (This setting is equivalent to selecting the Default To Main Client Printer option on the Client Settings tab in the RDP-Tcp Properties dialog box on all servers that fall within the scope of the policy.)
- **Do Not Allow Client Printer Redirection** This policy setting essentially disables printer redirection completely. If you enable this policy setting, users cannot redirect print jobs from the remote computer to a local client printer in Terminal Services sessions. (This setting is equivalent to selecting the Windows Printer option on the Client Settings tab in the RDP-Tcp Properties dialog box on all servers that fall within the scope of the policy.)

- **Specify Terminal Server Fallback Printer Driver Behavior** This policy setting determines the behavior that occurs when the terminal server does not have a printer driver that matches the client's printer. By default, when this occurs, no printer is made available within the Terminal Services session. However, you can use this policy setting to fall back to a Printer Control Language (PCL) printer driver or a PostScript (PS) printer driver. Alternatively, both printer drivers should be used as a fallback in case no suitable driver is found for the client printer.
- **Use Terminal Services Easy Printer Driver First** The Terminal Services Easy Printer driver enables users to print from a terminal server session to the correct printer on their client computer reliably. It also enables users to have a more consistent printing experience between local and remote sessions. By default, the terminal server first tries to use the Terminal Services Easy Printer driver to install all client printers. However, you can use this policy setting to disable the use of the Terminal Services Easy Printer driver.
- **Redirect Only The Default Client Printer** By default, all client printers are redirected to Terminal Services sessions. However, if you enable this policy setting, only the default client printer is redirected in Terminal Services sessions.

Exam Tip Be sure to understand these Group Policy settings for the 70-643 exam.

Quick Check

1. You want to prepare to take a server in a server farm offline. You do not want to force any users off. What should you do?
2. You want to enable audio in Terminal Services connections to a server named TS1. What should you do?

Quick Check Answers

1. Configure Terminal Server User Logon Mode to deny new user logons.
2. Clear the Audio check box on the Client Settings tab in RDP-Tcp properties on TS1.

Practice: Installing and Configuring a License Server

After you have purchased TS CALs from Microsoft or a third-party reseller, you need to install and activate the license server. In this exercise, you will install a Terminal Services license server on Server1. You will then open the TS Licensing Manager console to review the procedures for activating a license server and installing TS CALs.

▶ Exercise 1: Install the TS Licensing Server Role

In this exercise, you will use the Add Roles Wizard to install a Terminal Services license server on the Contoso.com domain controller.

1. Log on to Contoso.com from Server1 as a domain administrator.
2. Open Server Manager.
3. Scroll down the Terminal Services section, and then click Add Role Services.
4. On the Select Role Services page, select the TS Licensing check box and then click Next.
5. On the Configure Discovery Scope For TS Licensing page, read all the text on the page.

   Note that you can configure the license server for the local Active Directory domain or for the entire forest in a multi-domain environment. The current Active Directory environment is composed of a single-domain forest.
6. On the Configure Discovery Scope For TS Licensing page, leave the default selection of This Domain and then click Next.
7. On the Confirm Installation Selections page, read all the text on the page and then click Install.
8. When the installation completes, the Installation Results page appears.
9. On the Installation Results page, click Close.

▶ Exercise 2: Activate a Terminal Services Licensing Server

In this exercise, you will activate the license server and review the process for installing TS CALs. This process requires Server1 to be connected to the Internet.

1. While you are logged on to Server1 as a domain administrator, open the TS Licensing Manager console by clicking Start, pointing to Administrative Tools, pointing to Terminal Services, and then clicking TS Licensing Manager.

   The TS Licensing Manager console opens.

   Although TS Licensing Manager is installed automatically on any server on which you have installed the TS Licensing role service, you do not need to manage the licensing server from the server itself. You can also install TS Licensing Manager on any server and connect to the license server remotely.
2. In the TS Licensing Manager console tree, expand the All Servers node and then select the SERVER1 node. (The node should be marked by a red X at this point because it has not been activated.)
3. Right-click the SERVER1 node and then click Activate Server.

   The Activate Server Wizard appears.
4. On the Welcome To The Activate Server Wizard page, read all the text on the page and then click Next.
5. On the Connection Method page, read all the text on the page and then answer the following question: By default, what is the default Connection Method assigned to the license server?

   Automatic Connection (Recommended)

6. On the Connection Method page, in the Connection Method drop-down list, select Web Browser.
7. Read the new associated Description and Requirements sections that have been refreshed on the page. The Web Browser connection method is useful when the licensing server does not connect to the Internet but rather when you can connect to both the licensing server and the Internet from a third server.
8. On the Connection Method page, in the Connection Method drop-down list, select Telephone.
9. Read the new associated Description and Requirements sections that have been refreshed on the page. The Telephone connection method is useful when your network is not connected to the Internet.
10. On the Connection Method page, in the Connection Method drop-down list, select Automatic Connection and then click Next.

    The Activate Server Wizard dialog briefly appears while Server1 contacts the activation server at the Microsoft Clearinghouse. After a moment, the Company Information page appears.
11. On the Company Information page, enter appropriate information in the First Name, Last Name, and Company text boxes. Then, choose your country from the Country Or Region drop-down list.
12. Click Next.
13. Another Company Information page appears. You may optionally provide the requested information. Click Next..

    The Activate Server Wizard dialog appears briefly, and then the Completing The Activate Server Wizard page appears. Note that the Start Install Licenses Wizard Now check box is selected.
14. On the Completing The Activate Server Wizard page, read all the text and then click Next.

    The Welcome To The Install Licenses Wizard page appears.
15. Leave all windows open and proceed to Exercise 3.

▶ Exercise 3: Review the Process to Install TS CALs

Installing client licenses is the last stage of deploying a license server. Even if you do not have any TS CALs to install at this point, it is a good idea to review the pages of the Install Licenses Wizard to gain a better understanding of this deployment process in its entirety.

1. In this exercise, you will review the process of installing TS CALs in your newly activated server.

   On the Welcome To The Install Licenses Wizard page, read all the text on the page and then click Next. This page is shown in Figure 7-23.

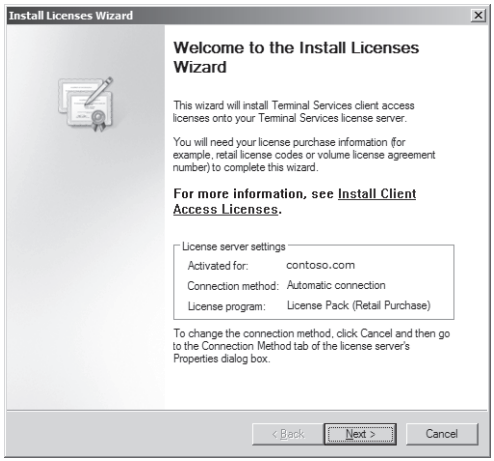

Figure 7-23 The Welcome page of the Install Licenses Wizard

The Install Licenses page briefly appears, and then the License Program page appears. The License Program page is shown in Figure 7-24.

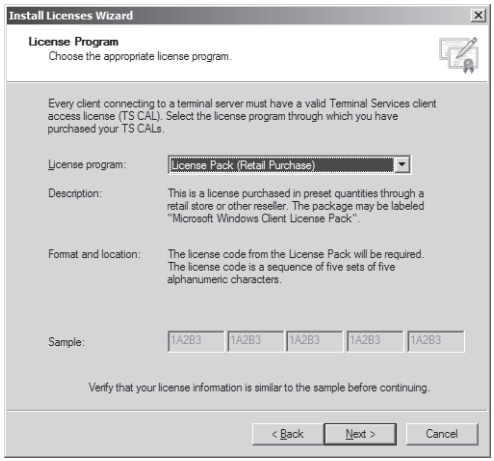

Figure 7-24 The License Program page of the Install Licenses Wizard

Read all the text on the License Program page.

Review the options from the License Program drop-down list.

Take a few moments to explore the various license program options by selecting each option and reading all of the associated text on the page.

In the License Program drop-down list, ensure that the default option of License Pack (Retail Purchase) is selected, and then click Next.

The License Code page appears. The License Program page is shown in Figure 7-25.

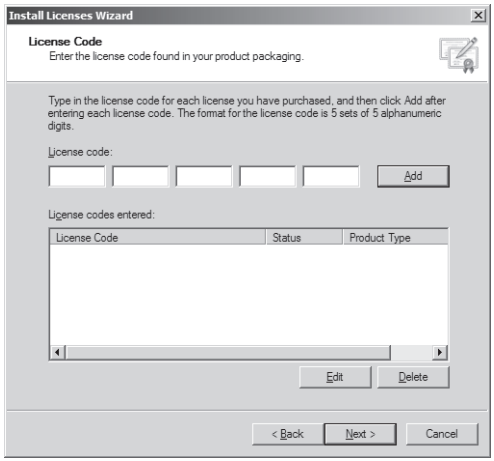

Figure 7-25 The License Code page of the Install Licenses Wizard

2. Read all the text on the page.

   If you have obtained a valid license code, you can perform the remaining steps of the practice. Otherwise, you can click Cancel to close the wizard and then simply read the remaining steps of the practice.

3. In the License Code text boxes, type a valid license code and then click the Add button.

4. On the License Code page, click Next.

   The Install Licenses Wizard dialog briefly appears, and then the Completing The Install Licenses Wizard page appears. This page is shown in Figure 7-26.

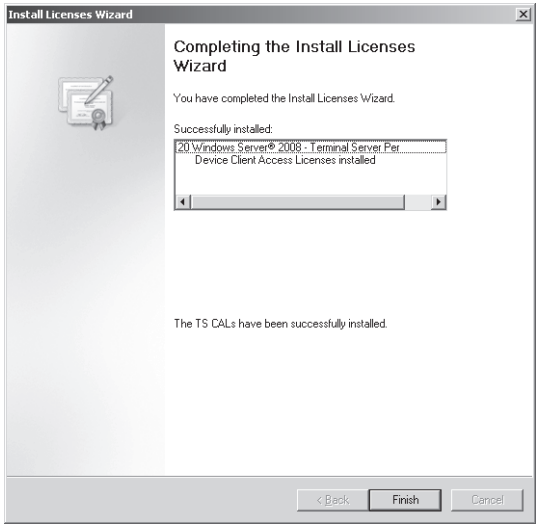

Figure 7-26 The Completing page of the Install Licenses Wizard

5. On the Completing The Install Licenses Wizard page, click Finish.
6. In the TS Licensing Manager console tree, the Server1 node is now designated with a green check mark, as shown in Figure 7-27. The licensing server is now configured.
7. Close all open windows and log off Server1.

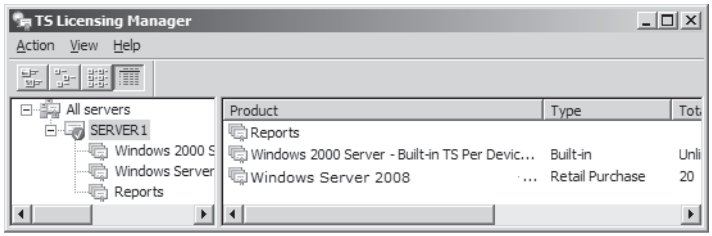

Figure 7-27 Successful deployment of a licensing server

Lesson Summary

- The main tool used for configuring Terminal Services is the Terminal Services Configuration console.
- You can edit RDP-Tcp properties to configure Terminal Services user session features such as encryption strength, timeout settings, and printer availability.
- The Terminal Services server properties enable you to configure features such as load balancing, license server discovery, and Terminal Server User Logon Mode.
- Group Policy offers additional control for Terminal Services printer redirection, notably for the option to fall back to a generic printer driver and to redirect only the default client printer.

Lesson Review

The following questions are intended to reinforce key information presented in this lesson. The questions are also available on the companion CD if you prefer to review them in electronic form.

NOTE Answers

Answers to these questions and explanations of why each answer choice is correct or incorrect are located in the "Answers" section at the end of the book.

---

1. Your company network has implemented a terminal server farm named TSFARM1. The farm consists of five computers running Windows Server 2008, including a server named TSLB1 on which the TS Session Broker role service is installed. You want to add a sixth computer running Windows Server 2008, named TSLB6, to the farm. After configuring the server with the same hardware and software options as those of the other

farm members, you attempt to join TSLB6 to the farm by specifying TSLB1 as the Session Broker Server and TSFARM1 as the farm name in the TS Session Broker properties on TSLB6. You verify that some users who attempt to connect to the virtual server name TSFARM1 are able to establish Terminal Services sessions on TSLB6, but these users are not able to reconnect to disconnected sessions. You want users connecting to TSLB6 through TSFARM1 to be able to reconnect to disconnected RDP sessions. What should you do?

A. Add TSLB6 to the Session Directory Computers local group on TSLB6.

B. Add TSLB6 to the Session Directory Computers local group on TSLB1.

C. In the DNS server, add a Host (A) record named TSFARM1 that maps to the IP address of TSLB6.

D. In the DNS server, add a Host (A) record named TSLB6 that maps to the IP address of TSLB6.

2. Your company network consists of a single Active Directory domain named contoso.com. In the company network, you have deployed Terminal Services on a computer named TS1 that is running Windows Server 2008. Some users who connect to TS1 through RDP complain that they cannot print to their local printers successfully. You want to ensure that TS1 uses a generic PostScript printer driver whenever Terminal Services cannot find an adequate driver for Terminal Services client printers. What should you do?

A. On the Client Session tab of RDP-Tcp properties on TS1, select the Windows Printer option.

B. On the Client Session tab of RDP-Tcp properties on TS1, select the Default To Main Client Printer option.

C. In a Group Policy object (GPO), configure the User Terminal Services Easy Printer Driver First policy setting, and then apply the GPO so that TS1 falls within the scope of the policy.

D. In a Group Policy object (GPO), configure the Specify Terminal Server Fallback Printer Driver policy setting with the PS option and then apply the GPO so that TS1 falls within the scope of the policy.

Chapter Review

To further practice and reinforce the skills you learned in this chapter, you can:

- Review the chapter summary.
- Review the list of key terms introduced in this chapter.
- Complete the case scenario. This scenario sets up a real-world situation involving the topics of this chapter and asks you to create solutions.
- Complete the suggested practices.
- Take a practice test.

Chapter Summary

- Terminal Services allows users to establish and interact with a desktop session on a remote computer. In Windows Server 2008, Terminal Services includes many new and important features such as TS Gateway, RemoteApp, and TS Web Access.
- Terminal Services requires client access licenses (CALs) either for all connecting users or for all connecting devices. If you do not purchase and install Terminal Services CALs, the feature will stop working after 120 days.
- To install Terminal Services on a computer running Windows Server 2008, add the Terminal Services server role.
- The main tool used for configuring Terminal Services is the Terminal Services Configuration (TSC) console. In the TSC console, you can edit RDP-Tcp properties to configure Terminal Services user session features such as encryption strength, timeout settings, and printer availability. You can also edit server properties to configure features such as load balancing, license server discovery, and Terminal Server User Logon Mode.

Key Terms

Do you know what these key terms mean? You can check your answers by looking up the terms in the glossary at the end of the book.

- Network Level Authentication (NLA)
- Printer Redirection
- Remote Desktop for Administration (RDA)
- Remote Desktop Protocol (RDP)
- Terminal Server User Logon Mode
- Terminal Services connection

- Terminal Services session
- Terminal Services Client Access License (TS CAL)
- TS Session Broker

Case Scenarios

In the following case scenario, you will apply what you've learned in this chapter. You can find answers to these questions in the "Answers" section at the end of this book.

Case Scenario 1: Choosing a TS Licensing Strategy

You work as a network administrator in a large company. Your department has implemented two terminal servers recently, named TS1 and TS2, and you have been tasked with making licensing recommendations for each server.

TS1 is an application server. Although the application is not considered mission critical, as many as five users tend to be connected to it simultaneously. Overall, 20 users need to connect to TS1 at some point during the day. They can connect from any of 50 different computers.

TS2 is a DNS server that occasionally requires remote maintenance and administration. Only administrators connect to TS2.

1. Do you need to install Terminal Services on TS1? Which type of client access licenses would you purchase, if any?
2. Do you need to install Terminal Services on TS2? Which type of client access licenses would you purchase, if any?

Case Scenario 2: Troubleshooting a Terminal Services Installation

You work in IT support for a large company whose network consists of a single Active Directory domain. One of your responsibilities is supporting terminal servers in the Advertising department. Over the course of a week, you encounter the following two problems:

1. You deploy Terminal Services on a new computer running Windows Server 2008 named App3, but you discover that no users running Windows XP can connect to it. What should you do?
2. Users that connect to a Terminal Server named App1 complain that they can not always reconnect to a disconnected session. What should you do?

Suggested Practices

To help you successfully master the exam objectives presented in this chapter, complete the following tasks.

Deploy a Terminal Server Farm

- **Practice: Create a load-balanced terminal server farm** Using either virtual or physical computers, join two identical installations of Windows Server 2008 to a domain. Install the Terminal Server role service on both computers, but the TS Session Broker role service on just one. Add both computer names to the Session Directory Computers local group on the Session Broker computer. Use the TS Session Broker tab in the Terminal Services Configuration console on both computers to configure the Terminal Services farm. Create Host (A) records for the farm name in DNS, one record for each server IP address. Then, connect to the server farm through from a remote RDP client.

Watch a Webcast

- **Practice: Watch a Webcast about Terminal Services in Windows Server 2008** Watch the Webcast "A Technical Overview of Windows Server 2008 Terminal Services," by Blain Barton, available on the companion CD in the Webcasts folder.

Take a Practice Test

The practice tests on this book's companion CD offer many options. For example, you can test yourself on just one exam objective, or you can test yourself on all the 70-643 certification exam content. You can set up the test so that it closely simulates the experience of taking a certification exam, or you can set it up in study mode so that you can look at the correct answers and explanations after you answer each question.

MORE INFO Practice tests

For details about all the practice test options available, see the "How to Use the Practice Tests" section in this book's introduction.
