Microsoft

Administrator's Pocket Consultant

Microsoft IIS 6.0

William R. Stanek

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Cataloging-in-Publication Data pending.

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 QWE 8 7 6 5 4 3

Distributed in Canada by H.B. Fenn and Company Ltd.

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at www.microsoft.com/mspress. Send comments to *mspinput@microsoft.com*.

Acquisitions Editor: Jeff Koch
Project Editors: Valerie Woolley, Julie Miller
Technical Editor: Bob Hogan

Body Part No. X08-64090

Table of Contents

Part I
Microsoft Windows Server 2003 Web Administration Fundamentals

Part II
Web Server Administration

4 Customizing Web Server Content 69

Part III

Essential Services Administration

12 Administering the Indexing Service 305

Part IV
Performance, Optimization, and Maintenance

13 Performance Tuning and Monitoring 331

Part I

Microsoft Windows Server 2003 Web Administration Fundamentals

The goal of Part I of this book is to introduce you to IIS 6.0, focusing on the fundamentals you need for Web administration. Chapter 1 provides an overview of Web administration tools, techniques, and concepts. Chapter 2 explores core Web administration using Microsoft Internet Information Services (IIS). You'll learn about administration components, Internet Information Services (IIS) Manager, and server configurations.

Chapter 1

Overview of Microsoft Web Services

Each new version of Microsoft Internet Information Services (IIS) has represented a major advance in Web server technology. The changes have been dramatic, and they've improved reliability, availability, scalability, manageability, and security. However, no version of IIS has brought the kinds of changes you'll find in IIS 6.0—so if you think you know IIS 6 because you knew a previous version, think again.

Microsoft's entire .NET strategy is tied to IIS 6., so much so that you can think of IIS as the heart of Web application services within the Microsoft Windows .NET Framework. IIS is no longer a simple bundle of services for putting up a Web site—it's a complete solution for hosting Web servers and Web applications, and the Web application architecture is one of the most versatile you'll find anywhere.

IIS 6 has been redesigned from the bottom up. For starters, ASP.NET and the Windows .NET Framework are fully integrated into IIS 6, which significantly changes the way you use IIS. Further, unlike IIS 5, where the main Web server process was often a major choke point that severely affected performance, IIS 6 has a redesigned request processing architecture that allows the server to perform better, to reserve fewer resources, to handle more virtual servers, to detect failures and resolve them, and much more.

IIS 6 has many other new and enhanced features. Few are more important than the changes to the security architecture. IIS 6 has multiple levels of security, and it adds authentication mechanisms (including .NET Passport authentication and delegated authentication),improves Secure Sockets Layer (SSL) by enhancing performance and adding support for crypto service providers, and supports Uniform Resource Locator (URL) authorization whereby administrators can control access according to applications and URLs.

Because of the many changes, a lot of what you know about IIS is obsolete or irrelevant. But it's not all bad news. There's a light at the end of the tunnel—well, it's more like a freight train coming right at you—but it's there. The changes in IIS 6 are well worth the time and effort you'll spend learning the new architecture and the new techniques required to manage Web servers. Our dependence on ASP.NET and Windows .NET Framework will only grow over time, and the more you learn about the heart of the .NET architecture—IIS 6—the better prepared you'll be for now and for the future.

Note Throughout this book I'll refer to administration of IIS, Web applications, and the Indexing Service as *Microsoft Web administration* or simply *Web administration*. Microsoft Indexing Service is used to create text indexes of the contents and properties of files so that the files can be searched using standard queries.

As you get started with Microsoft Web administration, you should concentrate on these key areas:

- What's new or changed in IIS 6
- How IIS works with your hardware
- How IIS works with Microsoft Windows–based operating systems
- Which administration tools are available
- Which administration techniques you can use to manage and maintain IIS

Note In this book, the term *Windows Server 2003* refers to these members of the Microsoft Windows Server 2003 family: Windows Server 2003, Standard Edition; Windows Server 2003, Enterprise Edition; Windows Server 2003, Datacenter Edition; and Windows Server 2003, Web Edition. In addition, all procedures described in this book are based on the default version of Windows Server 2003; if you are using the Classic Start menu, some of the steps will be slightly different.

Introducing IIS 6

Internet Information Services (IIS) is designed to provide secure, scalable solutions for creating and managing World Wide Web sites and servers. You can use IIS to publish information on intranets, extranets, and the Internet. Because today's Web sites use related services, like File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), ASP.NET, and Windows .NET Framework, IIS bundles these services as part of a comprehensive offering. A separate but related service is the Indexing Service, which is used to build catalogs of documents that can be searched. When you add this capability to a Web site, it allows users to search for topics of interest using a standard Hypertext Markup Language (HTML) form.

IIS 6 Request Processing Architecture

Unlike IIS 5, where the main Web server process was often a major choke point that severely affected performance, IIS 6 has a redesigned request processing architecture that allows the server to perform better, to reserve fewer resources, to handle more virtual servers, to detect failures and resolve them, and much more. This architecture has several key features:

- **HTTP listener process** In IIS 6, the main Web server process is a kernel-mode driver called Http.sys. It's used for Hypertext Transfer Protocol (HTTP) parsing and caching. It's responsible for listening for requests and passing them off to worker processes.

- **Worker processes** Worker processes run in an isolated mode that allows administrators to group different Web applications. Worker processes are isolated by application pool and can be allocated on demand, meaning they're allocated system resources when they become active and don't use system resources when they're inactive. This architecture improvement, along with others, ensures that IIS 6 can support many more concurrent processes than previous versions.
- **Application pools** Groups of Web applications are called *application pools*. Application pools are separated from one another by process boundaries and are serviced by one or more worker processes, which applications in the pool share. All Web sites and applications on a server are assigned to an application pool. Settings for application pools allow you to monitor worker processes and to recover automatically from any problems that might occur.
- **Application pool request queue** When requests are passed off from Http.sys to worker processes, the requests are placed in the appropriate application pool request queue. Each application pool has a separate request queue. Worker processes assigned to the application pool handle the request in first in, first out (FIFO) order. You can assign worker processes a processor affinity so that specific processors handle their workload.

Although you'll learn even more about the request processing architecture in Chapter 2, "Core IIS Administration," these two chapters only scratch the surface of the dramatic change the new architecture represents. To understand the architecture completely, you'll need to read the chapters in Part II, "Web Server Administration." These chapters discuss site, server, and application configuration; worker process assignment; and application pool configuration.

IIS 6 Security Architecture

The security architecture is another major area where IIS 6 has been redesigned. The new security architecture has several important features that you should know about right now:

- **Capability lockdown** IIS isn't installed by default on Windows Server 2003. When you install IIS, the default installation allows only static content (HTML files) to be served, and all other functions and types of content must be specifically enabled. Nonstatic content is managed through the Web Service Extensions settings. See Chapter 4, "Customizing Web Server Content," for details. Further, if you upgraded the operating system on a server that was previously running IIS, the IIS service might be disabled. To reenable IIS, you might need to enable the IIS service as well as the associated services.
- **Privilege changes** By default, many IIS 6 features run using the built-in account NetworkService. This account has very few privileges and is designed to ensure that IIS and related processes have very few privileges on the server. Although good for security and reducing potential vulnerabilities, it might change the way you use IIS, and some applications or features

might work differently than you expect. Be sure to take a look at this account's privileges.

- **Tool and file restrictions** IIS won't serve requests for invalid files. It verifies all file requests before serving them, checking file extensions and for the existence of the requested content. IIS won't run command-line tools or other command-line executables.
- **Authentication enhancements** IIS has a number of enhancements for authenticating requests, including URL authorization and delegated authentication, but the most important change is without doubt the support for .NET Passport authentication. Through their .NET Passport identification, users can be validated and authorized access according to the access controls in their corresponding Active Directory service user account.

> **Real World** There's a way to run Web applications in IIS 5 mode. It's called IIS 5 isolation mode. Although operating in this mode might solve problems with applications that won't run under the new IIS 6 architecture, security restrictions might also be affecting the way applications are running. Be sure to read Chapter 7, "Enhancing Web Server Security," so that you understand the changes to the security architecture.

Additional IIS 6 Features

IIS 6 has many additional features. Some that you'll want to learn about include:

- **FTP restart** FTP restart allows clients to resume FTP downloads without having to download the entire file again if an interruption occurs during transfer. When a connection is broken during a download, compliant clients (such as Microsoft Internet Explorer 5) can reestablish their file transfer using the REST command, and the file transfer will resume where it left off.
- **FTP user isolation** IIS 6 allows you to isolate users to their own directories so that they can't view or overwrite other users' content.
- **Health monitoring** Just as Windows Server 2003 monitors the health of its running processes, so does IIS 6. IIS 6 takes this monitoring a few steps further, though. It can detect and recover from memory leaks, problems in code, and blocking calls. IIS can also check for nonresponsive processes and then recycle or restart processes as necessary.
- **Host headers** Host headers allow you to host multiple Web sites on a single computer with only one Internet Protocol (IP) address. Here, IIS uses the host name passed in the HTTP header to determine the site that a client is requesting.
- **HTTP 1.1 and HTTP compression** IIS fully supports the HTTP 1.1 protocol and the compression enhancements it defines. Using HTTP compression, you can compress both static and dynamic results of HTTP queries for transmission to HTTP 1.1–compliant clients. Unlike IIS 5, where compression was implemented using an Internet Server Application Programming Interface (ISAPI) filter and could only be enabled for an entire server, IIS 6 builds in compression as a feature that you can control precisely to the file level.

- **Kernel-mode cache** Http.sys runs in kernel mode and passes requests directly to the worker processes without intermediaries. Previously requested static content can be cached, and unlike previous versions of IIS, dynamic content can be cached in kernel mode as well to improve performance. To better support Active Server Pages (ASP), ASP templates are stored in memory and deallocated from memory to free space for new templates. Unlike previous versions, IIS 6 uses a persistent ASP template cache. Here, deallocated templates are written to disk, where they can be accessed and reallocated. IIS 6 also has a heuristics-based caching policy. This policy is designed to ensure that files are cached when it makes sense and aren't cached otherwise.
- **On-demand starting and time-out** You can configure application pools so that worker processes start on demand and time out when they're no longer needed. By starting on demand, the process uses resources only when it's active. By timing out, the resources used by the worker process can be freed up when the process has been idle for a certain amount of time.
- **Process accounting and process throttling** Process accounting provides information about how individual Web sites use CPU resources. Process throttling allows you to limit CPU usage for out-of-process applications and thereby potentially reduce performance problems on the server as a whole.
- **Rapid-fail protection** Rapid-fail protection allows IIS to monitor worker processes for failure. If IIS detects failure, IIS can take actions to record and recover, such as logging a related event in the event logs and restarting the worker process.
- **SSL 3 and TLS** SSL 3 and Transport Layer Security (TLS) provide secure methods of exchanging information between clients and servers. SSL 3 and TLS also enable the use of client certificates that can be read by Internet Server Application Programming Interface (ISAPI) server pages. Client certificates are used to authenticate users and control access by mapping the client certificate to a Windows user account.
- **WebDAV** Web Distributed Authoring and Versioning (WebDAV) extends the HTTP 1.1 protocol and is integrated into IIS. Using WebDAV, remote users can publish, lock, and manage resources on a Web server using an HTTP connection.
- **XML metabase** The IIS metabase is now formatted using Extensible Markup Language (XML) and stored in plaintext files. XML's structure makes it easier to search and maintain the metabase and also improves performance when working with the metabase. The XML metabase can be edited while IIS is running. It can be used to save configurations at the server, site, or application level so they can be used on other servers, which can help ensure that configurations across server farms are exact copies of each other. The metabase also supports automatic versioning and history. This means that IIS automatically tracks changes to the metabase and changes that are made can be rolled back to restore a previous configuration.

Choosing Appropriate Web Server Hardware

Guidelines for choosing hardware for Internet servers are much different from those for choosing other types of servers. A Web hosting provider might host multiple sites on the same computer and might also have service level agreements that determine the level of availability and performance required. On the other hand, a busy e-commerce site might have a dedicated Web server or even multiple load-balanced servers. Given that Internet servers are used in a wide variety of circumstances and might be either shared or dedicated, here are some guidelines for choosing server hardware:

- **Memory** The amount of memory that's required depends on many factors, including the requirements of other services, the size of frequently accessed content files, and the random access memory (RAM) requirements of the Web applications. High-volume servers should have a minimum of 512 MB of RAM. More RAM will allow more files to be cached, reducing disk requests.

Note Don't forget that as you add physical memory, virtual paging to disk grows as well. With this in mind, you might want to ensure that the Pagefile.sys file is on the appropriate disk drive.

More Info For detailed information on memory management and performance tuning, see Chapter 13, "Performance Tuning and Monitoring."

- **CPU** The CPU processes the instructions received by the computer. The clock speed of the CPU and the size of the data bus determine how quickly information moves among the CPU, RAM, and system buses. Static content, such as HTML and images, place very little burden on the processor, and standard Windows Server 2003 recommended configurations should suffice. Faster clock speeds and multiple processors increase the upper capacity of a Web server, particularly for sites that rely on dynamic content.
- **SMP** IIS supports symmetric multiprocessors (SMPs) and can use additional processors to improve performance. If the system is running only IIS and doesn't rely on dynamic content or encryption, a single processor might suffice. You should always use multiple processors if IIS is running alongside other services, such as Microsoft SQL Server or Microsoft Exchange Server.
- **Disk drives** The amount of data storage capacity you need depends entirely on the size of content files and the number of sites supported. You need enough disk space to store all your data plus workspace, system files, and virtual memory. Input/output (I/O) throughput is just as important as drive capacity. However, disk I/O is rarely a bottleneck for Web sites on the public Internet—generally, bandwidth limits throughput. High-bandwidth sites should consider hardware-based redundant array of independent disks (RAID) solutions using copper or fiber channel–based small computer system interfaces (SCSIs).

- **Data protection** Unless you can tolerate hours of downtime, you should add protection against unexpected drive failures by using RAID. RAID 0 (disk striping without parity) offers optimal read/write performance, but if a drive fails, IIS won't be able to continue operation until the drive is replaced. Because of this, RAID 0 isn't the recommended choice. RAID 1 (disk mirroring) creates duplicate copies of data on separate drives, but recovery from drive failure might interrupt operations while you restore the failed drive from backups. RAID 5 (disk striping with parity) offers good protection against single-drive failure but has poor write performance. Keep in mind that if you've configured redundant load-balanced servers, you might not need RAID. With load balancing, the additional servers might offer the necessary fault tolerance.
- **UPS** Sudden power loss and power spikes can seriously damage hardware. To prevent this, get an uninterruptible power supply (UPS). A UPS system gives you time to shut down the system properly in the event of a power outage, and it's also important in maintaining system integrity when the server uses write-back caching controllers that do not have on-board battery backups. Professional hosting providers often offer UPS systems that can maintain power indefinitely during extended power outages.

If you follow these hardware guidelines, you'll be well on your way to success with IIS.

Choosing the Server Operating System

IIS 6, Web applications, and the Indexing Service are designed to run on Windows-based operating systems. Four versions of Windows Server 2003 are available. Each server edition has different features:

- **Windows Server 2003, Web Edition** This version of the software is designed to provide Web services for deploying Web sites and Web-based applications. As such, this server edition includes Windows .NET Framework, IIS, ASP.NET and network load balancing features but lacks many other features, including Active Directory. In fact, the only other key Windows features in this edition are the Distributed File System (DFS), Encrypting File System (EFS), and Remote Desktop for administration. Windows Server 2003, Web Edition, supports up to 2 GB of RAM and two central processing units (CPUs).
- **Windows Server 2003, Standard Edition** This version is designed to provide services and resources to other systems on a network. It's a direct replacement for Windows NT 4 Server and Windows 2000 Server. The operating system has a rich set of features and configuration options. Windows Server 2003, Standard Edition, supports up to 4 GB of RAM and four CPUs.
- **Windows Server 2003, Enterprise Edition** This version extends the features provided in Windows Server 2003, Standard Edition to include support for Cluster Service, Metadirectory services, and Services for Macintosh. It also

supports 64-bit Intel Itanium-based computers, hot-swappable RAM, and Non-Uniform Memory Access (NUMA). Enterprise servers can have up to 32 GB of RAM on x86, 64 GB of RAM on Itanium, and eight CPUs.

- **Windows Server 2003, Datacenter Edition** This version is the most robust Windows server. It has enhanced clustering features and supports very large memory configurations with up to 64 GB of RAM on x86 and 512 GB of RAM on Itanium. It has a minimum CPU requirement of 8 and can support up 32 CPUs in all.

Note The various server editions support the same core features and administration tools. This means you can use the techniques discussed in this book regardless of which edition of Windows Server 2003 you're using.

Most of the time you'll want to consider using Windows Server 2003, Web Edition, for your Web server and Web application server needs. However, as mentioned above, the Web edition has specific feature limitations. If you need Active Directory to be installed on the server (which usually isn't the case), you'll want to install a different version. If you need Cluster Service or other high availability features, you'll need to install Windows Server 2003, Enterprise Edition, or Windows Server 2003, Datacenter Edition.

Other feature limitations of the Web Edition will affect your decision as well. The Web Edition doesn't support the following Windows features: 64-bit Support for Intel Itanium-Based Computers, Cluster Service, Enterprise Universal Description, Discovery and Integration (UDDI) Services, Fax Service, Hot Add Memory, Internet Authentication Service (IAS), Internet Connection Firewall (ICF), Metadirectory Services (MMS) Support, Network Bridge, Internet Connection Sharing (ICS), NUMA, Remote Installation Services (RIS), Removable and Remote Storage, Services for Macintosh, Terminal Server, Terminal Server Session Directory, and Windows Media Services. Check the Windows Server documentation for any changes.

Working with IIS 6: What You Need to Know Right Now

As you've seen, IIS 6 is very different from its predecessors. IIS 6 has a new processing architecture, a new security architecture, and many other enhancements. As you might expect with all these changes, there are many things you should know right away about IIS 6 components, configuration, and services.

Installing Web and Application Server Components and Default Sites

IIS and Indexing Service are no longer installed during the installation of the operating system. You install these and other Web server components through the Windows Components Wizard, accessible through Add Or Remove Programs on the Control Panel. The key Web server components you might want to use include:

- **Certificate Services** Installs a certificate authority to issue public key certificates for use in authentication.
- **E-mail Services** Provides basic Post Office Protocol 3 (POP3) services so that POP3 mail clients can send and receive mail in the domain. Once you install this service, you define a default domain for mail exchange and then create and manage mailboxes. This basic service works well for datacenters and remote locations where e-mail exchange is needed but you don't need the power and versatility of Exchange Server.
- **Indexing Service** Installs indexing service for fast full-text searching of Web documents.
- **Web Application Server** Provides IIS and ASP.NET services. You can install ASP.NET, COM+, Distributed Transaction Coordinator (DTC), IIS, Message Queuing, Microsoft Data Engine, and the Web Application Server Console.

By default, all subcomponents of certificate services, e-mail services, and indexing services are installed when the related option is selected in the Windows Components Wizard. For the Application Server component this isn't the case. You'll want to select this component and then click Details. Then, add components as necessary by selecting them. Some of these subcomponents have subcomponents of their own as well. The one you'll want to check is IIS. In the Application Server dialog box, select Internet Information Services (IIS) and then click Details.

The IIS application server components include:

- **Background Intelligent Transfer Service (BITS) Server Extensions** Installs an extension that allows Web clients to use available bandwidth for data transfers and restart incomplete transfers.
- **Common Files** Installs common files required by IIS programs and documentation that covers server administration and publishing site content.
- **File Transfer Protocol (FTP) Service** Installs the FTP server service used to transfer files using FTP.
- **FrontPage 2002 Server Extensions** Installs extensions that allow Web site authoring and administration using Microsoft FrontPage and Microsoft Visual InterDev. If you elect to install these extensions, the Administration Web site isn't installed. IIS is configured so that you can manage servers and applications using FrontPage or the Microsoft SharePoint Team Services.
- **Internet Information Services (IIS) Manager** Installs the MMC snap-in for the IIS administration tools.

Note Throughout this book, we will refer to the Internet Information Services (IIS) Manager server component as the *IIS snap-in*.

- **Internet Printing** Installs extensions that allow Web-based printer management and printing to shared printers over the Internet, an extranet, or an intranet.
- **NNTP Service** Installs the Network News Transfer Protocol (NNTP) service used to create and manage newsgroups.
- **SMTP Service** Installs the Simple Mail Transfer Protocol (SMTP) service used for outgoing mail from a Web server.
- **World Wide Web Server** Installs the Web service used to publish and manage Web sites.

When you install Internet services, default sites are created on the computer. In most instances these default sites are active by default. If the default sites aren't active, you can start them using the IIS snap-in. To start the snap-in, click Start, choose All Programs, Administrative Tools, and then Internet Information Services (IIS) Manager. Default sites you see might include:

- **Default FTP Site** The default site for FTP services—which is installed only when you elect to install this option as part of the IIS installation. By default, anonymous connections are allowed access to FTP sites. Disable this service if you don't intend to use FTP for file transfers.
- **Default Web Site** The default site for Web services. By default, anonymous connections are allowed access to Web sites. Disable anonymous connections unless your site is ready to go public.
- **Administration Web Site** The default site for browser-based administration. By default, this site is only accessible from the local system. If you wish to use this service for remote administration, change the default IP filtering.

Note When the administration Web site is stopped, you can't manage sites using the Remote Administration tools. These tools are Web-based and depend on the administration Web site. This Web site isn't enabled by default. You must enable ASP as a valid Web Server Extension, as discussed in Chapter 3, "Configuring Web Sites and Servers," and also start the site. If you install the FrontPage Server Extensions on a Web server, you use the SharePoint tools for Web-based administration.

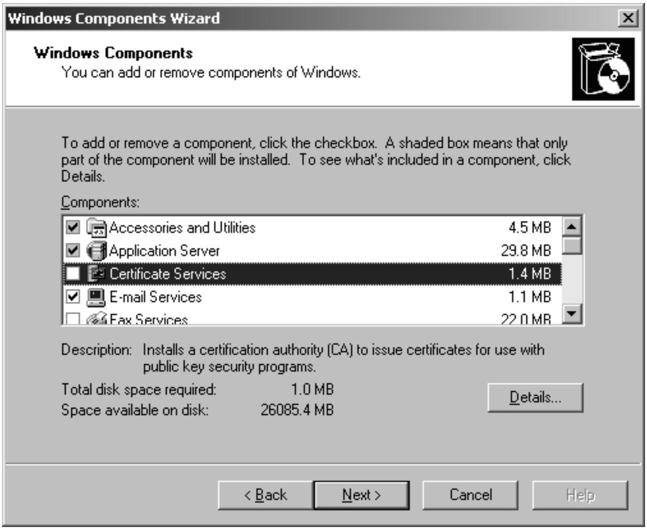
- **Default SMTP Virtual Server** The default site for SMTP services. If you don't use pages that generate e-mail messages, don't start SMTP services. By default, only servers that authenticate themselves in the domain can relay mail on the server. This denies permission to relay e-mail through the server and protects the server from being used to deliver unsolicited e-mail messages.
- **Default NNTP Virtual Server** The default site for NNTP services. The default configuration allows client posting and updates from news feeds and grants permission to other servers to pull articles from the server. If necessary, change these settings before starting an NNTP server.

If an IIS feature you want to use isn't available in the IIS snap-in, you can install it using the Windows Components Wizard. To access and use this wizard, follow these steps:

1. Log on to the computer using an account with administrator privileges.
2. Click Add Or Remove Programs in the Control Panel. This displays the Add Or Remove Programs dialog box.

Note Throughout this book, I refer to clicking or double-clicking, the most common techniques used for accessing folders and running programs. Through the Taskbar And Start Menu Properties dialog box, you can change the look and feel of the graphical interface. Some options, such as the Control Panel, can appear as menus with clickable menu items that run programs or as menu items that open dialog boxes. You can also change the mouse click options with the Folder Options utility in the Control Panel to allow either single-click open/run or double-click to open. Because of this, when I say click, you might actually have to double-click, or vice versa.

3. Click Add/Remove Windows Components to start the Windows Components Wizard, shown in Figure 1-1.



Windows Components

You can add or remove components of Windows.

To add or remove a component, click the checkbox. A shaded box means that only part of the component will be installed. To see what's included in a component, click Details.

Components:

	Component	Size
☑	Accessories and Utilities	4.5 MB
☑	Application Server	29.8 MB
☐	Certificate Services	1.4 MB
☑	E-mail Services	1.1 MB
☐	Fax Services	22.0 MB

Description: Installs a certification authority (CA) to issue certificates for use with public key security programs.

Total disk space required: 1.0 MB

Space available on disk: 26085.4 MB

Figure 1-1. *Use the Windows Component Wizard to select components to add or remove.*

4. Select Certificate Services, E-Mail Services, or Indexing Service as necessary.

5. Select Application Server. Click Details to add and remove individual components. You can now select subcomponents to install or uninstall them.
6. Select Internet Information Services (IIS). Click Details to add and remove individual components. You can now select subcomponents to install or uninstall them.
7. When ready to continue, click OK twice and then Next. The selected components are then installed (or uninstalled).
8. Click Finish when prompted.

Installing Internet Services and Service-Related Accounts

When you install Web and application server components, several services are installed on the computer. You can check for these services using the Services utility or Computer Management. Both utilities are found on the Administrative Tools menu. Services you might see include:

- **ASP.NET State Service** Provides support for out-of-process session states when using ASP.NET
- **Background Intelligent Transfer Service** Transfers files in the background using idle network bandwidth
- **Certificate Services** Provides services for creating, managing, and removing X.509 certificates
- **COM+ Event System** Provides system event notification services for COM components
- **COM+ System Application** Provides configuration and tracking for COM components
- **Cryptographic Services** Provides management services for certificate authorities
- **Distributed Transaction Coordinator** Coordinates transactions for Microsoft Distributed Transaction Coordinator (DTC)
- **FTP Publishing Service** Provides services for transferring files using FTP and also allows administration of an FTP server through the IIS snap-in
- **HTTP SSL** Enables SSL by providing the necessary services for Hypertext Transfer Protocol Secure (HTTPS)
- **IIS Admin Service** Allows administration of IIS through the IIS snap-in
- **Indexing Service** Indexes the contents and properties of files, providing quick access to files through a flexible query language
- **Message Queuing** Provides the necessary services for distributed messaging and message queuing
- **Microsoft POP3 Service** Provides POP3 service for mail transfer and retrieval

- **MSSQL$UDDI** Provides Web database services for the Microsoft Data Engine
- **MSSQLServerADHelper** Provides Active Directory helper services for the Microsoft Data Engine
- **Network News Transport Protocol (NNTP)** Provides network news services and allows administration of NNTP servers through the IIS snap-in
- **Simple Mail Transport Protocol (SMTP)** Provides mail transfer services and allows administration of SMTP sites through the IIS snap-in
- **SQLAgent$UDDI** Provides SQL Server Agent services for the Microsoft Data Engine
- **Web Element Manager** Provides access to user interface elements needed for the Remote Administration Web tools
- **World Wide Web Publishing Service** Provides services for transferring files using HTTP and also allows administration of an HTTP server

By default, most Web-related services run as the Local Service account. This allows the services to interact with the operating system. To tighten security, some services, such as the Microsoft POP3 service and the World Wide Web Publishing Service, run as the NetworkService account. This account has fewer privileges than the Local Service account.

Note You might find that the World Wide Web Publishing Service and other services normally running under the NetworkService account are running under the Local Service account on your system. This can happen if you install components, such as Certificate Services, that require more interaction with the operating system than a standard IIS installation.

When you install IIS, several accounts are created as well. These accounts are:

- **IIS_WPG** The IIS Worker Process Group account. All worker processes running under IIS use this group account. If this account is disabled or locked out, IIS won't function normally.
- **IUSR_*ComputerName*** The Internet guest account used by anonymous users to access Internet sites. If this account is disabled or locked out, anonymous users can't access Internet services. In a domain, this account is a member of the Domain Users and Guests groups. Otherwise, it's only a member of the Guests group.
- **IWAM_*ComputerName*** An account used by IIS to run out-of-process applications. If this account is disabled or locked out, out-of-process applications can't start. As all applications and sites configured under IIS 6 are technically out-of-process, IIS might not work properly if this account isn't available. In a domain, this account is a member of the Domain Users and IIS_WPG groups. Otherwise, it's only a member of the IIS_WPG groups.

Tip The IUSR and IWAM accounts have a password that never expires and can't be changed by users. You can, however, set and manage the password for these accounts as you would for any other account.

Other Web server and application components might cause additional accounts to be created, including the following:

- **ASPNET** An account used to run ASP.NET worker processes. This account is a member of the Domain Users group.
- **Cert Publishers** A group account that allows member users to publish X.509 public key certificates.

Web Administration Tools and Techniques

Web administrators will find that there are many ways to manage Web and application servers. The key administration tools and techniques are covered in the following sections.

Managing Resources with Key Administration Tools

Many tools are available for managing Web resources. Key tools you'll use are shown in Table 1-1. Most of these tools are available on the Administrative Tools menu. Click Start and choose All Programs, Administrative Tools, and then the tool you want to use. You can use all the tools listed in the table to manage local and remote resources. For example, you can connect to a new computer in the IIS snap-in and then, afterward, you can remotely manage all its sites and services from your system.

Table 1-1. Quick Reference for Key Web Administration Tools

Administration Tool	Purpose
Active Directory Users and Computers	Manages domain user, group, and computer accounts.
Certification Authority	Manages certificate services for public key X.509 certificates.
Computer Management	Manages services, storage, and applications. The Services And Applications node provides quick access to Indexing Service catalogs and IIS sites and servers.
Data Sources	Configures and manages Open Database Connectivity (ODBC) data sources and drivers. Data sources link Web front ends with database back ends.
DNS	Public Internet sites must have fully qualified domain names (FQDNs) to resolve properly in browsers. Use the Domain Name System (DNS) administrative snap-in to manage the DNS configuration of your Windows Server DNS servers.

Table 1-1. Quick Reference for Key Web Administration Tools

Administration Tool	Purpose
Event Viewer	Manages events and system logs.
Internet Information Services Snap-In	Manages Web and application server resources using a Microsoft Management Console (MMC) snap-in.
Remote Administration	Manages Web and application server resources using a browser-based interface. Formerly called Internet Services Manager.
Performance	Tracks system performance, pinpoints performance problems, and configures system event logs and alerts.
POP3 Service	Used to view and manage POP3 e-mail domains and mailboxes.
Services	Views service information; starts and stops system services; configures service logons and automated recoveries.

Installing Administration Tools

When you add services to a server, the tools needed to manage those services are automatically installed. If you want to manage these servers remotely, you might not have these tools installed on your workstation. In this case you need to install the administration tools on the workstation you're using.

To install the Windows administration tools, follow these steps:

1. Log on to the workstation using an account with administrator privileges.
2. Insert the Windows Server CD-ROM into the CD-ROM drive.
3. When the Autorun screen appears, click Perform Additional Tasks, and then click Browse This CD. This starts Windows Explorer.
4. Double-click the I386 folder, and then double-click Adminpak.msi. The complete set of Windows Server management tools is installed on your workstation or server.

Real World The Windows 2000 Administration tools are incompatible with Windows XP Professional and Windows Server 2003. If you upgrade to Windows XP Professional from Windows 2000 Professional, you'll find that many of the Windows 2000 administration tools won't work and you'll encounter errors frequently. You should uninstall these tools and instead install the Windows Server Administration Tools Pack (Adminpak.msi) on the Windows XP Professional systems that administrators use. The Windows Server administration tools are compatible with both Windows 2000 and Windows Server 2003.

While you're working with the distribution CD-ROM, you might want to install the Windows Support Tools. The support tools are a collection of utilities for handling everything from system diagnostics to network monitoring. You can install the support tools by completing the following steps:

1. Insert the Windows Server CD-ROM into the CD-ROM drive.
2. When the Autorun screen appears, click Perform Additional Tasks, and then click Browse This CD. This starts Windows Explorer.
3. In Windows Explorer, double-click Support and then double-click Tools.
4. Double-click Suptools.msi. This starts the Windows Support Tools Setup Wizard. Click Next.
5. Read the End User License Agreement, and then, if you agree and want to continue, click I Agree and then click Next.
6. Enter your user information and then click Next.
7. Select the destination directory for the support tools. The default location is %ProgramFiles%\System Tools. If you don't want to use the default location, type a new directory path or click Browse to search for a location. The tools use about 23 MB of disk space.
8. Click Install Now.

Note %ProgramFiles% refers to the ProgramFiles environment variable. The Windows operating system has many environment variables, which are used to refer to user-specific and system-specific values. I'll often refer to environment variables using this syntax: *%Variable Name%*.

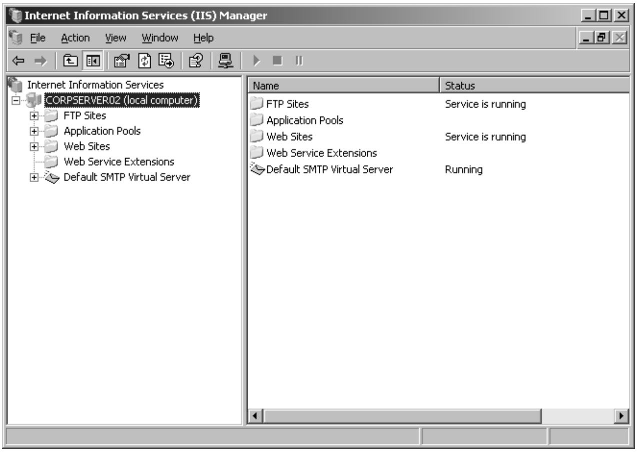
Web Administration Techniques

Web administrators have many options for managing IIS. The key administration tools are:

- Internet Information Services snap-in
- Remote Administration (formerly Internet Services Manager)
- IIS Administration objects (which are manipulated by the administration scripts)
- Administration scripts

The IIS snap-in provides the standard administration interface for IIS. Figure 1-2 shows the main window for the IIS snap-in. To start the IIS snap-in, click Start and choose All Programs, Administrative Tools, and then Internet Information Services (IIS) Manager.

When started, the IIS snap-in automatically connects to the local IIS installation, if it's available. Once you connect to remote IIS installations, the IIS snap-in automatically connects to these installations upon startup as well. You can change this behavior by disconnecting from the remote server while in the snap-in. See Chapter 3 for more information on using the IIS snap-in.



Internet Information Services (IIS) Manager
File
Action
View
Window
Help
Internet Information Services
CORPSERVER02 (local computer)
FTP Sites
Application Pools
Web Sites
Web Service Extensions
Default SMTP Virtual Server

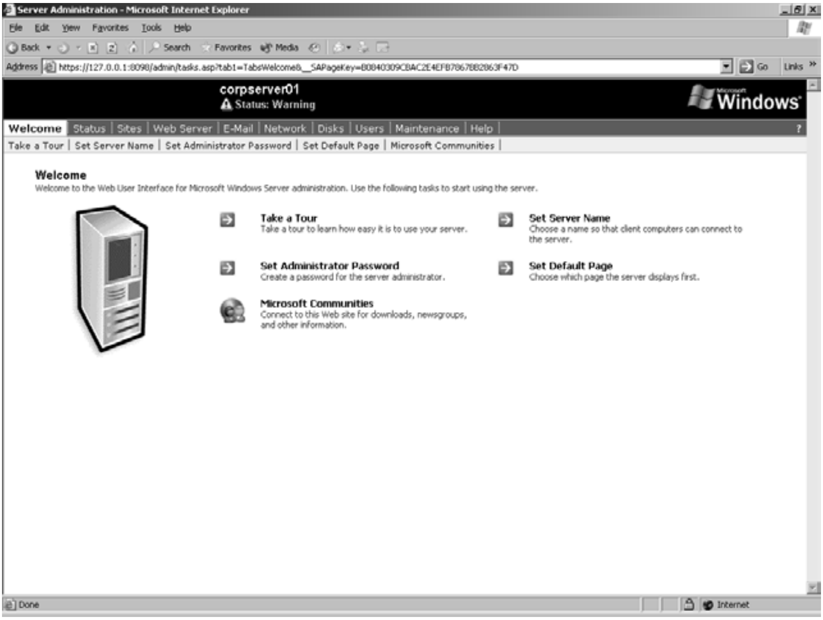
Name	Status
FTP Sites	Service is running
Application Pools	
Web Sites	Service is running
Web Service Extensions	
Default SMTP Virtual Server	Running

Figure 1-2. *Use the IIS snap-in to manage local and remote IIS installations.*

Remote Administration uses the administration Web site to access remote IIS installations using a secure HTTP connection. You can allow or disallow remote browser-based administration by starting or stopping the Administration Web site. When installed, IIS randomly selects two Transmission Control Protocol (TCP) port numbers from 2000 to 9999 and assigns these port numbers to the administration Web site. One TCP port is used for standard, not secure, connections and the other for secure connections to the administration Web site. The default configuration is to allow only secure connections for administration.

The site responds to browser requests for all permitted domains, but the administrator must specify the port number because it differs from the default HTTPS port 443. For example, if the server's domain name is primary.microsoft.com and the administrative port is 9394, you can connect to the administration Web site by typing the following URL into your browser window: *https://primary.microsoft.com:9394/.*

Figure 1-3 shows the main window for Remote Administration. By default, basic authentication is configured. Since you're required to use a secure connection (HTTPS), this setting is adequate, but you might want to consider using another authentication technique. With basic authentication, you're prompted for a user name and password when the site is accessed. If you provide the proper logon information and are a member of the Windows Administrators group, you'll be permitted to administer IIS remotely through the administration Web site.


Server Administration - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address https://127.0.0.1:8098/admin/tasks.asp?tab1=TabsWelcome&__SAPageKey=B0840309CBAC2E4EFB7867BB2063F47D
Go Links
Windows
Welcome Status Sites Web Server E-Mail Network Disks Users Maintenance Help
Take a Tour | Set Server Name | Set Administrator Password | Set Default Page | Microsoft Communities
Done
Internet

corpserver01
Status: Warning

Welcome
Welcome to the Web User Interface for Microsoft Windows Server administration. Use the following tasks to start using the server.

Take a Tour
Take a tour to learn how easy it is to use your server.

Set Server Name
Choose a name so that client computers can connect to the server.

Set Administrator Password
Create a password for the server administrator.

Set Default Page
Choose which page the server displays first.

Microsoft Communities
Connect to this Web site for downloads, newsgroups, and other information.

Figure 1-3. *Use Remote Administration to manage remote IIS installations.*

In previous versions of IIS, you could designate Web site operators who could remotely administer IIS. Web site operators were a special group of users who had elevated privileges on individual Web sites. IIS 6 doesn't allow you to designate operators. But that's okay because most of the time operator accounts weren't used.

You can also manage IIS settings and configuration through Windows Script Host (WSH)—and you'll be happy to know that you no longer need to use Active Directory Service Interface (ADSI) to manage the metabase. Because the metabase is now formatted as XML and stored in plaintext files, you can say goodbye to complex key paths and all that other stuff that went along with it!

The IIS Windows scripts are stored in the %SystemRoot%\System32 directory. Table 1-2 provides an overview of each of the scripts. These scripts are all written using VBScript.

Table 1-2. Quick Reference for Key IIS Administration Scripts

Administration Script	Purpose
Iisapp.vbs	Reports process IDs and application pool IDs for currently running worker processes. These processes run as W3wp.exe.
Iisback.vbs	Allows you to back up or restore the IIS configuration. You can also list backups or delete individual backups.
Iiscnfg.vbs	Imports, exports, or copies the IIS configuration.
Iisext.vbs	Configures the Web Server Extensions.
Iisftp.vbs	Manages and lists available FTP sites.
Iisftpdr.vbs	Manages and lists virtual directories for an FTP site under a given root.
Iisvdir.vbs	Manages and lists virtual directories for a Web site under a given root.
Iisweb.vbs	Manages and lists available Web sites.

The scripts are designed to work with the command-line Windows Script Host, Cscript.exe. This host must be registered as the default scripting host on the computer you're using to execute the scripts. You can ensure that Cscript.exe is registered as the default host by entering the following command in a command prompt:

```
cscript //H:cscript
```

Because the IIS scripts are stored in the %SystemRoot%\System32 directory, you can run a script from any directory on the server by typing the script name on the command-line, such as:

```
iisweb /query
```

Type the script name followed by /? on the command-line to display basic help information.

Chapter 2

Core IIS Administration

Core Internet Information Services (IIS) administration tasks revolve around connecting to servers, managing services, and saving metabase configurations. In IIS you connect to individual servers and manage their IIS components through the IIS snap-in, the Application Server snap-in, or the Remote Administration tool. You can use a single IIS server to host multiple resources. Web and File Transfer Protocol (FTP) resources are referred to as Web sites and FTP sites, respectively. Simple Mail Transfer Protocol (SMTP) and Network News Transfer Protocol (NNTP) resources are referred to as SMTP virtual servers and NNTP virtual servers, respectively.

Sites and virtual servers are server processes that have their own configuration information, which can include Internet Protocol (IP) addresses, port numbers, and authentication settings. To perform most administration tasks with sites and servers, you'll need to log in to the IIS server using an account that has administrator privileges. You can find detailed information on security in Chapter 7, "Enhancing Web Server Security."

Understanding the IIS Architecture

Most administrators don't understand the actual underpinnings of IIS. Yet to really understand how IIS works, you have to understand the architecture. You can think of IIS as a layer over the operating system where, in most cases, you might need to perform an operating system level task before you perform an IIS task. This is true in several key areas:

- **Directories** Sites, virtual servers, and other resources use the Microsoft Windows Server 2003 file and directory structure. Before you create IIS resources, such as sites or virtual servers, you should ensure that any necessary directories have been created.
- **Permissions** Windows Server 2003 permissions ultimately determine whether users can access files and directories. Before users can access files and directories, you must ensure that the appropriate users and groups have access at the operating system level. After you set operating system (OS)–level permissions, you must set IIS-specific security permissions.

Windows services and processes are other areas where Windows Server 2003 and IIS are tightly integrated. IIS has two operating modes that affect services and processes. These operating modes are:

- **IIS 5 isolation mode** The standard processing mode of IIS 5
- **Worker Process isolation mode** The default processing mode of IIS 6.0 on a clean install

The IIS 5 isolation mode and worker process isolation mode are mutually exclusive. The World Wide Web Service can operate only in one mode or the other, which means that all Web sites configured on a server use the same operating mode.

The sections that follow examine each operating mode, providing a discussion of how, why, and when the modes are used, as well as providing details of what the components of each mode are. IIS application and application pools are discussed in detail in Chapter 5, "Running IIS Applications," and Chapter 6, "Managing ASP.NET, Application Pools, and Worker Processes."

Understanding and Using IIS 5 Isolation Mode

You use IIS 5 isolation mode to run Web applications that were developed for older versions of IIS. Using this operating mode affects how IIS is used and how IIS interacts with other components.

IIS 5 Isolation Mode Overview

IIS 5 isolation mode operates nearly the same as the standard mode of IIS 5 depicted in Figure 2-1. Service Host processes control all resources of the same type running on a server. Because of this, Windows Server 2003 uses the Service Host to manage all instances of a specific resource, such as Web or FTP sites, running on a server. For example, if you start or stop the World Wide Web Publishing Service, you're controlling all Web sites running on the server through the related Service Host process.

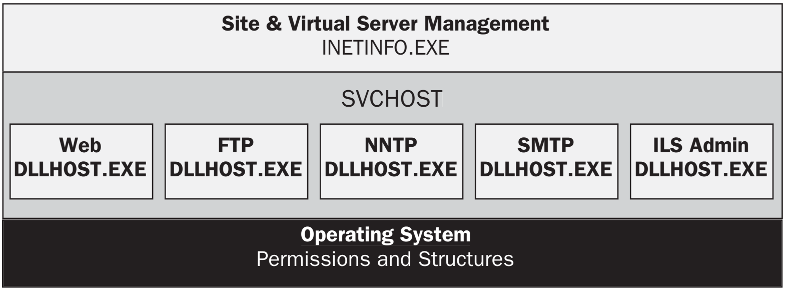


Figure 2-1. *Here is a conceptual view of the IIS 5 isolation mode.*

Because of the layered structure of IIS, starting or stopping an Internet Information Service doesn't directly affect the Service Host. Instead, Windows Server 2003 uses an intermediary to control the Service Host for you. This intermediary is the

InetInfo process. A single instance of Inetinfo.exe is used to manage the Service Hosts as well as Internet Server Application Programming Interface (ISAPI) applications that run within the IIS process context. When you control IIS individually, Windows Server 2003 controls the Service Host through InetInfo. InetInfo also makes it possible to manage all IIS resources running on a server. You can, for example, issue a restart command in the IIS snap-in that restarts IIS completely. See the section entitled "Starting, Stopping, and Restarting All Internet Services," later in this chapter, for more details.

ISAPI applications are a key part of the IIS 5 architecture. ISAPI applications are server-based applications that run on IIS Web sites. As Figure 2-2 shows, you use DLL Host (Dllhost.exe) to manage out-of-process ISAPI applications. Any pooled ISAPI applications running on the server run within the context of a single instance of Dllhost.exe. In contrast, isolated ISAPI applications run within the context of separate DLL Host processes.

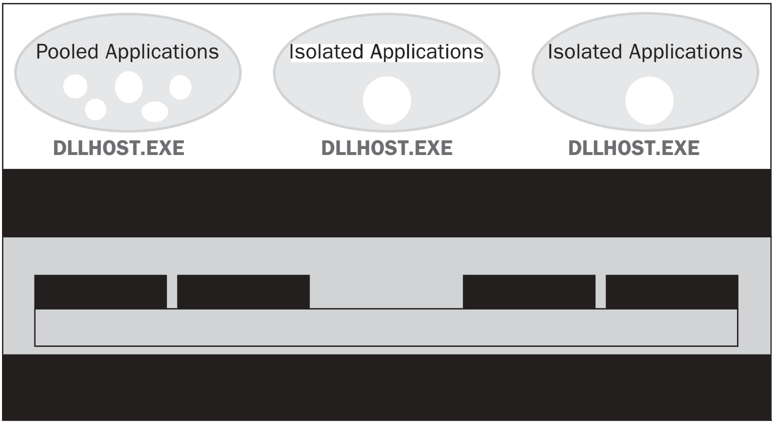


Figure 2-2. *Use DLL Host (Dllhost.exe) to manage out-of-process ISAPI applications.*

Understanding Application Incompatibilities and Consequences

You must use IIS 5 isolation mode for Web applications that aren't compatible with the IIS 6 worker process model. Characteristics that might make an application written for IIS 5 incompatible with worker process mode include:

- **Session states that are managed in-process** You can configure IIS 6 to recycle worker processes periodically, on demand, or when a specific criterion is met. When a worker process is recycled, session state data might be lost.

- **Code that sends out requests to other worker processes** IIS 6 worker processes can't communicate or send out requests to other processes. Worker processes are completely isolated to prevent applications or sites in one application pool from stopping applications or sites in another application pool.
- **Components that don't support loading by multiple processes** Multiple IIS 6 worker processes might load and run ISAPI and COM components concurrently. If concurrent instances of an ISAPI or COM component can't run simultaneously, the components are incompatible for worker process isolation mode.

If you have incompatible applications, you have several choices:

- Switch to IIS 5 isolation mode, forcing all applications to run in this mode and losing all the benefits of IIS 6 worker process isolation mode. To switch to IIS 5 isolation mode in IIS Manager, right-click Web Sites and then click Properties. In the Service tab, select Run WWW Service In IIS 5 isolation mode, and then click OK. Afterward, when prompted to restart the World Wide Web Service, click Yes. The reconfiguration process can take several minutes, so be patient.
- Configure separate Web application servers, each running in a different mode. Run IIS 5–compatible applications on servers running IIS 5 isolation mode. Run IIS 6–compatible applications on servers running worker process isolation mode.
- Migrate incompatible applications to IIS 6 architecture. If you do this, be sure to look at these server support functions: CustomError (to use IIS custom errors), ExecuteUrl (to replace read raw data filters), ReportUnhealthy (to force recycle unstable or questionable process), and VectorSend (to manage multiple buffer and file handles). The IIS 6 architecture also supports Unicode Uniform Resource Locators (URLs), COM+ partitions, dynamic-link library (DLL) runtime versioning (fusion), and poolable objects using the multithreaded apartment model.

Switching to IIS 5 mode also affects the way ASP.NET is used on the server. In IIS 5 isolation mode, ASP.NET uses its own processing model. This processing model is similar to worker process isolation mode and has similar capabilities. Process model configurations for ASP.NET applications are taken from the Windows .NET Framework XML file, which is called Machine.config.

In contrast, under the default configuration, ASP.NET and IIS are directly integrated. ASP.NET uses the worker process model architecture of IIS 6. ASP.NET applications can take advantage of IIS 6 features, and applications are configured through the application pool settings. The only exception is that if you've configured maximum input/output (I/O) threads or maximum worker threads in a Machine.config file, these settings will still be read and used. All other configuration settings in the Machine.config file are ignored.

Switching to IIS 5 Isolation Mode

To switch to IIS 5 isolation mode, follow these steps:

1. Expand the Internet Information Services node in the IIS or the Application Server snap-in.

Note If the server you want to work with isn't listed, right-click Internet Information Services, select Connect, and then type the server name or click Browse to find a server. If necessary, select Connect As and provide your logon credentials for the remote server. Click OK.

2. Expand the server node. Right-click Web Sites and then click Properties.
3. In the Service tab, select Run WWW Service In IIS 5 Isolation Mode, and then click OK.
4. When prompted to restart the World Wide Web Service, click Yes. Windows Server 2003 then reconfigures processing and restarts the Web service. This process can take several minutes, so be patient.

Understanding and Using Worker Process Isolation Mode

Worker process isolation mode is the default mode of IIS. This mode allows sites and applications to:

- Recycle worker threads
- Monitor process health
- Use advanced application pooling configurations
- Take advantage of other IIS 6 features

From a high level, worker process isolation mode is similar to IIS 5 isolation mode. Service Host processes control all resources of the same type running on a server. Starting, pausing, or stopping a service affects all sites of the same type on the server. It doesn't directly affect the Service Host. Instead, Windows Server 2003 uses an intermediary to control the Service Host for you. For non-Web services, this intermediary is the InetInfo process. A single instance of Inetinfo.exe is used to manage the FTP, SMTP, and NNTP Service Hosts.

Management of the Web service and Web applications is internalized. The Web Administration Service component of the Web Service Host is used to manage the service itself. Worker processes are used to control applications, and no ISAPI applications run within the IIS process context.

Worker processes are used in several ways:

- **Single worker process—single application** Here, a single worker process running in its own context (isolated) handles requests for a single application, as well as instances of any ISAPI extensions, and filters the application needs. The application is the only one assigned to the related application pool.

- **Single worker process—multiple applications** Here, a single worker process running in its own context (isolated) handles requests for multiple applications assigned to the same application pool, as well as instances of any ISAPI extensions, and filters the application needs.
- **Multiple worker processes—single application** Here, multiple worker processes running in their own context (isolated) share responsibility for handling requests for a single application, as well as instances of any ISAPI extensions, and filter the application needs. The application is the only one in the related application pool.
- **Multiple worker processes—multiple applications** Here, multiple worker processes running in their own context (isolated) share responsibility for handling requests for multiple applications assigned to the same application pool, as well as instances of any ISAPI extensions, and filter the application needs.

Benefits of Using Worker Processing Mode

Running IIS in worker processing mode has many benefits. In this mode, all sites run within an application context and have an associated application pool. The default application pool is DefaultAppPool. You can also assign sites and applications to custom application pools.

Each application or site in an application pool can have one or more worker processes associated with it. The worker processes handle requests for the site or application.

You can configure application pools to manage worker processes in many ways. You can configure automatic recycling of worker threads based on a set of criteria, such as when the process has been running for a certain amount of time or uses a specific amount of memory. You can also have IIS monitor the health of worker threads and take actions to recover automatically from failure. These features might eliminate or reduce your dependence on third-party monitoring tools or services.

In worker processing mode, you can also create a Web garden where you configure multiple worker processes to handle the workload. Applications configured using this technique are more responsive, more scalable, and less prone to failure. Why? A Hypertext Transfer Protocol (HTTP) listener, called Http.sys, listens for incoming requests and places them in the appropriate application pool request queue. When a request is placed in the queue, an available worker process assigned to the application can take the request and begin processing it. Idle worker processes handle requests in first in, first out (FIFO) order.

Worker processes can also be started on demand. If there are unallocated worker processes and no current idle worker processes, IIS can start a new worker process to handle the request. In this way, resources aren't allocated until they're needed, and IIS can handle many more sites than it could if all processes were allocated on startup.

Switching to Worker Processing Mode

To switch to worker processing mode, follow these steps:

1. Expand the Internet Information Services node in the IIS or the Application Server snap-in.

Note If the server you want to work with isn't listed, right-click Internet Information Services, select Connect, and then type the server name or click Browse to find a server. If necessary, select Connect As and provide your logon credentials for the remote server. Click OK.

2. Expand the server node. Right-click Web Sites and then click Properties.
3. In the Service tab, clear Run WWW Service In IIS 5 Isolation Mode, and then click OK.
4. When prompted to restart the World Wide Web Service, click Yes. Windows Server 2003 then reconfigures processing and restarts the Web service. When the process is finished, you'll have an Application Pools node that you can use to manage the default application pool and any other pools you create on the server.

Working with IIS and URLs

To retrieve files from IIS servers, clients must know three things: the server's address, where on the server the file is located, and which protocol to use to access and retrieve the file. Normally, this information is specified as a URL. URLs provide a uniform way of identifying resources that are available using IPs. The basic mechanism that makes URLs so versatile is their standard naming scheme.

URL schemes name the protocol the client will use to access and transfer the file. Clients use the name of the protocol to determine the format for the information that follows the protocol name. The protocol name is generally followed by a colon and two forward slashes. The information after the double slash marks follows a format that depends on the protocol type referenced in the URL. Here are two general formats:

protocol://hostname:port/path_to_resource

protocol://username:password@hostname:port/path_to_resource

Host name information used in URLs identifies the address to a host and is broken down into two or more parts separated by periods. The periods are used to separate domain information from the host name. Common domain names for Web servers begin with *www*, such as *www.microsoft.com*, which identifies the Microsoft WWW server in the commercial domain. Domains you can specify in your URLs include:

- **com** Commercial sites
- **edu** Education sites

- **gov** Nonmilitary government sites
- **mil** Military sites
- **net** Network sites
- **org** Organizational sites

Port information used in URLs identifies the port number to be used for the connection. Generally, you don't have to specify port numbers in your URLs unless the connection will be made to a port other than the default. As shown in Table 2-1, port 80 is the default port for HTTP. If you request a URL on a server using the URL *http://www.microsoft.com/docs/my-yoyo.htm,* port 80 is assumed to be the default port value. On the other hand, if you wanted to make a connection to port 8080, you'd need to type in the port value, such as *http://www.microsoft.com:8080/docs/my-yoyo.htm.*

Port values that fall between zero and 1023, referred to as *well-known ports*, are reserved for specific data type uses on the Internet. Port values between 1024 and 49151 are considered registered ports, and those between 49152 and 65535 are considered dynamic ports.

Table 2-1. Default Ports for IIS Resources

Protocol	Default Port
FTP	21
SMTP	25
HTTP	80
NNTP	119

The final part of a URL is the path to the resource. This path generally follows the directory structure from the server's home directory to the resource specified in the URL.

URLs for FTP can also contain a user name and password. User name and password information allow users to log in to an FTP server using a specific user account. For example, the following URL establishes a connection to the Microsoft FTP server and logs on using a named account: *ftp://sysadmin:rad$4@ftp.microsoft.com/public/download.doc.*

In this instance, the account logon is *sysadmin*, the password is *rad$4*, the server is *ftp.microsoft.com*, and the requested resource is *public/download.doc.*

If a connection is made to an FTP server without specifying the user name and password, you can configure the server to assume that the user wants to establish an anonymous session. In this case the following default values are assumed: *anonymous* for user name and the user's e-mail address as the password.

URLs can use uppercase and lowercase letters, the numerals 0-9, and a few special characters, including:

- Asterisks (*)
- Dollar signs ($)
- Exclamation points (!)

- Hyphens (-)
- Parentheses (left and right)
- Periods (.)
- Plus signs (+)
- Single quotation marks (‘)
- Underscores (_)

You’re limited to these characters because other characters used in URLs have specific meanings, as shown in Table 2-2.

Table 2-2. Reserved Characters in URLs

Character	Meaning
:	The colon is a separator that separates protocol from the rest of the URL scheme; separates host name from the port number; and separates user name from the password.
//	The double slash marks indicate that the protocol uses the format defined by the Common Internet Scheme Syntax (see RFC 1738 for more information).
/	The slash is a separator and is used to separate the path from host name and port. The slash is also used to denote the directory path to the resource named in the URL.
~	The tilde is generally used at the beginning of the path to indicate that the resource is in the specified user’s public Hypertext Markup Language (HTML) directory.
%	Identifies an escape code. Escape codes are used to specify special characters in URLs that otherwise have a special meaning or aren’t allowed.
@	The at symbol is used to separate user name and/or password information from the host name in the URL.
?	The question mark is used in the URL path to specify the beginning of a query string. Query strings are passed to Common Gateway Interface (CGI) scripts. All the information following the question mark is data the user submitted and isn’t interpreted as part of the file path.
+	The plus sign is used in query strings as a placeholder between words. Instead of using spaces to separate words that the user has entered in the query, the browser substitutes the plus sign.
=	The equal sign is used in query strings to separate the key assigned by the publisher from the value entered by the user.
&	The ampersand is used in query strings to separate multiple sets of keys and values.
^	The caret is reserved for future use.
{}	Braces are reserved for future use.
[]	Brackets are reserved for future use.

To make URLs even more versatile, you can use escape codes to specify characters in URLs that are either reserved or otherwise not allowed. Escape codes have two components: a percent sign and a numeric value. The percent sign identifies the start of an escape code. The number following the percent sign identifies the character being escaped. The escape code for a space is a percent sign followed by the number 20 (%20). You could use this escape code in a URL such as this one:

http://www.microsoft.com/docs/my%20party%20hat.htm

IIS and Application Server Snap-In Essentials

The IIS snap-in is a Microsoft Management Console (MMC) snap-in for managing IIS resources in Windows domains. You'll use this tool to perform administration routine tasks, such as starting Internet services, starting individual sites, and rebooting servers remotely.

If you work with ASP.NET, Windows .NET Framework, or Component Services, you can also use the Application Server snap-in. Application Server has three primary nodes:

- **.NET Configuration** Configure and manage Windows .NET Framework assemblies, assembly caches, remoting services, runtime security, and ASP.NET applications
- **Internet Information Services (IIS) Manager** Configure and manage Web, FTP, SMTP, and NNTP services
- **Component Services** Configure and manage COM components and COM+ applications

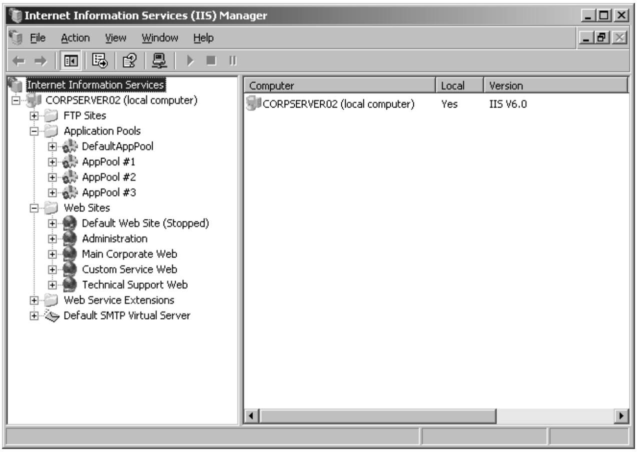
Other than the fact that there are additional administration nodes, you manage IIS in the Application Services snap-in using the same techniques as those with the IIS snap-in. Because of this, I won't provide a separate discussion on using the Application Services snap-in. If you prefer the Application Services snap-in, start from this tool rather than the IIS snap-in, as specified.

Note The Remote Administration tool provides a browser-based interface for managing Web and FTP resources. This tool has many of the same features as the IIS snap-in. For details on starting and using this tool, see the section entitled "Web Administration Techniques" in Chapter 1, "Overview of Microsoft Web Services."

Starting and Using the Internet Information Services (IIS) Manager Snap-In

The IIS snap-in is accessible in several locations. You can access the snap-in through a preconfigured console by clicking Start and choosing All Programs, Administrative Tools, and then Internet Information Services (IIS) Manager. Or you can access the snap-in through Manage Your Server, which can also be started from the Administrative Tools menu. Once you start Manage Your Server, click Open The Web Interface For Remote Administration Of Web Servers.

Figure 2-3 shows the main window for the IIS snap-in. The snap-in automatically connects to local IIS installations (if available). You can connect to one or more remote computers as well. Each additional computer to which you connect has a separate node that you can use to manage its resources.



Computer	Local	Version
CORPSERVER02 (local computer)	Yes	IIS V6.0

Figure 2-3. *Use the IIS snap-in to manage Web, FTP, SMTP, and NNTP resources.*

When you select the Internet Information Services node in the left pane, the right pane displays a summary of current computer connections. The connection summary provides:

- **Computer** Name of the computer to which you're connected.
- **Local** States whether you're connected to a local IIS installation. If the field value is set to Yes, you're connected to a local IIS installation. Otherwise, you're connected to a remote installation.
- **Version** Version of IIS installed on the computer.
- **Status** Status of the computer, such as unavailable or restarting.

If you expand the computer node, you'll find individual nodes for each service configured, application pools (if running in worker processing mode) and Web Service Extensions. When you select Web Sites or FTP Sites under a computer node in the left pane, the right pane displays an overview of these resources on the computer. The resource overview provides:

- **Description** Basic description of site or virtual server assigned through the Properties dialog box.
- **Identifier** Unique numeric value associated with the site.
- **State** Status of the site or virtual server, such as running, stopped, paused or unknown.
- **Host Header Value** Host name passed in the HTTP header to clients (if applicable).

- **IP Address** IP address of the site or virtual server. Incoming IP traffic is mapped by port and IP address to a specific site or virtual server instance. The value All Unassigned allows the HTTP, FTP, SMTP, or NNTP protocol to respond on all unassigned IP addresses that are configured on the server.
- **Port** Port number the site or virtual server listens on. Default ports for FTP and HTTP are 21 and 80, respectively.
- **SSL Port** Secure port number the site or virtual server listens on. Default port for HTTP is 443.
- **Status** Additional status information for the site or virtual server.

When first accessed, the IIS snap-in automatically connects to local IIS installations (if available). You can connect to other computers. If you do this, each computer will have its own node.

Connecting to Other Servers

Most of the time you'll manage IIS installations from your desktop system. When you do this, you'll need to establish a remote connection to the server you want to manage. The steps for establishing remote connections are:

1. Start the IIS snap-in.
2. In the left pane, right-click Internet Information Services and then select Connect. The Connect To Computer dialog box is displayed.
3. In the Computer Name field, type the name of the computer to which you want to connect. You can also type the server's IP address or fully qualified domain name (FQDN). Click Browse to search for a computer.
4. If you need to authenticate yourself on the computer, select Connect As and then provide the user name and password for an account with the appropriate privileges.
5. Click OK.

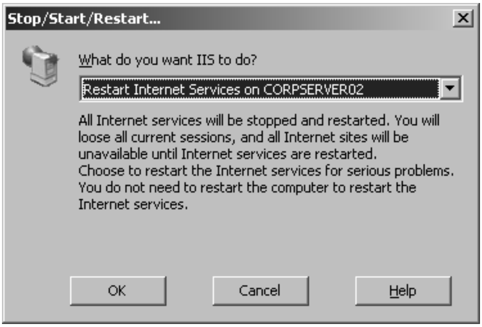
Real World Firewalls and proxy servers might affect your ability to connect to systems at remote locations. If you need to connect regularly to servers through firewalls or proxies, you'll need to consider the administration techniques you might want to use and then consult your company's network or security administrator to determine what steps need to be taken to allow those administration techniques. Typically, the network/security administrator will have to open TCP or UDP ports to allow remote communications between your computer or network and the remote computer or network. Each type of tool you want to use might require you to open different ports. For example, if you want to remotely administer a Web site using the Web tool, you'll need to open the standard and secure TCP port for the administration Web site. However, corporate policy might not allow the administrator to perform these tasks or might require prior approval from an information technology (IT) manager.

Starting, Stopping, and Restarting All Internet Services

As discussed earlier in the chapter, Window Server 2003 uses the Inetinfo.exe process to manage all Internet Information Services. InetInfo is able to do this because it tracks all IIS resources running on a computer and can issue commands to these resources. As an administrator, you can control InetInfo through the IIS snap-in or the Iisreset.exe command-line utility.

If you want to start, stop, or restart all of your Internet services from within the IIS snap-in, follow these steps:

1. In the IIS snap-in, select the icon for the computer you want to work with. If the computer isn't shown, connect to it, as discussed in the section of this chapter entitled "Connecting to Other Servers," and then select it.
2. Choose All Tasks from the Action menu and then select Restart IIS. This displays the Stop/Start/Restart dialog box shown in Figure 2-4.



Restart Internet Services on CORPSERVER02

All Internet services will be stopped and restarted. You will loose all current sessions, and all Internet sites will be unavailable until Internet services are restarted.
Choose to restart the Internet services for serious problems. You do not need to restart the computer to restart the Internet services.

Figure 2-4. *Stop, start, and restart all Internet Services.*

3. Use the drop-down menu to perform the following tasks:
   - Start Internet Services on *computername.*
   - Stop Internet Services on *computername*.
   - Restart *computername*.
   - Restart Internet Services on *computername*.
4. Click OK.

With the Restart Internet Services command, the sequence of tasks is important to understand. This command performs the following tasks:

1. Stops all Internet Information Services running on the computer, including World Wide Web Publishing Service, FTP Publishing Service, Network News Transport Protocol Service, Simple Mail Transport Protocol Service, and IIS Admin Service.

2. Attempts to resolve potential problems with runaway processes or hung applications by stopping all Dr. Watson (Drwtsn32.exe), MTX (Mtx.exe), and DLL Host (Dllhost.exe) processes.
3. Starts all Internet Information Services and then starts DLL Hosts as necessary.

You can also use the Iisreset.exe command-line utility to start, stop, and restart Internet Services. To start any Internet Information Services that are stopped on the local computer, type the following command:

```
iisreset /start
```

To stop all Internet Information Services that are running, paused, or in an unknown state on the local computer, type the following command:

```
iisreset /stop
```

To stop and then restart Internet Information Services on the local computer, type the following command:

```
iisreset /restart
```

You can also control Internet Information Services on remote computers. To do this, use the following syntax:

```
iisreset computername command
```

such as:

```
iisreset engsvr01 /restart
```

Table 2-3 provides a listing of all switches for the Iisreset.exe command-line utility. Rebooting computers is covered in the section of this chapter entitled "Rebooting IIS Servers."

Table 2-3. IISRESET Switches Defined

Switch	Definition
/DISABLE	Disable restarting of Internet Services on the local system.
/ENABLE	Enable restarting of Internet Services on the local system.
/NOFORCE	Don't forcefully terminate Internet services if attempting to stop them gracefully fails.
/REBOOT	Reboot the local or designated remote computer.
/REBOOTONERROR	Reboot the computer if an error occurs when starting, stopping, or restarting Internet services.
/RESTART	Stop and then restart all Internet services. Attempt to resolve potential problems with runaway processes or hung applications.
/START	Start all Internet services that are stopped.
/STATUS	Display the status of all Internet services.

Table 2-3. IISRESET Switches Defined

Switch	Definition
/STOP	Stop all Internet services that are running, paused, or in an unknown state.
/TIMEOUT:*val*	Specify the time-out value (in seconds) to wait for a successful stop of Internet services. On expiration of this time-out, the computer can be rebooted if the /REBOOTONERROR parameter is specified. With /STOP and /RESTART, an error is issued. The default value is 20 seconds for restart, 60 seconds for stop, and 0 seconds for reboot.

Starting, Stopping, and Pausing Individual Resources

Sites and virtual servers that use the same Internet Service can be controlled individually or as a group. You can control individual sites and virtual servers much like you do other server resources. For example, if you're changing the configuration of a site or performing other maintenance tasks, you might need to stop the site, make the changes, and then restart it. When a site is stopped, the site doesn't accept connections from users and can't be used.

An alternative to stopping a site or virtual server is to pause it. Pausing a resource prevents new client connections but doesn't disconnect current connections. When you pause a site or virtual server, active clients can continue to retrieve documents, work with messages, and perform other tasks. No new connections are accepted, however.

You can start, stop, or pause a site or virtual server by completing the following steps:

1. Start the IIS snap-in.
2. In the left pane, select the icon for the computer you want to work with. If the computer isn't shown, connect to it as discussed in the section of this chapter entitled "Connecting to Other Servers," and then select it.
3. Select FTP Sites or Web Sites as necessary, and then right-click the site or virtual server you want to manage. You can now:
    - Select Start to start the site or virtual server.
    - Select Stop to stop the site or virtual server.
    - Select Pause to pause the site or virtual server. After you pause a site or virtual server, click Pause again when you want to resume normal operations.

Note Groups of sites or virtual servers running under the same Internet Service are controlled through their master process. For example, the master process for all Web sites running on a computer is the World Wide Web Publishing service. Stopping this service stops all Web sites using the process and all connections to these sites are disconnected immediately. Starting this service restarts all Web sites that were running when the World Wide Web Publishing service was stopped. To learn how to control Internet Services, see the section of this chapter entitled "Managing IIS Services."

Rebooting IIS Servers

The IIS snap-in and Iisreset.exe utility have extensions that allow you to reboot local and remote computers. In order to use these extensions, you must have installed IIS on the computer and you must be a member of a group that has the appropriate user rights. To reboot a local system, you must have the right to shut down the system. To reboot a remote system, you must have the right to force shutdown from a remote system. You should only reboot an IIS server if the restart IIS procedure fails.

You reboot an IIS server with the snap-in by completing the following tasks:

1. In the IIS snap-in, select the icon for the computer you want to work with. If the computer isn't shown, connect to it as discussed in the section of this chapter entitled "Connecting to Other Servers," and then select it.
2. Click Action and then select Restart IIS. This displays the Stop/Start/Restart dialog box shown previously in Figure 2-4.
3. Select Restart *computername* on the drop-down list and then click OK.
4. A system shutdown message is sent to the target computer. This message explains that the computer is being shut down in 5 minutes. After completing the shutdown process, the system will reboot.

To reboot a computer using Iisreset.exe, type the following command:

```
iisreset computername /reboot
```

such as the following example:

```
iisreset engsvr01 /reboot
```

If users are working on files or performing other tasks that need to be exited gracefully, you should set a time-out value for services and processes to be stopped. By default the time-out is zero seconds, which forces immediate shutdown and tells Windows Server 2003 not to wait for services to be shut down gracefully. You could set a time-out value of 60 seconds when rebooting engsvr01 as follows:

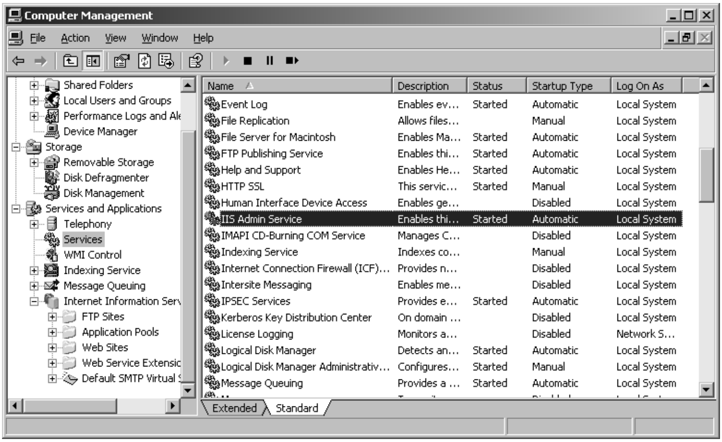
```
iisreset engsvr01 /reboot /timeout:60
```

Managing IIS Services

Each IIS server in the organization relies on a set of services for publishing pages, transferring files, and more. To manage IIS services, you'll use the Services node in the Computer Management console, which you start by doing the following:

1. Click Start and choose All Programs, Administrative Tools, and then Computer Management. Or select Computer Management in the Administrative Tools folder in Control Panel.
2. Right-click the Computer Management entry in the console tree and select Connect To Another Computer on the shortcut menu. You can now choose the IIS server whose services you want to manage.
3. Expand the Services And Applications node by clicking the plus sign (+) next to it and then choose Services.

Figure 2-5 shows the Services view in the Computer Management console.



Name	Description	Status	Startup Type	Log On As
Event Log	Enables ev...	Started	Automatic	Local System
File Replication	Allows files...		Manual	Local System
File Server for Macintosh	Enables Ma...	Started	Automatic	Local System
FTP Publishing Service	Enables thi...	Started	Automatic	Local System
Help and Support	Enables He...	Started	Automatic	Local System
HTTP SSL	This servic...	Started	Manual	Local System
Human Interface Device Access	Enables ge...		Disabled	Local System
IIS Admin Service	Enables thi...	Started	Automatic	Local System
IMAPI CD-Burning COM Service	Manages C...		Disabled	Local System
Indexing Service	Indexes co...		Manual	Local System
Internet Connection Firewall (ICF)...	Provides n...		Disabled	Local System
Intersite Messaging	Enables me...		Disabled	Local System
IPSEC Services	Provides e...	Started	Automatic	Local System
Kerberos Key Distribution Center	On domain ...		Disabled	Local System
License Logging	Monitors a...		Disabled	Network S...
Logical Disk Manager	Detects an...	Started	Automatic	Local System
Logical Disk Manager Administrativ...	Configures...	Started	Manual	Local System
Message Queuing	Provides a ...	Started	Automatic	Local System

Figure 2-5. *Use the Services node to manage IIS services.*

The key fields of this dialog box are used as follows:

- **Name** The name of the service.
- **Description** A short description of the service and its purpose.
- **Status** The status of the service as started, paused, or stopped. (Stopped is indicated by a blank entry.)
- **Startup Type** The startup setting for the service.

Note Automatic services are started when the system boots up. Manual services are started by users or other services. Disabled services are turned off and can't be started.

- **Log On As** The account the service logs on as. The default in most cases is the local system account.

Key IIS Services

Table 2-4 provides a summary of key services that IIS uses or depends on. Note that the services available on a particular IIS server depend on its configuration. Still, this is the core set of services that you'll find on most IIS servers.

Table 2-4. Key IIS Services

Name	Description
Event Log	Logs event informational, warning and error messages issued by IIS and other applications
FTP Publishing Service	Provides services for transferring files using FTP and also allows administration of an FTP server
IIS Admin Service	Allows administration of IIS through the IIS snap-in
Indexing Service	Indexes the contents and properties of files, providing quick access to files through a flexible query language
Network News Transport Protocol (NNTP)	Provides network news services and allows administration of NTTP servers through the IIS snap-in
Simple Mail Transport Protocol (SMTP)	Provides mail transfer services and allows administration of SMTP sites through the IIS snap-in
World Wide Web Publishing Service	Provides services for transferring files using HTTP and also allows administration of an HTTP server

Starting, Stopping, and Pausing IIS Services

As an administrator, you'll often have to start, stop, or pause IIS services. You manage IIS services through the Computer Management console or through the Services utility. When you manage IIS services at this level, you're controlling all sites or virtual servers that use the service. For example, if a computer publishes three Web sites and you stop the World Wide Web Publishing Service, all three Web sites are stopped and are inaccessible.

To start, stop, or pause services in the Computer Management console, follow these steps:

1. In the left-hand pane, right-click the Computer Management entry in the console tree and select Connect to Another Computer on the shortcut menu. You can now choose the IIS server whose services you want to manage.
2. Expand the Services And Applications node by clicking the plus sign (+) next to it, and then choose Services.
3. In the right-hand pane, right-click the service you want to manipulate, and then select Start, Stop, or Pause as appropriate. You can also choose Restart to have Windows stop and then start the service after a brief pause. In addition, if you pause a service, you can use the Resume option to resume normal operation.

Tip When services that are set to start automatically fail, the status is listed as Blank and you'll usually receive notification in a dialog box. Service failures can also be logged to the system's event logs. In Windows Server 2003, you can configure actions to handle service failure automatically. For example, you could have Windows Server 2003 attempt to restart the service for you. For details, see the section of this chapter entitled "Configuring Service Recovery."

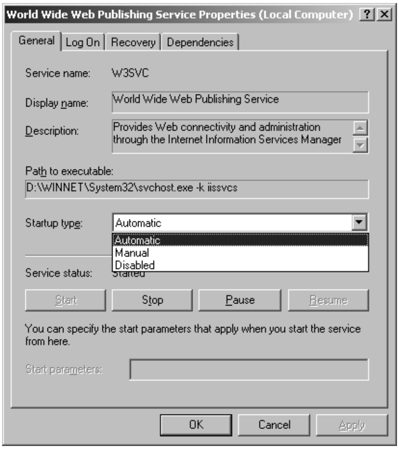
Configuring Service Startup

Essential IIS services are configured to start automatically, and normally they shouldn't be configured with another startup option. That said, if you're troubleshooting a problem, you might want a service to start manually. You might also want to disable a service so that its related virtual servers don't start. For example, if you move an SMTP virtual server to a new server, you might want to disable the SMTP service on the original IIS server. In this way the SMTP service isn't used, but you could turn it on if you need to (without your having to reinstall SMTP support).

You configure service startup as follows:

1. In the left-hand pane of the Computer Management console, connect to the IIS server whose services you want to manage.
2. Expand the Services And Applications node by clicking the plus sign (+) next to it, and then choose Services.
3. In the right-hand pane, right-click the service you want to configure and then choose Properties.

4. On the General tab, use the Startup Type drop-down list to choose a startup option as shown in Figure 2-6. Select Automatic to start the service when the system boots up. Select Manual to allow the service to be started manually. Select Disabled to turn off the service.
5. Click OK.



Service name: W3SVC

Display name: World Wide Web Publishing Service

Description: Provides Web connectivity and administration through the Internet Information Services Manager

Path to executable:
D:\WINNET\System32\svchost.exe -k iissvcs

Startup type: Automatic

Service status: Started

You can specify the start parameters that apply when you start the service from here.

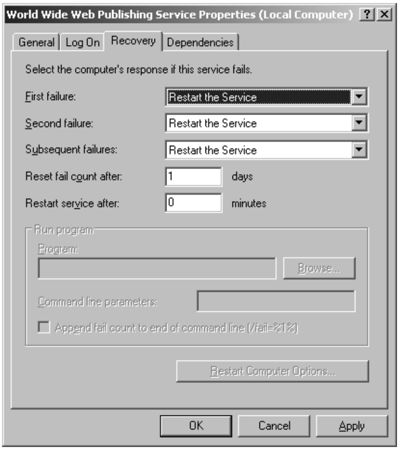
Start parameters:

Figure 2-6. *For troubleshooting, you might want to change the service startup option.*

Configuring Service Recovery

You can configure Windows services to take specific actions when a service fails. For example, you could attempt to restart the service or reboot the server. To configure recovery options for a service, follow these steps:

1. In the left-hand pane of the Computer Management console, connect to the computer whose services you want to manage.
2. Expand the Services And Applications node by clicking the plus sign (+) next to it and then choose Services.
3. In the right-hand pane, right-click the service you want to configure and then choose Properties.
4. Select the Recovery tab, shown in Figure 2-7. You can now configure recovery options for the first, second, and subsequent recovery attempts. The available options are:
   - Take No Action
   - Restart The Service
   - Run A Program
   - Restart The Computer



World Wide Web Publishing Service Properties (Local Computer)

Select the computer's response if this service fails.

First failure: Restart the Service

Second failure: Restart the Service

Subsequent failures: Restart the Service

Reset fail count after: 1 days

Restart service after: 0 minutes

Run program

Program:

Command line parameters:

☐ Append fail count to end of command line (/fail=%1%)

Figure 2-7. *You can configure services to recover automatically in case of failure.*

5. Configure other options based on your previously selected recovery options. If you elected to restart the service, you'll need to specify the restart delay. After stopping the service, Windows Server 2003 waits for the specified delay before trying to start the service. In most cases a delay of 1–2 minutes should be sufficient.
6. Click OK.

When you configure recovery options for critical services, you *might* want Windows Server 2003 to try to restart the service on the first and second attempts and then reboot the server on the third attempt.