Microsoft

Internet Information Services (IIS) 7.0

William R. Stanek

Author and Series Editor

Administrator's Pocket Consultant

Microsoft®

Internet Information Services (IIS) 7.0 Administrator’s Pocket Consultant

William R. Stanek

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

Library of Congress Control Number: 2007939309

Printed and bound in the United States of America.

2 3 4 5 6 7 8 9 10   LSI   9 8 7 6 5 4

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.

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

This book expresses the author’s views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions Editor: Martin DelRe
Developmental Editor: Karen Szall
Project Editor: Maureen Zimmerman
Editorial Production: Interactive Composition Corporation
Technical Reviewer: Bob Hogan; Technical Review services provided by Content Master, a member of CM Group, Ltd.

Body Part No. X14-25665

*To my wife and children for their love, their support,
and their extraordinary ability to put up with the
clackety-clackety of my keyboard.*

Contents at a Glance

Table of Contents

Acknowledgments

Writing *Internet Information Services (IIS) 7.0 Administrator's Pocket Consultant* was a lot of fun—and a lot of work. As I set out to write this book, my first goal was to determine what had changed between IIS 6 and IIS 7.0 and what new administration options were available. With any product, and especially with IIS 7.0, this meant doing a great deal of research to determine exactly how things work and a lot of digging into the configuration internals. Thankfully I'd already written many books on IIS, Web technologies, and Web publishing, so I had a starting point of reference for my research—but it was by no means a complete one.

When you start working with IIS 7.0, you'll see at once that this release is different from previous releases. What won't be readily apparent, however, is just how different IIS 7.0 is from its predecessors—and that's because many of the most significant changes to the product are under the surface. These changes affect the underlying architecture and not just the interfaces—and these changes were some of the hardest for me to research and write about.

Because pocket consultants are meant to be portable and readable—the kind of book you use to solve problems quickly and easily and get the job done wherever you might be—I had to carefully review my research to make sure I focused on the core of IIS 7.0 administration. The result is the book you hold in your hand, which I hope you'll agree is one of the best practical, portable guides to IIS 7.0.

It is gratifying to see techniques I've used time and again to solve problems put into a printed book so that others may benefit from them. But no man is an island, and this book couldn't have been written without help from some very special people. As I've stated in all my previous books with Microsoft Press, the team at Microsoft Press is top-notch. Throughout the writing process, Maureen Zimmerman was instrumental in helping me stay on track and getting the tools I needed to write this book. Maureen did a top-notch job managing the editorial process. Thanks also to Martin DelRe for believing in my work and shepherding it through production.

Unfortunately for the writer (but fortunately for readers), writing is only one part of the publishing process. Next came editing and author review. I must say, Microsoft Press has the most thorough editorial and technical review process I've seen anywhere—and I've written a lot of books for many different publishers. Bob Hogan was the technical editor for the book. Joel Rosenthal was the copy editor for the book. I want to thank both of them for their careful reviews. Bob and Joel are great to work with!

I also want to thank Lucinda, Jack, Karen, Denise, and everyone else at Microsoft who has helped me during this project. Hopefully, I haven't forgotten anyone but if I have, it was an oversight. *Honest.*;-)

Introduction

Welcome to *Internet Information Services (IIS) 7.0 Administrator's Pocket Consultant*. As the author of over 65 books, I've been writing professionally about Web publishing and Web servers since 1994. Over the years, I've written about many different Web server technologies and products, but my favorite has always been Internet Information Services (IIS). IIS provides the core services for hosting Web servers, Web applications, and Microsoft Windows SharePoint services. From top to bottom, IIS 7.0 is substantially different from earlier versions of IIS. For starters, the underlying configuration architecture for IIS has been completely reconstructed—IIS configuration architecture is now based entirely on Extensible Markup Language (XML) and XML schema.

Having written many top-selling Web publishing and XML books, I was able to bring a unique perspective to this book—the kind of perspective you can gain only after working with technologies for many years. You see, long before IIS 7.0 architecture was built on XML and related technologies, I was working with, researching, and writing about these technologies. The advantage for you, the reader, is that my solid understanding of these technologies allowed me to dig into the IIS configuration architecture and to provide a comprehensive roadmap to this architecture and the hundreds of related configuration settings in this book.

In addition, as you've probably noticed, there's more than enough information about IIS 7.0 on the Web and in other printed books. There are tutorials, reference sites, discussion groups, and more to help make it easier to use IIS 7.0. However, the advantage to reading this book instead is that all the information you need to learn IIS 7.0 is organized in one place and presented in a straightforward and orderly fashion. This book has everything you need to customize IIS installations, master IIS configuration, and maintain IIS servers.

In this book, I teach you how features work, why they work the way they work, and how to customize them to meet your needs. You'll also learn why you may want to use certain features and when to use other features to resolve any issues you are having. In addition, this book provides tips, best practices, and examples of how to optimize IIS 7.0 to meet your needs. This book won't just teach you how to configure IIS—it'll teach you how to squeeze every last bit of power out of the application and how to make the most of the features and options included in IIS 7.0.

Also, unlike many other books on the subject, this book doesn't focus on a specific user level. This isn't a lightweight, beginners-only book. Regardless of whether you are a novice administrator or a seasoned professional, many of the concepts in this book will be valuable to you. And you'll be able to apply them to your IIS server installations.

Who Is This Book For?

Internet Information Services (IIS) 7.0 Administrator's Pocket Consultant covers core services for hosting Web servers, Web applications, and Windows SharePoint services. The book is designed for:

- Current Microsoft Web administrators and developers
- Administrators and developers of intranets and extranets
- Administrators and developers migrating to Microsoft Web-based solutions
- Programmers, engineers, and quality assurance personnel who manage internal or test servers running any of these services

To pack in as much information as possible, I assume that you already have basic networking skills and a basic understanding of Web servers. With this in mind, I don't devote entire chapters to understanding the World Wide Web, Web services, or Web servers. I do, however, cover configuration, enterprise-wide server management, performance tuning, optimization, automation, and much more.

I also assume that you're fairly familiar with the standard Windows user interface and that if you plan to use the scripting techniques outlined in the book, you know scripting. If you need help learning Windows or scripting, you should read other resources (many of which are available from Microsoft Press and the Microsoft Web site as well).

How This Book Is Organized

Rome wasn't built in a day, and this book wasn't intended to be read in a day, a week, or even 21 days. Ideally, you'll read this book at your own pace, a little each day, as you work your way through all the features IIS has to offer. This book is organized into four parts with 14 chapters and a comprehensive reference in an appendix. The chapters are arranged in a logical order, taking you from planning and deployment tasks to configuration and maintenance tasks.

Speed and ease of reference are essential parts of this hands-on guide. This book has an expanded table of contents and an extensive index to help you find answers to problems quickly. Many other quick reference features have been added to the book as well, including quick step-by-step instructions, lists, tables with fast facts, and extensive cross-references.

As with all the books in the Pocket Consultant series, *Internet Information Services (IIS) 7.0 Administrator's Pocket Consultant* is designed to be a concise and easy-to-use resource for managing Web servers running IIS. This book is the readable resource guide that you'll want on your desktop at all times, as it covers everything you'll need to perform core Web administration tasks. Because the focus is on giving you maximum value in a pocket-sized guide, you don't have to wade through hundreds of pages of extraneous information to find what you're looking for. Instead, you'll find exactly what you need to get the job done.

In short, this book is designed to be the one resource you can turn to whenever you have questions regarding Web server administration. To this end, the book zeroes in on daily administration procedures, frequently used tasks, documented examples, and options that are representative while not necessarily inclusive. One of the key goals I had when writing this book is to keep the content so concise that the book remains compact and easy to navigate, while ensuring that it is packed with as much information as possible. Thus, rather than a hefty 1,000-page tome or a lightweight 100-page quick reference, you get a valuable resource guide that can help you quickly and easily perform common tasks, solve problems, and implement advanced IIS techniques, such as failed request tracing, handler mapping, customized Hypertext Transfer Protocol (HTTP) redirection, and integrated request processing.

Conventions Used in This Book

I've used a variety of elements to help keep the text clear and easy to follow. You'll find code terms and listings in `monospace` type, except when I tell you to actually type a command. In that case, the command appears in **bold** type. When I introduce and define a new term, I put it in *italics*.

Other conventions include:

- **Note** Provides additional details on a particular point that needs emphasis.
- **Tip** Offers helpful hints or additional information.
- **Caution** Warns you when there are potential problems you should look out for.
- **More Info** Points to more information on the subject.
- **Real World** Provides real-world advice when discussing advanced topics.
- **Best Practice** Examines the best technique to use when working with advanced configuration and administration concepts.

I truly hope you find that *Internet Information Services (IIS) 7.0 Administrator's Pocket Consultant* provides everything you need to perform the essential administrative tasks on IIS servers as quickly and efficiently as possible. You are welcome to send your thoughts to me at *williamstanek@aol.com*.

Other Resources

No single magic bullet exists for learning everything you'll ever need to know about IIS. While some books claim to be all-in-one guides, there's simply no way one book can do it all. With this in mind, I hope you'll use this book as it is intended to be used: as a concise and easy-to-use resource.

Your current knowledge will largely determine your success with this or any other IIS resource or book. As you encounter new topics, take the time to practice what you've learned and read about. Seek out further information as necessary to get the practical hands-on know-how and knowledge you need.

Throughout your studies, I recommend that you regularly visit Microsoft's IIS site (*http://www.iis.net*) and Microsoft's support site (*http://support.microsoft.com*) to stay current with the latest changes in the software. To help you get the most out of this book, there's a corresponding Web site at *http://www.williamstanek.com/iis* which contains information about IIS, updates to the book, and updated information about IIS.

Support

Every effort has been made to ensure the accuracy of this book. Microsoft Press provides corrections for its books at the following address:

http://www.microsoft.com/mspress/support

If you have comments, questions, or ideas about this book, please send them to Microsoft Press using either of the following methods:

Postal Mail:

Microsoft Press
Attn: *Internet Information Services (IIS) 7.0 Administrator's Pocket Consultant* Editor
One Microsoft Way
Redmond, WA 98052-6399

E-mail:

MSPINPUT@MICROSOFT.COM

Chapter 1
IIS 7.0 Administration Overview

Let's start with the bad news right up front: Internet Information Services (IIS) 7.0 isn't what you think it is. Although IIS 7.0 *is* the latest release of Internet Information Services, it *isn't* what it seems. IIS does look a lot like its predecessors, but this is deceiving because under the surface, the architecture is completely different. So much has changed, in fact, that perhaps it might have been better if Microsoft had given IIS 7.0 a new product name. That way you'd know that IIS 7.0 was completely different from its predecessors, allowing you to start with a fresh perspective and a reasonable expectation of having to learn a whole new bag of tricks. Seasoned IIS pros also are going to have to unlearn some old tricks; and that's not only going to be difficult, it might be the single biggest obstacle to mastering IIS 7.0.

IIS 7.0 provides the core services for hosting Web servers, Web applications, and Microsoft Windows SharePoint Services. Throughout this book, I'll refer to administration of IIS, Web applications, and Windows SharePoint Services as *Microsoft Web administration* or simply *Web administration*. As you get started with Microsoft Web administration, you should concentrate on these key areas:

- What's new or changed in IIS 7.0
- How IIS 7.0 configuration schema and global configuration architecture are used
- How IIS 7.0 works with your hardware
- How IIS works with Windows-based operating systems
- Which administration tools are available
- Which administration techniques you can use to manage and maintain IIS

Working with IIS 7.0: What You Need to Know Right Now

Microsoft fully integrated Microsoft ASP.NET and the Microsoft .NET Framework into IIS 7.0. Unlike IIS 6, IIS 7.0 takes ASP.NET and the .NET Framework to the next level by integrating the ASP.NET runtime extensibility model with the core server architecture, allowing developers to fully extend the core server architecture by using ASP.NET and the .NET Framework. This tighter integration makes it possible to use existing ASP.NET features such as .NET Roles, Session Management, Output Caching, and Forms Authentication with all types of content.

IIS 7.0 has generalized the Hypertext Transfer Protocol (HTTP) process activation model that IIS 6 introduced with application pools and made it available for all protocols through an independent service called the Windows Process Activation Service, and developers can use Windows Communication Foundation (WCF) protocol adapters to take advantage of the capabilities of this service. You also should know up front that IIS 7.0 includes a metabase compatibility component that allows your existing scripts and applications to continue running but does not use a metabase to store configuration information. Instead of a metabase, IIS 7.0 uses a distributed configuration system with global and application-specific configuration files that are based on a customizable set of Extensible Markup Language (XML) schema files. These XML schema files define the configuration elements and attributes in addition to valid values for those elements and attributes, providing you precise control over exactly how you can configure and use IIS.

Microsoft built the configuration system around the concept of modules. *Modules* are standalone components that provide the core feature set of an IIS server. Microsoft ships more than 40 independent modules with IIS 7.0. Either these modules are IIS 7.0–native modules that use a Win32 DLL or IIS 7.0–managed modules that use a .NET Framework Class Library contained within an assembly. Because all server features are contained within modules, you can modify the available features easily by adding, removing, or replacing a server's modules. Further, by optimizing the installed modules based on the way an IIS server is used, you can enhance security by reducing the attack surface area and improve performance by reducing the resources required to run the core services.

> **Note** Because modules are such an important part of IIS 7.0, you'll find much discussion about them and how they are used in this book. Chapter 2, "Deploying IIS 7.0 in the Enterprise," introduces all the available native and managed modules. Chapter 5, "Managing Global IIS Configuration," details how to install and manage modules. The appendix, "Comprehensive IIS 7.0 Module and Schema Reference," provides a complete guide to using modules and schemas.

IIS 7.0 is more secure than IIS 6 because of built-in request filtering and rules-based Uniform Resource Locator (URL) authorization support. You can configure request filtering to reject suspicious requests by scanning URLs sent to a server and filtering out unwanted requests. You can configure URL authorization rules to require logon and allow or deny access to specific URLs based on user names, .NET roles, and HTTP request methods. To make it easier to resolve problems with the server and Web applications, IIS 7.0 includes new features for diagnostics, real-time request reviewing, and error reporting. These features allow you to:

- View the current running state of the server.
- Trace failed requests through the core server architecture.
- Obtain detailed error information to pinpoint the source of a problem.

IIS 7.0 has many other new and enhanced features, but few are as important as the new set of administration tools, including new graphical, command-line, and scripting administration tools. The new graphical administration tool uses a browser-like interface and adds features for delegated administration, remote administration over Secure HTTP (HTTPS), and extensibility through custom user interface components. The new command-line administration tool makes it possible to perform most configuration tasks with a single line of command text. With ASP.NET, you can manage IIS configuration through the .NET Framework by using the Microsoft.Web.Administrators application programming interface (API). With scripting, you can manage IIS configuration through the IIS 7.0 Windows Management Instrumentation (WMI) provider.

Because of the many changes, much of what you know about IIS is obsolete or irrelevant. But 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 7.0 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 the .NET Framework will only grow over time, and the more you learn about the heart of the .NET architecture—IIS 7.0—the better prepared you'll be for now and for the future.

With IIS 7.0, key components that were a part of previous IIS releases are no longer available or work in different ways than they did before. Because IIS 7.0 does not use a metabase, applications designed for IIS 6 will not run on IIS 7.0 without special actions being taken. To run IIS 6 applications, you must install the IIS 6 compatibility and metabase feature. To manage IIS 6 applications and features, you must install IIS 6 Manager, IIS 6 scripting tools, and IIS 6 WMI compatibility. Additionally, IIS 7.0 does not include Post Office Protocol version 3 (POP3) or Simple Mail Transfer Protocol (SMTP) services. With IIS 7.0, you can send e-mail messages from a Web application by using the SMTP E-mail component of ASP.NET.

IIS Manager is the graphical user interface (GUI) for managing both local and remote installations of IIS 7.0. To use IIS Manager to manage an IIS server remotely, Web Management Service (WMSVC) must be installed and started on the IIS server you want to manage remotely. WMSVC is also required when IIS site or application administrators want to manage features over which they've been delegated control.

The Web Management Service provides a hostable Web core that acts as a standalone Web server for remote administration. After you install and start WMSVC on an IIS server, it listens on port 8172 on all unassigned IP addresses for four specific types of requests:

- **Login Requests** IIS Manager sends login requests to WMSVC to initiate connections. On the hostable Web core, login requests are handled by Login.axd. The authentication type is either NT LAN Manager (NTLM) or Basic, depending on what you select when you are prompted to provide credentials in the connection dialog box.

- **Code Download Requests** If login is successful, WMSVC returns a list of user interface (UI) modules for the connection. Each IIS Manager page corresponds to a specific UI module. If there's a module that IIS Manager doesn't have, it will request to download the module binaries. Code download requests are handled by Download.axd.
- **Management Service Requests** After a connection is established, your interactions with IIS Manager cause management service requests. Management service requests direct module services in WMSVC to read or write configuration data, runtime state, and providers on the server. Management service requests are handled by Service.axd.
- **Ping Requests** Ping requests are made from within the WMSVC service to the hostable Web core. Ping requests are made by Ping.axd to ensure that the hostable Web core continues to be responsive.

The Web Management Service stores a limited set of editable configuration values in the registry. Each time the service is started, the Web configuration files are regenerated in the following directory: *%SystemRoot%*\ServiceProfiles\LocalService\AppData\Local\Temp\WMSvc. To enhance security, WMSVC requires SSL (HTTPS) for all connections. This ensures that data passed between the remote IIS Manager client and WMSVC is secure. Additionally, WMSVC runs as Local Service with a reduced permission set and a locked down configuration. This ensures that only the minimal set of required modules are loaded when the hostable Web core starts. See Chapter 3, "Core IIS 7.0 Administration," for more information.

> **Note** *%SystemRoot%* refers to the SystemRoot environment variable. The Windows operating system hasmany environment variables, which are used torefer to user- and system-specific values. Often, I'll refer to environment variables in this book using this syntax: *%VariableName%*.

Introducing IIS 7.0 Configuration Architecture

You can use IIS 7.0 to publish information on intranets, extranets, and the Internet. Because today's Web sites use related features, such as ISAPI filters, ASP, ASP.NET, CGI, and the .NET Framework, IIS bundles these features as part of a comprehensive offering. What you need to know right now about IIS 7.0 is how IIS 7.0 uses the configuration schema and its global configuration system. In Chapter 2, you'll learn about the available setup features and the related configuration modules.

IIS 7.0 Configuration Schema

Unlike IIS 6, in which the main configuration information is stored in metabase files, IIS 7.0 has a unified configuration system for storing server, site, and application settings. You can manage these settings by using an included set of managed code, scripting APIs, and management tools. You can also manage these settings by directly editing the configuration files themselves. Direct editing of configuration files is possible because the files use XML and are written in plain-language text files based on a predefined set of XML schema files.

> **Note** IIS 7.0 always takes the master state for configuration from the configuration files. This is a dramatic change from IIS 6, in which the master state was taken from the in-memory configuration database, which was flushed periodically to disk.

Using the XML schema to specify the configuration settings ensures that the related configuration files are well-structured XML, which is easy to modify and maintain. Because configuration values are stored using easy-to-understand text strings and values, they are easy to work with. By examining the schema itself, you can determine the exact set of acceptable values for any configuration option. IIS shares the same schema with ASP.NET configuration, ensuring that configuration settings for ASP.NET applications are just as easy to manage and maintain.

On an IIS server, schema files are stored in the *%SystemRoot%*\System32\Inetsrv \Config\Schema directory. The four standard schema files are:

- **IIS_schema.xml** This file provides the IIS configuration schema.
- **ASPNET_schema.xml** This file provides the ASP.NET configuration schema.
- **FX_schema.xml** This file provides the .NET Framework configuration schema (providing features beyond what the ASP.NET schema offers).
- **rscaext.xml** This file provides the Runtime Status and Control API (RSCA) configuration schema, providing dynamic properties for obtaining detailed runtime data.

IIS reads in the schema files automatically during startup of the application pools. The IIS schema file is the master schema file. Within the IIS schema file, you'll find configuration sections for each major feature of IIS, from application pooling to failed request tracing. The ASP.NET schema file builds on and extends the master schema with specific configuration sections for ASP.NET. Within the ASP.NET schema file, you'll find configuration sections for everything from anonymous identification to output cache settings. The FX schema file builds on and extends the ASP.NET schema file. Within the FX schema file, you'll find configuration settings for application settings, connection strings, date-time serialization, and more.

Whereas configuration sections are also grouped together for easier management, section groups do not have schema definitions. If you want to extend the configuration features and options available in IIS, you can do this by extending the XML schema. You extend the schema by following these basic steps:

1. Specify the desired configuration properties and the necessary section container in an XML schema file.
2. Place the schema file in the *%SystemRoot%*\System32\Inetsrv\Config\Schema directory.
3. Reference the new section in IIS 7.0's global configuration file.

The basic syntax for a schema file is as follows:

```
<!–
The text within this section is a comment. It is standard
practice to provide introductory details in the comments at the
beginning of the schema file.
-->
<configSchema>
    <sectionSchema name="configSection1">
    </sectionSchema>
    <sectionSchema name="configSection2">
    </sectionSchema>
    <sectionSchema name="configSection3">
    </sectionSchema>
</configSchema>
```

As an administrator or developer, you don't necessarily need to be able to read and interpret XML schemas to succeed. However, because having a basic understanding of schemas is helpful, I'll introduce the essentials. Within schema files, configuration settings are organized into sets of related features called *schema sections*. The schema for a configuration section is defined in a <sectionSchema> XML element. For example, the features related to the HTTP listener in IIS are defined with a schema section named system.applicationHost/listenerAdapters. In the IIS_schema.xml file, this section is defined as follows:

```
<sectionSchema name="system.applicationHost/listenerAdapters">
 <collection addElement="add" >
  <attribute name="name" type="string" required="true" isUniqueKey="true" />
  <attribute name="identity" type="string" />
  <attribute name="protocolManagerDll" type="string" />
  <attribute name="protocolManagerDllInitFunction" type="string" />
 </collection>
</sectionSchema>
```

This schema definition states that the system.applicationHost/listenerAdapters element can contain a collection of add elements with the following attributes:

- **name** A unique string that is a required part of the add element.
- **identity** An identity string that is an optional part of the add element.
- **protocolManagerDll** A string that identifies the protocol manager DLL.
- **protocolManagerDllInitFunction** A string that identifies the initialization function for the protocol manager DLL.

An attribute of an element is either optional or required. If the attribute definition states required="true" as with the name attribute, the attribute is required and must be provided when you are using the related element. Otherwise, the attribute is considered optional and does not need to be provided when you are using the related element. In addition to being required, attributes can have other enforced conditions, including:

- **isUniqueKey** If set to true, the related value must be unique.
- **encrypted** If set to true, the related value is expected to be encrypted.

With some attributes, you'll see default values and possibly an enumerated list of the acceptable string values and their related internal values. In the following example, the identityType attribute has a default value of NetworkService and a list of other possible values:

```
<attribute name="identityType" type="enum" defaultValue="NetworkService">
 <enum name="LocalSystem" value="0"/>
 <enum name="LocalService" value="1"/>
 <enum name="NetworkService" value="2"/>
 <enum name="SpecificUser" value="3"/>
</attribute>
```

The friendly name of a value is provided to make the value easier to work with. The actual value used by IIS is provided in the related value definition. For example, if you set identityType to LocalService, the actual configuration value used internally by IIS is 2.

As a standard rule, you cannot use enumerated values in combination with each other. Because of this, the identityType attribute can have only one possible value. In contrast, attributes can have flags, which can be used together to form combinations of values. In the following example, the logEventOnRecycle attribute uses flags and has a default set of flags that are used in combination with each other:

```
<attribute name="logEventOnRecycle" type="flags" defaultValue="Time,
Memory, PrivateMemory">
 <flag name="Time" value="1"/>
 <flag name="Requests" value="2"/>
 <flag name="Schedule" value="4"/>
 <flag name="Memory" value="8"/>
 <flag name="IsapiUnhealthy" value="16"/>
 <flag name="OnDemand" value="32"/>
 <flag name="ConfigChange" value="64"/>
 <flag name="PrivateMemory" value="128"/>
</attribute>
```

Again, the friendly name is provided to make the value easier to work with. The actual value used by IIS is the sum of the combined flag values. With a setting of "Time, Requests, Schedule," the logEventOnRecycle attribute is set to 7 (1+2+4=7).

Attribute values can also have validation. IIS performs validation of attribute values when parsing the XML and when calling the related API. Table 1-1 provides an overview of the validators you'll see in schemas.

Table 1-1 Summary of Attribute Validation Types in an IIS XML Schema

Validation Type	Validation Parameter	Validation Fails If...
validationType= "applicationPoolName"	validationParameter=""	A validated value contains these characters: |<>&\"
validationType= "integerRange"	validationParameter= "<minimum>, <maximum>[,exclude]"	A validated value is outside [inside] range, in integers.
validationType= "nonEmptyString"	validationParameter=""	A validated value has a string value that is not set.
validationType= "siteName"	validationParameter=""	A validated value contains these characters: /\.?
validationType= "timeSpanRange"	validationParameter= "<minimum>,<maximum>, <granularity> [,exclude]"	A validated value is outside [inside] range, in seconds.
validationType= "requireTrimmedString"	validationParameter=""	A validated value has white space at start or end of value.

IIS 7.0 Global Configuration System

IIS uses a global configuration system that is difficult to understand at first but gets easier and easier to understand once you've worked with it awhile. Because there's no sense trying to ease into this, I'll dive right in. If you'll hang with me for a few pages, I'll get you through the roughest parts and zero in on exactly what you need to know—I promise.

IIS configuration settings are stored in configuration files that together set the running configuration of IIS and related components. One way to think of a configuration file is as a container for the settings you apply and their related values. You can apply multiple configuration files to a single server and the applications it is running. Generally, you manage configuration files at the .NET Framework root level, the server root level, and the various levels of a server's Web content directories. A server's Web content directories include the root directory of the server itself, the root directories of configured Web sites, and any subdirectories within Web sites. The root levels and the various levels of a server's Web content directories can be described as containers for the settings you apply and their values. If you know a bit about object-oriented programming, you might expect the concepts of parent-child relationship and inheritance to apply—and you'd be right.

Through inheritance, a setting applied at a parent level becomes the default for other levels of the configuration hierarchy. Essentially, this means that a setting applied at a parent level is passed down to a child level by default. For example, if you apply a setting at the server root level, the setting is inherited by all Web sites on the server and by all the content directories within those sites.

The order of inheritance is as follows:

```
.NET Framework root ➔ server root ➔ Web Site root ➔
top-level directory ➔ subdirectory
```

This means that the settings for the current .NET Framework root are passed down to IIS, the settings for IIS are passed down to Web sites, and the settings for Web sites are passed down to content directories and subdirectories. As you might expect, you can override inheritance. To do this, you specifically assign a setting for a child level that contradicts a setting for a parent. As long as overriding a setting is allowed (that is, overriding isn't blocked), the child level's setting will be applied appropriately. To learn more about overriding and blocking, see Chapter 5.

When working with the configuration files, keep the following in mind:

- The .NET Framework root IIS applies depends on the current running version of ASP.NET and the .NET Framework. The default configuration files for the .NET Framework root are Machine.config and Web.config, which are stored in the *%SystemRoot%*\Microsoft.net\Framework\Version\Config\Machine.config directory. Machine.config sets the global defaults for the .NET Framework settings in addition to some ASP.NET settings. Web.config sets the rest of the global defaults for ASP.NET. See Chapter 8, "Running IIS Applications," and Chapter 9, "Managing Applications, Application Pools, and Worker Processes," for more information about configuring the .NET Framework and ASP.NET.

- The default configuration file for the server root is ApplicationHost.config, which is stored in the *%SystemRoot%*\System32\Inetsrv\Config directory. This file controls the global configuration of IIS. See Chapter 5 for more information about configuring IIS servers.

- The default configuration file for a Web site root is Web.config. This file is stored in the root directory of the Web site to which it applies and controls the behavior for the Web site. See Chapters 8 and 9 for more information about configuring IIS applications.

- The default configuration file for a top-level content directory or a content subdirectory is Web.config. This file is stored in the content directory to which it applies and controls the behavior of that level of the content hierarchy and downwards. See Chapter 6 for more information about configuring content directories.

In some cases, you may want a .config file to include some other .config file. This can be done by using the configSource attribute to refer to the .config file containing the settings you want to use. Currently, the referenced .config file must reside in the same directory as the original .config file. Note that this behavior may change to allow .config files in other directories to be used. To see how this works, consider the following example from the ApplicationHost.config file:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- applicationHost.config -->
<configuration>
 <system.webServer>
  <httpErrors>
   <error statusCode="401" prefixLanguageFilePath="%SystemDrive%\
inetpub\custerr" path="401.htm" />
   <error statusCode="403" prefixLanguageFilePath="%SystemDrive%\
inetpub\custerr" path="403.htm" />
   <error statusCode="404" prefixLanguageFilePath="%SystemDrive%\
inetpub\custerr" path="404.htm" />
   <error statusCode="405" prefixLanguageFilePath="%SystemDrive%\
inetpub\custerr" path="405.htm" />
   <error statusCode="406" prefixLanguageFilePath="%SystemDrive%\
inetpub\custerr" path="406.htm" />
   <error statusCode="412" prefixLanguageFilePath="%SystemDrive%\
inetpub\custerr" path="412.htm" />
   <error statusCode="500" prefixLanguageFilePath="%SystemDrive%\
inetpub\custerr" path="500.htm" />
   <error statusCode="501" prefixLanguageFilePath="%SystemDrive%\
inetpub\custerr" path="501.htm" />
   <error statusCode="502" prefixLanguageFilePath="%SystemDrive%\
inetpub\custerr" path="502.htm" />
  </httpErrors>
 </system.webServer>
</configuration>
```

In this example, error elements specify how certain types of HTTP error status codes should be handled. If you wanted to customize the error handling for a server, you might want to extend or modify the default values in a separate .config file and then reference the .config file in ApplicationHost.config. To do this, you would update the ApplicationHost.config file to point to the additional .config file. An example follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- applicationHost.config -->
<configuration>
 <system.webServer>
  <httpErrors configSource=errorMode.config />
</configuration>
```

You would then create the errorMode.config file and store it in the same directory as the ApplicationHost.config file. The following is an example of the contents of the errorMode.config file:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- errorMode.config -->
```

```
<configuration>
 <system.webServer>
  <httpErrors>
  <error statusCode="401" prefixLanguageFilePath="%SystemDrive%\inetpub\
custerr" path="401.htm" />
  <error statusCode="403" prefixLanguageFilePath="%SystemDrive%\inetpub\
custerr" path="403.htm" />
  <error statusCode="404" prefixLanguageFilePath="%SystemDrive%\inetpub\
custerr" path="404.htm" />
  <error statusCode="405" prefixLanguageFilePath="%SystemDrive%\inetpub\
custerr" path="405.htm" />
  <error statusCode="406" prefixLanguageFilePath="%SystemDrive%\inetpub\
custerr" path="406.htm" />
  <error statusCode="412" prefixLanguageFilePath="%SystemDrive%\inetpub\
custerr" path="412.htm" />
  <error statusCode="500" prefixLanguageFilePath="%SystemDrive%\inetpub\
custerr" path="500.htm" />
  <error statusCode="501" prefixLanguageFilePath="%SystemDrive%\inetpub\
custerr" path="501.htm" />
  <error statusCode="502" prefixLanguageFilePath="%SystemDrive%\inetpub\
custerr" path="502.htm" />
   </httpErrors>
  </system.webServer>
</configuration>
```

When you make these or other types of changes in configuration files, you don't need to worry about restarting IIS or related services. IIS automatically picks up the changes and uses them. In these examples, you'll note that we're working with the system.webServer section of the configuration file. As per the schema definition files, all settings are defined within specific configuration sections. Although sections cannot be nested, a section can exist within a section group, and that section group can in turn be contained in a parent section group. A section group is simply a container of logically related sections.

In ApplicationHost.config, section groups and individual sections are defined in the configSections element. The configSections element controls the registration of sections. Every section belongs to one section group. By default, ApplicationHost.config contains these section groups:

- **system.applicationHost** Defines the following sections: applicationPools, configHistory, customMetadata, listenerAdapters, log, sites, and webLimits.
- **system.webServer** Defines the following sections: asp, caching, cgi, defaultDocument, directoryBrowse, globalModules, handlers, httpCompression, httpErrors, httpLogging, httpProtocol, httpRedirect, httpTracing, isapiFilters, modules, odbcLogging, serverRuntime, serverSideInclude, staticContent, urlCompression, and validation. Includes the security and tracing subgroups.

- **system.webServer.security** A subgroup of system.webServer that defines the following sections: access, applicationDependencies, authorization, ipSecurity, isapiCgiRestriction, and requestFiltering. Includes the authentication subgroup.
- **system.webServer.security.authentication** A subgroup of system.webServer .security that defines the following sections: anonymousAuthentication, basicAuthentication, clientCertificateMappingAuthentication, digestAuthentication, iisClientCertificateMappingAuthentication, and windowsAuthentication.
- **system.webServer.security.tracing** A subgroup of system.webServer.security that defines the traceFailedRequests and traceProviderDefinitions sections.

In ApplicationHost.config, section groups and individual sections are defined as follows:

```
<configSections>
 <sectionGroup name="system.applicationHost">
  <section name="applicationPools" allowDefinition="AppHostOnly"
overrideModeDefault="Deny" />
  <section name="configHistory" allowDefinition="AppHostOnly"
overrideModeDefault="Deny" />
  <section name="customMetadata" allowDefinition="AppHostOnly"
overrideModeDefault="Deny" />
  <section name="listenerAdapters" allowDefinition="AppHostOnly"
overrideModeDefault="Deny" />
  <section name="log" allowDefinition="AppHostOnly"
overrideModeDefault="Deny" />
  <section name="sites" allowDefinition="AppHostOnly"
overrideModeDefault="Deny" />
  <section name="webLimits" allowDefinition="AppHostOnly"
overrideModeDefault="Deny" />
  </sectionGroup>
 <sectionGroup name="system.webServer">
  …
 </sectionGroup>
</configSections>
```

In Machine.config, you'll also find definitions for section groups and individual sections. These are similar to those used in ApplicationHost.config but are used for configuring the .NET Framework and some ASP.NET settings. When working with either .config file, keep in mind that a section is the basic unit of deployment, locking, searching, and containment for configuration settings. Every section has a name attribute and optional allowDefinition and overrideModeDefault attributes. The name attribute sets the unique section name. The allowDefinition attribute specifies the level at which the section can be set:

- **Everywhere** The section can be set in any configuration file including directories mapped to virtual directories that are not application roots, and their subdirectories. Because this is the default setting, a section that doesn' t have an allowDefinition attribute uses this setting automatically.
- **MachineOnly** The section can be set only in ApplicationHost.config or Machine.config.

- **MachineToWebRoot** The section can be set only in the .NET Framework root's Machine.config or Web.config file, or in ApplicationHost.config.
- **MachineToApplication** The section can be set only in the .NET Framework root's Machine.config or Web.config file, in ApplicationHost.config, or in Web.config files for application root directories.
- **AppHostOnly** The section can be set only in Web.config files for application root directories.

The OverrideModeDefault attribute sets the default lockdown state of a section. Essentially, this means that it controls whether a section is locked down to the level in which it is defined or can be overridden by lower levels of the configuration hierarchy. If this attribute is not set, the default value is Allow. With Allow, lower level configuration files can override the settings of the related section. With Deny, lower level configuration files cannot override the settings of the related section. As discussed in Chapter 5, you'll typically use location tags to lock or unlock sections for specific Web sites or applications.

Because the complete configuration settings of a server and its related sites and applications are stored in the configuration files, you easily can back up or duplicate a server's configuration. Backing up a server's configuration is a simple matter of creating a copy of the configuration files. Similarly, duplicating a server's configuration on another server is a simple matter of copying the source configuration files to the correct locations on another server.

IIS 7.0 and Your Hardware

Before you deploy IIS 7.0, you should carefully plan the server architecture. As part of your planning, you need to look closely at pre-installation requirements and the hardware you will use. IIS 7.0 is no longer the simple solution for hosting Web sites that it once was. It now provides the core infrastructure for hosting Web servers, Web applications, and Windows SharePoint Services.

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 random access memory (RAM) that's required depends on many factors, including the requirements of other services, the size of frequently accessed content files, and the RAM requirements of the Web applications. In most installations, I recommend that you use at least 1 gigabyte (GB) of RAM. High-volume servers should have a minimum of 2 to 4 GB of RAM. More

RAM will allow more files to be cached, reducing disk requests. For all IIS installations, the operating system paging file size should at least equal the amount of RAM on the server.

> **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, one that has adequate space for the page file to grow, along with providing optimal input/output (I/O) performance.

> **More Info** For detailed information on memory management and performance tuning, see Chapter 12, "Performance Tuning, Monitoring, and Tracing."

- **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 recommended configurations should suffice. Faster clock speeds and multiple processors increase the performance scalability of a Web server, particularly for sites that rely on dynamic content. 32-bit versions of Windows run on Intel x86 or compatible hardware. 64-bit versions of Windows run on the x64 family of processors from AMD and Intel, including AMD64 and Intel Extended Memory 64 Technology (Intel EM64T). IIS provides solid benchmark performance on Intel Xeon, AMD Opteron, and AMD Athlon processors. Any of these CPUs provide good starting points for the typical IIS server. You can achieve significant performance improvements with a large processor cache. Look closely at the L1, L2, and L3 cache options available—a larger cache can yield much better performance overall.
- **SMP** IIS supports symmetric multiprocessors (SMPs) and can use additional processors to improve performance. If the system is running only IIS and doesn't handle 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. 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 interface (SCSI) devices.

- **Data protection** Unless you can tolerate hours of downtime, you should add protection against unexpected drive failures by using RAID. Hardware RAID implementations are always preferred over software RAID implementations. 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 and its contents are restored from backup. Because of this, RAID 0 isn't the recommended choice. RAID 1 (disk mirroring) creates duplicate copies of data on separate physical drives, allowing the server to remain operational when a drive fails, and even while the RAID controller rebuilds a replacement drive in a failed mirror. 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 properly configured UPS system allows the operating system to automatically shut down the server gracefully 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.

IIS 7.0 Editions and Windows

IIS 7.0 is available for both desktop and server editions of Windows. On Windows Vista, IIS 7.0 offers Web administrators and Web developers a complete platform for building and testing dynamic Web sites and Web applications. IIS 7.0 running on Windows Vista also enables process activation, process management, and the necessary HTTP infrastructure for creating WCF–based applications.

As discussed further in Chapter 2, the way IIS 7.0 works on Windows Vista depends on the edition of Windows Vista you are using. On Windows Vista Starter and Home Basic editions, IIS 7.0 cannot be used to host Web sites, Web applications, or Windows SharePoint Services. On these editions, a limited set of IIS features are available, such as Windows Activation Service components that are used to enable WCF-based applications. Users who install WCF-based applications will not need to install these components. The necessary components are installed automatically by WCF. With these editions, the simultaneous request execution limit for IIS is three, meaning that an application or a group of running applications could make up to three simultaneous requests for Web content through the installed IIS components.

On Windows Vista Home Premium, most of the IIS 7.0 features required for Web site development are available. The available features should allow most casual or hobbyist administrators and developers to build and test dynamic Web sites and Web applications. Many advanced features are missing, however, including advanced authentication components, advanced logging components, and FTP server components. As with Starter and Home Basic editions of Windows Vista, the simultaneous request execution limit for IIS is three for Windows Vista Home Premium, meaning you or running applications could make up to three simultaneous requests for Web content through the installed IIS components.

For Windows Vista Business, Enterprise, and Ultimate editions, all IIS 7.0 features are available. This means that professional Web administrators and Web developers have everything necessary to design, build, and test Web sites and Web applications. The simultaneous request execution limit is ten for these editions of Windows Vista, meaning you or running applications could make up to ten simultaneous requests for Web content through the installed IIS components.

With server editions of Windows, you can use IIS to host Web servers, Web applications, and Windows SharePoint Services. All features of IIS 7.0 are available on all editions of Windows Server 2008. On Windows Server operating systems, IIS 7.0 has no request execution limit. This means that an unlimited number of simultaneous requests can be made to the IIS 7.0 server core.

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 by Using Key Administration Tools

Many tools are available for managing Web resources. Key tools you'll use are shown in Table 1-2. 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, if you connect to a new computer in IIS Manager, you can manage all its sites and services remotely from your system.

Table 1-2 Quick Reference for Key Web Administration Tools

Administration Tool	Purpose
Active Directory Users and Computers	Manages domain user, group, and computer accounts.
Computer Management	Manages services, storage, and applications. The Services And Applications node provides quick access to Indexing Service catalogs and IIS sites and servers.

Table 1-2 Quick Reference for Key Web Administration Tools

Administration Tool	Purpose
Data Sources (ODBC)	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 DNS servers.
Event Viewer	Allows you to view and manages events and system logs. If you keep track of system events, you'll know when problems occur.
Internet Information Services (IIS) 6.0 Manager	Manages Web and application server resources that were designed for IIS 6. This tool is included for backward compatibility only.
Internet Information Services (IIS) Manager	Manages Web and application server resources that were designed for IIS 7.0.
Web Management Service (WMSVC)	Allows you to use the IIS Manager to manage Web and application server resources on remote servers.
Reliability and Performance Monitor	Tracks system reliability and performance allowing you to pinpoint performance problems.
Services	Views service information, starts and stops system services, and configures service logons and automated recoveries.

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 that case, you need to install the administration tools on the workstation you're using.

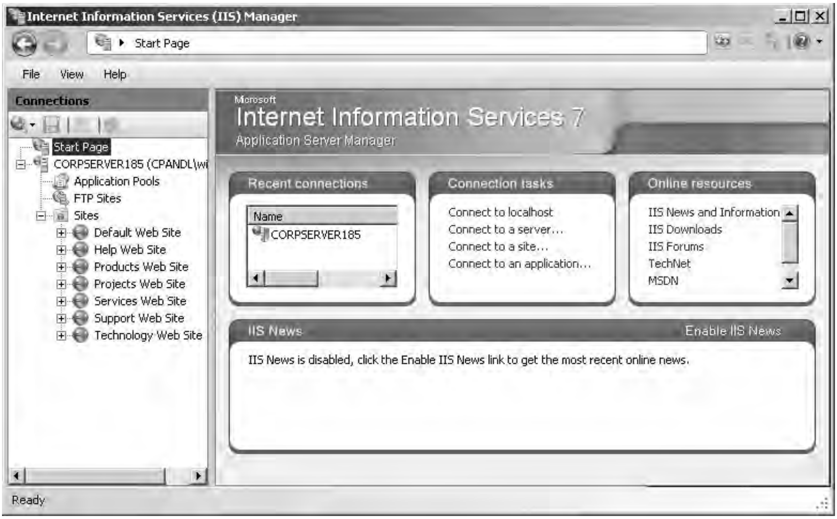
Web Administration Techniques

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

- IIS Manager (InetMgr.exe)
- IIS Administration objects made available through the IIS 7.0 WMI provider
- IIS command-line administration tool (AppCmd.exe)

IIS Manager provides the standard administration interface for IIS. To start IIS Manager, click Start and choose All Programs, Administrative Tools, and then Internet Information Services (IIS) Manager. When started, IIS Manager displays the Start page shown in Figure 1-1 and automatically connects to the local IIS installation, if it's available. On the Start page, you have the following options:

- **Connect to localhost** Connects you to the IIS installation on the local computer
- **Connect to a server** Allows you to connect to a remote server
- **Connect to a site** Allows you to connect to a specific Web site on a designated Web server
- **Connect to an application** Allows you to connect to a specific Web application on a designated site and server



IIS News is disabled, click the Enable IIS News link to get the most recent online news.

Figure 1-1 You can access servers, sites, and applications by using IIS Manager.

As discussed previously, remote access to an IIS server is controlled by the WMSVC. When you install and start WMSVC on an IIS server, it listens on port 8172 on all unassigned IP addresses and allows remote connections from authorized user accounts. You can connect to a remote server by following these steps:

1. In Internet Information Services (IIS) Manager, click Start Page in the console tree and then click Connect To A Server. This starts the Connect To A Server wizard.
2. Type or select the server name in the Server Name box. For a server on the Internet, type the FQDN of the server, such as www.adatum.com. For a server on the local network, type the computer name, such as WEBSVR87. Port 80 is the default port for connections. As necessary, you can provide the port to which you

want to connect. For example, if you want to connect to the server on port 8080, you would follow the server name by :8080, such as WEBSVR87:8080.

3. After you type the server name (and optionally the port number), click Next. IIS Manager will then try to use your current user credentials to log on to the server. If this fails, you'll need to provide the appropriate credentials on the presented Provide Credentials page before clicking Next to continue. Click Finish to complete the connection.

Tip If IIS Manager displays a connection error stating that the remote server is not accepting connections, you'll need to log on locally or through remote desktop. Once logged on, check to ensure the Management Service is started and configured properly. For more information, see the "Enabling and Configuring Remote Administration" section of Chapter 3.

You can connect to a specific Web site on a designated server by following these steps:

1. In Internet Information Services (IIS) Manager, click Start Page in the console tree and then click Connect To A Site. This starts the Connect To A Site Wizard.
2. Type or select the server name in the Server Name box, such as TESTSVR22. In the Site Name box, type or select the name of the Web site to which you want to connect, such as Default Web Site.
3. Click Next. IIS Manager will then try to use your current user credentials to log on to the server. If this fails, you'll need to provide the appropriate credentials on the presented Provide Credentials page before clicking Next to continue. Click Finish to complete the connection.

You can connect to a specific application on a designated site and server by following these steps:

1. In Internet Information Services (IIS) Manager, click Start Page in the console tree and then click Connect To An Application. This starts the Connect To An Application Wizard.
2. Type or select the server name in the Server Name box, such as TESTSVR22. In the Site Name box, type or select the name of the Web site to which you want to connect, such as Default Web Site.
3. In the Application Name box, type or select the relative path of the Web application to which you want to connect, such as /MyApplication or /Apps/Myapp.
4. Click Next. IIS Manager will then try to use your current user credentials to log on to the server. If this fails, you'll need to provide the appropriate credentials on the presented Provide Credentials page before clicking Next to continue. Click Finish to complete the connection.

As Figure 1-2 shows, IIS Manager has been completely redesigned for IIS 7.0. Instead of being a snap-in for the Microsoft Management Console, IIS Manager is now a standalone application with a browser-like interface. Once you connect to a server, site, or application, IIS Manager automatically connects to these installations upon startup. You can change this behavior by disconnecting from the remote server while in IIS Manager. See Chapter 3 for more information on using IIS Manager.

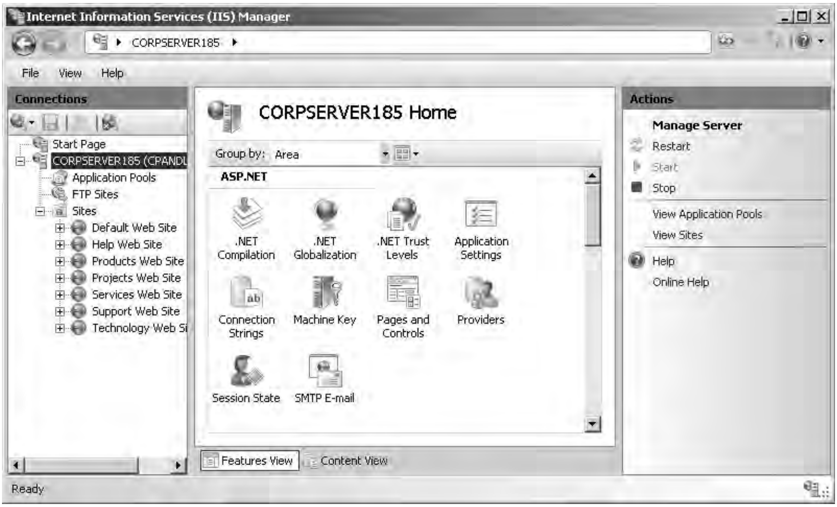


Figure 1-2 IIS Manager has a completely redesigned interface in IIS 7.0.

IIS 7.0 introduces the concept of delegated administration. With *delegated administration*, a machine administrator can delegate administrative control safely and securely. Delegated administration allows different levels of the configuration hierarchy to be managed by other users, such as site administrators or application developers. In a standard configuration, the default delegation state limits write access to most configuration settings to machine administrators only, and you must explicitly modify the delegation settings to grant write access to others. You'll learn more about IIS security and delegation in Chapter 10, "Managing Web Server Security."

IIS Manager and other graphical tools provide just about everything you need to work with IIS 7.0. Still, there are times when you might want to work from the command line, especially if you want to automate installation or administration tasks. To help you with all your command-line needs, IIS 7.0 includes the IIS command-line administration tool (AppCmd.exe). AppCmd.exe is located in the *%SystemRoot%*\System32\Inetsrv directory. By default, this directory is not in your command path. Because of this, you'll need either to add this directory to the default path or change to this directory each time you want to use this tool. Add this directory temporarily to your default path by typing the following at an elevated command prompt:

```
path %PATH%;%SystemRoot%\System32\inetsrv
```

Then add this directory permanently to your default path by typing the following at an elevated command prompt:

```
setx PATH %PATH%;%SystemRoot%\System32\inetsrv
```

> **Note** You use Path to temporarily update the command path for the current window. You use SETX PATH to permanently update the command path for future command windows.

Table 1-3 provides a summary of the core set of administration objects for the IIS command-line administration tool.

Table 1-3 Administration Objects for the IIS Command-Line Administration Tool

Object Type	Description	Related Commands
APP	Allows you to create and manage Web application settings by using related list, set, add, and delete commands	list, set, add, and delete
APPPOOL	Allows you to create and manage application pools by using related list, set, add, delete, start, stop, and recycle commands	list, set, add, delete, start, stop, and recycle
BACKUP	Allows you to create and manage backups of your server configuration by using list, add, delete, and restore commands	list, add, delete, and restore
CONFIG	Allows you to manage general configuration settings by using related list, set, search, lock, unlock, clear, reset, and migrate commands	list, set, search, lock, unlock, clear, reset, and migrate
MODULE	Allows you to manage IIS modules by using related list, set, add, delete, install, and uninstall commands	list, set, add, delete, install, and uninstall
REQUEST	Allows you to list current HTTP requests by using a related list command	list
SITE	Allows you to create and manage virtual sites by using related list, set, add, delete, start, and stop commands	list, set, add, delete, start, and stop
TRACE	Allows you to manage failed request tracing by using related list, configure, and inspect commands	list, configure, and inspect
VDIR	Allows you to create and manage virtual directory settings by using related list, set, add, and delete commands	list, set, add, and delete
WP	Allows you to list running worker processes by using a related list command	list

The basics of working with the IIS command-line administration tool are straightforward. Most administration objects support these basic commands:

- **ADD** Creates a new object with the properties you specify.
- **DELETE** Deletes the object you specify.
- **LIST** Displays a list of related objects. Optionally, you can specify a unique object to list, or you can type one or more parameters to match against object properties.
- **SET** Sets parameters on the object specified.

Some objects support other commands, including:

- **RECYCLE** Recycles the object you specify by deleting it and then re-creating it
- **START** Starts the object you specify if it is stopped
- **STOP** Stops the object you specify if it is started or otherwise active

To type commands, use the following basic syntax:

```
appcmd Command <Object-type>
```

where *Command* is the action to perform, such as list, add, or delete, and Object-type is the object on which you want to perform the action, such as app, site, or vdir. Following this, if you wanted to list the configured sites on a server, you could type the following command at an elevated command prompt:

```
appcmd list site
```

Because the IIS command-line administration tool will also accept plural forms of object names, such as apps, sites, or vdirs, you could also use:

```
appcmd list sites
```

In either case, the resulting output is a list of all configured sites on the server with their related properties, such as:

```
SITE "Default Web Site" (id:1,bindings:http/*:80:,state:Started)
```

You'll find a comprehensive discussion of using the IIS command-line administration tool in Chapter 4, "Managing IIS 7.0 from the Command Line." In addition, you will see examples of using this tool throughout the book.

Chapter 2

Deploying IIS 7.0 in the Enterprise

Before you deploy Internet Information Services (IIS) 7.0, you should carefully plan the machine and administration architecture. As part of your planning, you need to look closely at the protocols and roles IIS will use and modify both server hardware and technology infrastructure accordingly to meet the requirements of these roles on a per-machine basis. Your early success with IIS 7.0 will largely depend on your understanding of the ways you can use the software and in your ability to deploy it to support these roles.

IIS 7.0 Protocols

TCP/IP is a protocol suite consisting of Transmission Control Protocol (TCP) and Internet Protocol (IP). TCP/IP is required for internetwork communications and for accessing the Internet. Whereas TCP operates at the transport layer and is a connection-oriented protocol designed for reliable end-to-end communications, IP operates at the network layer and is an internetworking protocol used to route packets of data over a network.

IIS 7.0 uses protocols that build on TCP/IP, including:

- Hypertext Transfer Protocol (HTTP)
- Secure Sockets Layer (SSL)
- File Transfer Protocol (FTP)
- Simple Mail Transfer Protocol (SMTP)

HTTP and SSL

As you probably already know, HTTP is an application-layer protocol that makes it possible to publish static and dynamic content on a server so that it can be viewed in client applications, such as Microsoft Windows Internet Explorer. Publishing a Web document is a simple matter of making the document available in the appropriate directory on an HTTP server and assigning the appropriate permissions so that an HTTP client application can access the document. An HTTP session works like this:

1. The HTTP client application uses TCP to establish a connection to the HTTP server. The default (well-known) port used for HTTP connections is TCP port 80. You can configure servers to use other ports as well. For example, TCP port 8080 is a popular alternative to TCP port 80 for sites that are meant to have limited access.

2. After connecting to the server, the HTTP client application requests a Web page or other resource from the server. In the client application, users specify the pages or resources they want to access by using a Web address, otherwise known as a Uniform Resource Locator (URL).
3. The server responds to the request by sending the client the request resource and any other related files, such as images, that you've inserted into the requested resource. If you've enabled the HTTP Keep-Alive feature on the server, the TCP connection between the client and server remains open to speed up the transfer process for subsequent client requests. Otherwise, the TCP connection between the client and server is closed and the client must establish a new connection for subsequent transfer requests.

That in a nutshell is essentially how HTTP works. The protocol is meant to be simple yet dynamic, and it is the basis upon which the World Wide Web is built.

With HTTP, you can configure access to documents so that anyone can access a document or so that documents can be accessed only by authorized individuals. To allow anyone to access a document, you configure the document security so that clients can use Anonymous authentication. With Anonymous authentication, the HTTP server logs on the user automatically using a guest account, such as IUSR. To require authorization to access a document, configure the document security to require authentication using one of the available authentication mechanisms, such as Basic authentication, which requires a user to type a user name and password.

You can use Secure Sockets Layer (SSL) to enable Hypertext Transfer Protocol Secure (HTTPS) transfers. SSL is an Internet protocol used to encrypt authentication information and data transfers passed between HTTP clients and HTTP servers. With SSL, HTTP clients connect to Web pages using URLs that begin with *https://*. The *https* prefix tells the HTTP client to try to establish a connection using SSL. The default port used with secure connections is TCP port 443 rather than TCP port 80. See Chapter 10, "Managing Web Server Security," for more information on SSL.

FTP

FTP is an application-layer protocol that makes it possible for client applications to retrieve files from or transfer files to remote servers. FTP predates HTTP, and its usage is in decline as compared to HTTP. With FTP, you can publish a file so that a client can download it by making the file available in the appropriate directory on an FTP server and assigning the appropriate permissions so that an FTP client application can access the document. To upload a file to an FTP server, you must grant an FTP client application permission to log on to the server and access directories used for uploading files.

An FTP session works like this:

1. The FTP client application uses TCP to establish a connection to the FTP server. The default (well-known) port used for FTP connections is TCP port 21. FTP servers listen on this port for client connection requests. After the client and server establish a connection, the server randomly assigns the client a TCP port number above 1023. This initial TCP connection (with port 21 for the server and a random port for the client) is then used for transmission of FTP control information, such as commands sent from the client to the server and response codes returned by the server to the client.
2. The client then issues an FTP command to the server on TCP port 21. Standard FTP commands include GET for downloading a file, CD for changing directories, PUT for uploading files, and BIN for switching to binary mode.
3. When the client initiates a data transfer with the server, the server opens a second TCP connection with the client for the data transfer. This connection uses TCP port 20 on the server and a randomly assigned TCP port above 1023 on the client. After the data transfer is complete, the second connection goes in a wait state until the client initiates another data transfer or the connection times out.

That in a nutshell is how FTP works. As you can see, FTP is a bit clunkier than HTTP, but it is still fairly simple.

> **Real World** What sets FTP and HTTP apart is primarily the way you transfer files. FTP transfers files as either standard text or encoded binaries. HTTP has the capability to communicate the file format to the client, and this capability allows the client to determine how to handle the file. If the client can handle the file format directly, it renders the file for display. If the client has a configured helper application, such as with PDF documents, the client can call the helper application and let it render the file for display within the client window. The component that makes it possible for HTTP clients and servers to determine file format is their support for the Multipurpose Internet Mail Extensions (MIME) protocol. Using the MIME protocol, an HTTP server identifies each file with its corresponding MIME type. For example, an HTML document has the MIME type text/html, and a GIF image has the MIME type image/gif.

With FTP, you can allow anonymous downloads and uploads in addition to restricted downloads and uploads. To allow anyone to access a file, configure directory security so that clients can use Anonymous authentication. With Anonymous authentication, the FTP server logs the user on automatically using a guest account and allows the anonymous user to download or upload files as appropriate. To require authorization to log on and access a directory, configure directory security to require authentication using one of the available authentication mechanisms, such as Basic authentication, which requires a user to type a user name and password prior to logging on and downloading or uploading files.

SMTP

SMTP is an application-layer protocol that makes it possible for client applications to send e-mail messages to servers and for servers to send e-mail messages to other servers. A related protocol for retrieving messages from a server is Post Office Protocol version 3 (POP3). In IIS 6, full implementations of Simple Mail Transfer Protocol (SMTP) and Post Office Protocol version 3 (POP3) are included. IIS 7.0 does not include SMTP or POP3 services.

With IIS 7.0, a Web application can send e-mail on behalf of a user by using the SMTP E-mail component of Microsoft ASP.NET. An SMTP session initiated by a Web application works like this:

1. The Web application generates an e-mail message in response to something a user has done.
2. The System.Net.Mail API (a component of ASP.NET) delivers the email to an online SMTP server or stores the message on disk where it is stored for later delivery.
3. When sending mail to an SMTP server, the IIS server uses TCP port 25 to establish the connection. SMTP can be running on the local machine or on a different machine.

That is essentially how SMTP is used by Web applications. Microsoft doesn't provide other e-mail features as a part of IIS. However, a separate SMTP Server component is included as an optional feature that you can install on a computer running a Windows Server operating system.

IIS 7.0 Roles

You can deploy IIS on both desktop and server platforms. On desktop platforms, you can use IIS 7.0 for designing, building, and testing dynamic Web sites and Web applications. On server platforms, IIS 7.0 can have several different roles:

- **Application server** Application servers host distributed applications built using ASP.NET, Enterprise Services Network Support, and Microsoft .NET Framework 3.0. You can deploy application servers with or without Web Server (IIS) support. When you deploy an application server without Web Server (IIS) support, you configure application services through the application server core APIs and by adding or removing role services. Because the server lacks IIS configuration and administration components, you won't have any of the common IIS features and won't be able to configure the server by using IIS 7.0 modules, and you can't manage the server by using IIS 7.0 administration tools. To avoid these limitations, you should install the application server with Web Server (IIS) support. You'll then be able to use IIS features to better manage the application server installation.

- **Web server** Web servers use the services bundled in IIS 7.0 to host Web sites and Web applications. Web sites hosted on a Web server can have both static content and dynamic content. You can build Web applications hosted on a Web server by using ASP.NET and .NET Framework 3.0. When you deploy a Web Server, you can manage the server configuration by using IIS 7.0 modules and administration tools.
- **Microsoft Windows SharePoint Services server** Computers running Windows SharePoint Services enable team collaboration by connecting people and information. A SharePoint Services server is essentially a Web server running a full installation of IIS and using managed applications that provide the necessary collaboration functionality. When you deploy SharePoint Services, you can manage the server by using IIS 7.0 modules and administration tools in addition to several SharePoint-specific tools, including SharePoint Central Administration and the SharePoint Products and Technologies Configuration Wizard.

Table 2-1 organizes the 75 configuration features available for the three server roles into 14 general categories. Each entry for a particular configuration feature has one of the following values:

- **Available** Indicates a feature that is available for selection during installation. You can add available features as necessary to optimize the configuration of your server.
- **Default** Indicates a feature that is selected for installation by default. Although you may be able to deselect default features during setup, you should not do this in most cases because it could adversely affect the server performance or necessary core functionality.
- **Included** Indicates an included but unlisted feature that is part of the IIS server core. With application servers, these features are included only when you choose to install Web Server (IIS) support. With Web Server and SharePoint Services Server, these features are included automatically.
- **Not Installed** Indicates an available feature that is not installed as part of the standard setup. With Web and SharePoint Services servers, you can configure these features after installation by enabling the related modules. With application servers, these features are configurable after installation only when you choose to install Web Server (IIS) support or modify the role services associated with an installed Web server role.
- **Required** Indicates a feature that is required in order to install the server role. Setup selects required features automatically during installation.
- **N/A** Indicates a feature that is not applicable or available for a particular server role.

- **Web Common** Indicates a feature installed by default as part of the common Web Server (IIS) features of an application server.
- **WPASS Required** Indicates an application server feature required for Windows Process Activation Service Support.

Table 2-1 Configuration Features for Application and Web Servers and Computers Running SharePoint Services

Feature	Application Server	Web Server	SharePoint Services
.NET Framework 3.0			
.NET Framework 3.0	Required	Available	Required
Application Server Support			
Application Server Foundation	Default	N/A	N/A
COM+ Network Access	Available	N/A	N/A
TCP Port Sharing	WPASS Required	N/A	N/A
Web Server (IIS) Support	Available	N/A	N/A
Application Development Features			
.NET Extensibility	Web Common; WPASS Required	Available	Required
ASP	Available	Available	Available
ASP.NET	Web Common	Available	Required
CGI	Available	Available	Available
ISAPI Extensions	Web Common	Available	Required
ISAPI Filters	Web Common	Available	Required
Server-Side Includes	Available	Available	Available
Common HTTP Features			
Default Document	Web Common	Default	Required
Directory Browsing	Web Common	Default	Required
HTTP Errors	Web Common	Default	Required
HTTP Redirection	Web Common	Available	Available
Static Content	Web Common	Default	Required

Table 2-1 Configuration Features for Application and Web Servers and Computers Running SharePoint Services

Feature	Application Server	Web Server	SharePoint Services
Distributed Transaction Support			
Incoming Remote Transaction Support	Available	N/A	N/A
Outgoing Remote Transaction Support	Available	N/A	N/A
WS-Atomic Transaction Support	Available	N/A	N/A
Extended Features			
File Cache	Not Installed	Not Installed	Not Installed
Managed Engine	Not Installed	Not Installed	Not Installed
Token Cache	Not Installed	Not Installed	Not Installed
HTTP Trace	Not Installed	Not Installed	Not Installed
URI Cache	Not Installed	Not Installed	Not Installed
FTP Publishing Service			
FTP Management Console	Not Installed	Available	Not Installed
FTP Server	Not Installed	Available	Not Installed
Health and Diagnostics Features			
Custom Logging	Not Installed	Available	Not Installed
HTTP Logging	Web Common	Default	Required
Logging Tools	Web Common	Available	Required
ODBC Logging	Not Installed	Available	Not Installed
Request Monitor	Web Common	Default	Required
Tracing	Web Common	Available	Required
IIS Server Core			
Anonymous Authentication	Included	Included	Included
Configuration Validation	Included	Included	Included
HTTP Cache	Included	Included	Included
Protocol Support	Included	Included	Included

Table 2-1 Configuration Features for Application and Web Servers and Computers Running SharePoint Services

Feature	Application Server	Web Server	SharePoint Services
Performance Features			
Dynamic Content Compression	Web Common	Available	Required
Static Content Compression	Web Common	Default	Required
Security Features			
Basic Authentication	Web Common	Available	Required
Client Certificate Mapping Authentication	Web Common	Available	Available
Digest Authentication	Web Common	Available	Required
IIS Client Certificate Mapping Authentication	Web Common	Available	Available
IP and Domain Restrictions	Web Common	Available	Available
Request Filtering	Web Common; WPASS Required	Default	Available
URL Authorization	Web Common	Available	Available
Windows Authentication	Web Common	Available	Required
Web Management Tools			
IIS Management Console	Default	Default	Required
IIS Management Scripts and Tools	Web Common	Available	Not Installed
IIS Management Service	Web Common	Available	Not Installed
IIS 6 Management Compatibility	Not Installed	Available	Required
IIS Metabase Compatibility	Not Installed	Available	Required
IIS 6 WMI Compatibility	Not Installed	Available	Not Installed

Table 2-1 Configuration Features for Application and Web Servers and Computers Running SharePoint Services

Feature	Application Server	Web Server	SharePoint Services
IIS 6 Scripting Tools	Not Installed	Available	Not Installed
IIS 6 Management Console	Not Installed	Available	Not Installed
Windows Activation Service			
.NET Environment	Required	Available	Required
Configuration APIs	Required	Required	Required
Process Model	Required	Required	Required
Windows Process Activation Service Support			
HTTP Activation	WPASS Required	N/A	N/A
MSMQ Activation	WPASS Required	N/A	N/A
Named Pipes Activation	Available	N/A	N/A
TCP Activation	Available	N/A	N/A
Windows Process Activation Service Support (Additional)			
Message Queuing Server	WPASS Required	N/A	N/A
Non-HTTP Activation	WPASS Required	N/A	N/A
Windows SharePoint Services Support			
SharePoint Applications	N/A	N/A	Default
SharePoint Management Tools	N/A	N/A	Default

When configuring application servers, Web servers, and SharePoint Services, it is important to understand exactly what comprises the .NET Framework 3.0. The Microsoft .NET Framework 3.0 is a managed code programming model for Windows. It combines the power of the .NET Framework 2.0 with four new technologies:

- **Windows CardSpace (WCS)** A suite of .NET technologies for managing digital identities. Windows CardSpace supports any digital identity system and gives users consistent control of their digital identities. A digital identity can be as simple as an e-mail address and password used to log on to a Web site, or it can include a user's full contact and logon information. Client applications display each digital identity as an information card. Each card contains information about a particular digital identity, including what provider to contact to acquire a security token for the identity. By selecting a card and sending it to a provider such as Amazon or Yahoo!, users can validate their identity and log on to the service offered by the site.

- **Windows Communication Foundation (WCF)** A suite of .NET technologies for building and running connected systems. WCF supports a broad array of distributed systems capabilities to provide secure, reliable, and transacted messaging along with interoperability. Servers establish distributed communications through service endpoints. Service endpoints have an endpoint address, a binding that specifies how the endpoint can communicate, and a contract description that details what an endpoint communicates.
- **Windows Presentation Foundation (WPF)** A suite of .NET technologies for building applications with attractive and effective user interfaces. WPF supports tight integration of application user interfaces, documents, and media content, allowing developers to create a unified interface for all types of documents and media. This means that applications can use the same interface for displaying forms, controls, fixed-format documents, on-screen documents, 2D images, 3D images, video, and audio.
- **Windows Workflow Foundation (WF)** A suite of .NET technologies for building workflow-enabled applications on Windows. WF provides a rules engine that allows for the declarative modeling of units of application logic within the scope of an overall business process. What this means is that developers can use WF to model and implement the necessary programming logic for a business process from start to finish.

To support applications written for IIS 6, you can deploy IIS 7.0 with IIS 6 compatibility enabled. If you have existing IIS 6 server installations, you can also install the IIS 6 Management Compatibility tools to support remote administration of these server installations. You also can deploy IIS 7.0 to support remote administration. You can use both desktop and server platforms for remote administration of other IIS servers in addition to the sites and applications configured on these servers. For remote administration of an IIS server, you must enable the Web Management Service (WMSVC) on the server you want to manage remotely. Then install the Web management tools on the machine you want to use for remote administration.

Navigating the IIS 7.0 Role Services and Features

As discussed previously, you can deploy IIS 7.0 running on a computer running Windows Server 2008 to support three specific roles: application server, Web server, and Windows SharePoint Services server. You can deploy IIS 7.0 running on a Windows desktop to support designing, building, and testing sites and applications. The components used to support these roles are referred to as either role services or features, depending on which user interface you are working with. In the sections that follow, I discuss each of the server roles and the related role services.

Role Services for Application Servers

You use application servers running on Windows Server 2008 editions to host distributed applications built by using ASP.NET, Enterprise Services, and WCF. Figure 2-1 provides an overview of the related services for application servers.

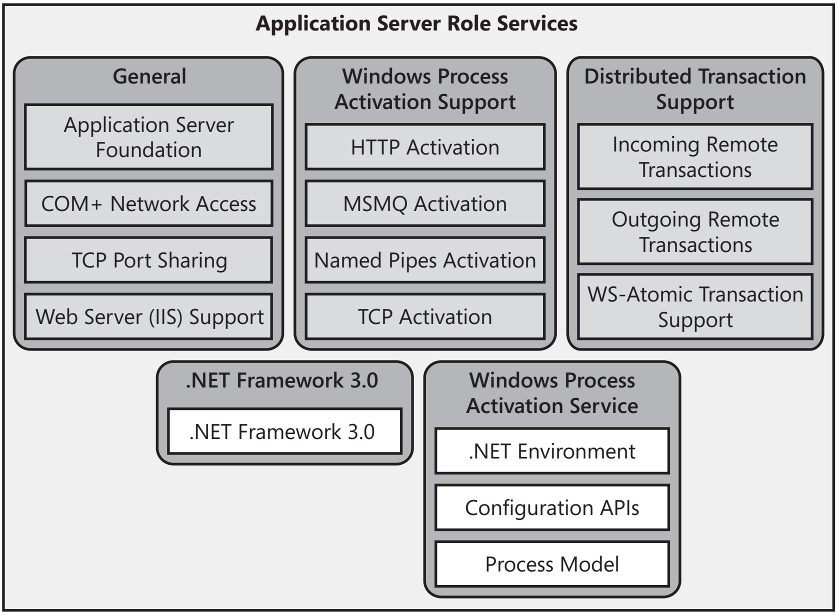

Application Server Role Services
General
Application Server Foundation
COM+ Network Access
TCP Port Sharing
Web Server (IIS) Support
Windows Process Activation Support
HTTP Activation
MSMQ Activation
Named Pipes Activation
TCP Activation
Distributed Transaction Support
Incoming Remote Transactions
Outgoing Remote Transactions
WS-Atomic Transaction Support
.NET Framework 3.0
.NET Framework 3.0
Windows Process Activation Service
.NET Environment
Configuration APIs
Process Model

Figure 2-1 Role services for application servers.

When you install an application server, only the Application Server Core and Enterprise Services Network Access services are included as standard core features. In addition to the standard core features, you must install the .NET Framework 3.0 components and the Windows Activation Service components. Other components are optional and should be installed based on the specific requirements of the distributed applications you are hosting.

Application servers can use the following general-purpose role services:

- **Application Server Foundation** Provides the core application server functionality through these .NET Framework 3.0 technologies: Windows CardSpace, WCF, WPF, and WF. These technologies allow you to deliver managed-code applications that model business processes.

- **COM+ Network Access** Enables application servers to invoke applications remotely over the network. Applications being invoked must have been built using Enterprise Services and provide support for hosting COM+ components.
- **TCP Port Sharing** Allows multiple applications to share a single TCP port. By using this feature, many Web applications can coexist on the same server in separate, isolated processes while sharing the network infrastructure required for sending and receiving data over TCP ports.
- **Web Server (IIS) Support** Allows the application server to host Web sites with both static and dynamic content. The Web sites support the standard IIS server extensions and allow you to create Web pages containing dynamic content. This allows an application server to host an internal or external Web site or provide an environment for developers to create Web applications. See Table 2-2 for a complete list of IIS features installed by default when you select this feature.

The Windows Process Activation Service supports distributed Web-based applications that use different protocols to transfer information. You can use the following related components:

- **.NET Environment** Installs the .NET Environment for use with managed code activation.
- **Configuration APIs** Installs the managed code APIs that allow you to configure the process model.
- **Process Model** Installs a process model for developing and running applications.

Windows Process Activation Service Support enables the application server to invoke applications remotely over a network by using protocols such as HTTP, Microsoft Message Queuing (MSMQ), named pipes, and TCP. This allows applications to start and stop dynamically in response to incoming requests, resulting in improved performance and enhanced manageability. To specify which protocols an application server can use with Windows Process Activation, you can use the following related role services:

- **HTTP Activation** Supports process activation over HTTP. This is the standard activation method used by most Web applications. Applications that support HTTP Activation can start and stop dynamically in response to requests that arrive via HTTP. With HTTP, the application and the computers with which it communicates need to be online to pass active communications back and forth without the need for queuing requests.
- **Message Queuing Activation** Supports process activation over Microsoft Message Queue (MSMQ). This activation method is used when the application server runs distributed messaging applications. Applications that support MSMQ Activation and message queuing can start and stop dynamically in

response to requests that arrive via MSMQ. With message queuing, source applications send messages to queues, where they are stored temporarily until target applications retrieve them. This queuing technique allows applications to communicate across different types of networks and with computers that may be offline.

- **Named Pipes Activation** Supports process activation over named pipes. Applications that support Named Pipes Activation can start and stop dynamically in response to requests that arrive via named pipes. You use this activation method when Web applications communicate with older versions of the Windows operating system. A *named pipe* is a portion of memory that one process can use to pass information to another process such that the output from one process is the input of the other process. Named pipes have standard network addresses such as \\.\Pipe\Sql\Query, which a process can reference on a local machine or a remote machine. The Named Pipes protocol is used primarily for local or remote connections by applications written for Microsoft Windows NT, Windows 98, and earlier versions of Windows.
- **TCP Activation** Supports process activation over TCP. Applications that support TCP Activation can start and stop dynamically in response to requests that arrive via TCP. With TCP, the application and the computers with which it communicates need to be online so they can pass active communications back and forth without the need for queuing requests.

When using Windows Process Activation Support, these additional roles services may be required:

- **Non-HTTP Activation** Provides non-HTTP activation support using any of the following: MSMQ, named pipes, and TCP. IIS installs this feature as a WCF Activation component.
- **Message Queuing Server** Provides the necessary server functions for message queuing.

> **Tip** Each of the Windows Process Activation Support features has a related set of required role services. With HTTP Activation, all the features listed as Web Common in Table 2-1 are required. With Message Queuing Activation, Message Queuing Server and Non-HTTP Activation are required. With TCP Activation and Named Pipes Activation, Non-HTTP Activation is required.

When applications communicate with each other, they may need to perform various types of transactions, such as queries to retrieve data stored in a database or a data submission to update data stored in a database. When the application server hosts the database or needs to query a single database to complete a transaction, transactions are fairly straightforward. Things get complex fast, though, when you are working with multiple databases hosted on multiple computers. A transaction that involves multiple databases hosted on multiple computers is referred to as a *distributed transaction*. With distributed transactions, you need a way to guarantee that all the data you need is either retrieved or submitted as appropriate, and this is where Distributed Transactions support comes into the picture. Distributed Transactions support provides services that help ensure that distributed transactions are successfully completed.

To enable Distributed Transactions support on an application server, you can use the following related role services:

- **Incoming Remote Transactions** Provides distributed transaction support to help ensure that incoming remote transactions are successfully completed
- **Outgoing Remote Transactions** Provides distributed transaction support to help ensure that outgoing remote transactions are successfully completed
- **WS-Atomic Transactions** Provides distributed transaction support for applications that use two-phase commit transactions with Simple Object Access Protocol (SOAP)–based exchanges. SOAP-based exchanges contain text-based commands that are formatted with XML. If you plan to use SOAP for two-phase commit transactions, you'll also need to set and configure HTTP endpoints.

> **Real World** WS-Atomic Transactions use SSL to encrypt network traffic when communicating with clients. To use SSL, you must install a server authentication certificate suitable for SSL encryption on the WS-AT site in IIS. If you obtain a certificate from a certificate authority (CA), you can import the certificate as part of the setup process. For small-scale and test environments, you also have the option of creating a self-signed certificate during setup. The drawback of this type of certificate is that you must install it manually on clients.

In your deployment planning, there is a distinct advantage to deploying an application server with Web Server support. When you deploy an application server with Web Server support, you can configure application services using the APIs provided by ASP.NET and the .NET Framework. Because the server includes IIS configuration and administration components, you'll have all of the common IIS features available and will be able to configure the server by using the IIS 7.0 modules and the IIS 7.0 administration tools.

Role Services for Windows Desktops and Web Servers

Web servers running on Windows Vista desktop editions or on Windows Server 2008 editions can host Web sites and Web applications. Figure 2-2 provides an overview of the related role services for Web servers.

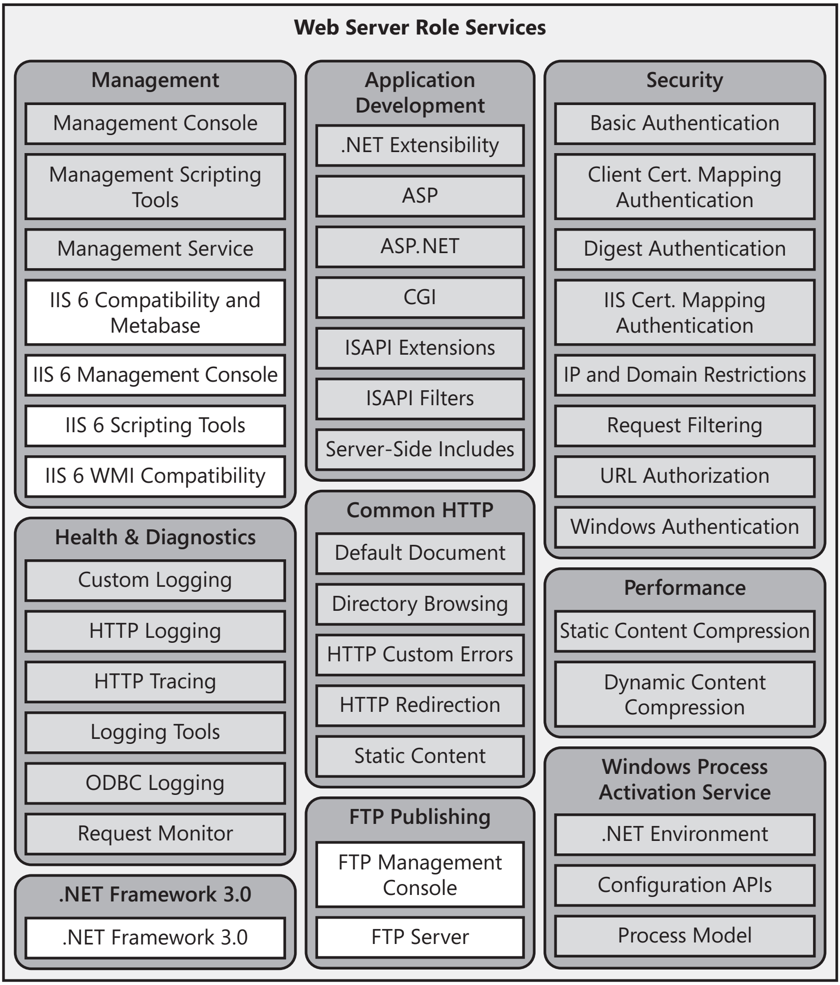


Figure 2-2 Role services for Web servers.

As summarized in Table 2-1, when you install a Web server, several configuration features are installed automatically as part of the server core, and other features are installed by default (if applicable for the operating system version you are using). These features represent core internal components in addition to the recommended minimum and required components for managing a Web server and publishing a Web site. In most installations of IIS 7.0, you will want to install additional features based on the specific requirements of the Web sites and Web applications the server is hosting.

As discussed in Chapter 1, "IIS 7.0 Administration Overview," Windows Server editions and Windows Vista editions have different sets of supported features. Table 2-2 provides a feature comparison based on Windows version and edition. The table also lists the related request limitations of Windows versions and editions. Because Windows Server editions have no request limitations, you can use them in live production environments. Because Windows Vista editions have severe request limitations, they are best suited for individual administrator or developer use and use in test and development environments.

Table 2-2 Feature Comparison Based on Windows Version and Edition

Feature	Windows Server 2008	Windows Vista Business & Ultimate	Windows Vista Home Premium	Windows Vista Home Basic
IIS Server Core				
Anonymous Authentication	Included	Included	Included	N/A
Configuration Validation	Included	Included	Included	N/A
HTTP Cache	Included	Included	Included	N/A
Protocol Support	Included	Included	Included	N/A
Common HTTP Features				
Default Document	Default	Default	Default	N/A
Directory Browsing	Default	Default	Default	N/A
HTTP Errors	Default	Default	Default	Default
HTTP Redirection	Available	Available	Available	Available
Static Content	Default	Default	Default	N/A
Application Development Features				
.NET Extensibility	Available	Available	Available	Available
ASP	Available	Available	Available	N/A
ASP.NET	Available	Available	Available	N/A
CGI	Available	Available	Available	N/A
ISAPI Extensions	Available	Available	Available	N/A
ISAPI Filters	Available	Available	Available	N/A

Table 2-2 Feature Comparison Based on Windows Version and Edition

Feature	Windows Server 2008	Windows Vista Business & Ultimate	Windows Vista Home Premium	Windows Vista Home Basic
Server-Side Includes	Available	Available	Available	N/A
Health and Diagnostics Features				
Custom Logging	Available	Available	Available	N/A
HTTP Logging	Default	Default	Default	Default
Logging Tools	Available	Available	Available	Available
ODBC Logging	Available	Available	N/A	N/A
Request Monitor	Default	Default	Default	Default
Tracing	Available	Available	Available	Available
Security Features				
Basic Authentication	Available	Available	Available	N/A
Client Certificate Mapping Authentication	Available	Available	N/A	N/A
Digest Authentication	Available	Available	N/A	N/A
IIS Client Certificate Mapping Authentication	Available	Available	N/A	N/A
IP and Domain Restrictions	Available	Available	Available	Available
Request Filtering	Default	Available	Available	Available
URL Authorization	Available	Available	Available	Available
Windows Authentication	Available	Available	N/A	N/A
Performance Features				
Static Content Compression	Default	Default	Default	N/A

Table 2-2 Feature Comparison Based on Windows Version and Edition

Feature	Windows Server 2008	Windows Vista Business & Ultimate	Windows Vista Home Premium	Windows Vista Home Basic
Dynamic Content Compression	Available	Available	Available	Available
Web Management Tools				
IIS Management Console	Default	Default	Default	N/A
IIS Management Scripts and Tools	Available	Available	Available	Available
IIS Management Service	Available	Available	Available	N/A
IIS 6 Management Compatibility	Available	Available	Available	Available
IIS Metabase compatibility	Available	Available	Available	Available
IIS 6 WMI Compatibility	Available	Available	Available	N/A
IIS 6 Scripting Tools	Available	Available	Available	N/A
IIS 6 Management Console	Available	Available	Available	N/A
FTP Publishing Service				
FTP Management Console	Available	Available	N/A	N/A
FTP Server	Available	Available	N/A	N/A
Windows Activation Service				
.NET Environment	Available	Available	Available	Available
Configuration APIs	Default	Available	Available	Available
Process Model	Default	Default	Default	Default
Limitations				
Request Execution Limit	Unlimited	10	3	3

As the table shows, many different features are available with Web servers. I'll discuss each of the features I haven't previously discussed in this section, and you'll also find detailed information on these features in appropriate chapters throughout this book. In the appendix, "Comprehensive IIS 7.0 Module and Schema Reference," you'll also find a detailed description of features with related configuration modules.

The IIS Server Core features provide the foundation functions for IIS. You can use these features as follows:

- **Anonymous Authentication** Supports anonymous access to a server. With anonymous access, any user can access content without having to provide credentials. Each server has to have at least one authentication mechanism configured, and this is the default mechanism.
- **Configuration Validation** Validates the configuration of a server and its applications. If someone improperly configures a server or application, IIS 7.0 generates errors that can help detect and diagnose the problem.
- **HTTP Cache** Improves performance by returning a processed copy of a requested Web page from cache, resulting in reduced overhead on the server and faster response times. IIS 7.0 supports several levels of caching including output caching in user mode and output caching in kernel mode. When you enable kernel-mode caching, cached responses are served from the kernel rather than from IIS user mode, giving IIS an extra boost in performance and increasing the number of requests IIS can process.
- **Protocol Support** Provides support for common protocols used by Web servers, including HTTP keep-alives, custom headers, and redirect headers. *HTTP keep-alives* allows clients to maintain open connections with servers, which speeds up the request process once a client has established a connection with a server. *Custom headers* and *redirect headers* allow you to optimize the way IIS works to support advanced features of the HTTP 1.1 specification.

The Common HTTP features install the common services required for serving Web content. You can use these features as follows:

- **Default Document** Supports displaying of default documents. When you've enabled this feature and a user enters a request with a trailing '/,' such as http://www.adatum.com/, IIS can redirect the request to the default document for the Web server or directory. For best performance, you should list the default document you use the most first and reduce the overall list of default documents to only those necessary.
- **Directory Browsing** Supports directory browsing functionality. When you've enabled default documents but there is no current default document, IIS can use this feature to generate a listing of the contents of the specified directory. If you haven't enabled the default document or directory browsing features, and a client requests a directory-level URL, IIS returns an empty response.

- **HTTP Errors** Supports custom error and detailed error notification. When you enable this feature and the server encounters an error, the server can return a customer error page to all clients regardless of location, a detailed error message to all clients regardless of location, or a detailed error for local clients and a custom error page for remote clients. IIS displays a custom error page based on the type of HTTP error that occurred.
- **HTTP Redirection** Supports redirection of HTTP requests to send users from an old site to a new site. In the default configuration for redirection, all requests for files in the old location are mapped automatically to files in the new location you specify. You can customize this behavior in several ways.
- **Static Content** Supports static Web content, such as HTML documents and GIF or JPEG images. The staticContent/mimeMap configuration collection in the applicationHost.config file determines the list of file extensions supported.

Note Each of these common features has a related IIS 7.0 native module that Setup installs and activates when you select the feature. For the exact mapping of common features to their corresponding native modules, see the appendix. You'll learn more about working with these features in Chapter 5, "Managing Global IIS Configuration."

The Application Development features install the features required for developing and hosting Web applications. You can use these features as follows:

- **.NET Extensibility** Enables a Web server to host .NET Framework applications and provides the necessary functionality for IIS integration with ASP.NET and the .NET Framework. When you are working with managed modules, you must also enable the Managed Engine. The *Managed Engine* is the actual server component that performs the integration functions.
- **ASP** Enables a Web server to host classic Active Server Pages (ASP) applications. Web pages that use ASP are considered to be dynamic because IIS generates them at request time. To use ASP, you must also use ISAPI Extensions.
- **ASP.NET** Enables a Web server to host ASP.NET applications. Web pages that use ASP.NET are considered to be dynamic because they are generated at request time. To use ASP.NET, you must also use .NET Extensibility, ISAPI Extensions and ISAPI Filters.
- **CGI** Enables a Web server to host Common Gateway Interface (CGI) executables. CGI describes how executables specified in Web addresses, also known as *gateway scripts*, pass information to Web servers. By default, IIS handles all files with the .exe extension as CGI scripts.
- **ISAPI Extensions** Allows ISAPI Extensions to handle client requests. In the IIS server core, several components rely on handlers that are based on ISAPI Extensions, including ASP and ASP.NET. By default, IIS handles all files with the .dll extension as ISAPI Extensions.

- **ISAPI Filters** Allows ISAPI Filters to modify Web server behavior. IIS uses ISAPI Filters to provide additional functionality. When you select ASP.NET as part of the initial setup, Setup configures an ASP.NET filter to provide this functionality. In applicationHost.config, each version of ASP.NET installed on the Web server must have a filter definition that identifies the version and path to the related filter.
- **Server-Side Includes** Allows a Web server to parse files with Server-Side Includes (SSI). SSI is a technology that allows IIS to insert data into a document when a client requests it. When this feature is enabled, files with the .stm, .shtm, and .shtml extension are parsed to see if they have includes that should be substituted for actual values. If this feature is disabled, IIS handles .stm, .shtm, and .shtml files as static content, resulting in the actual include command being returned in the request.

Health and Diagnostics features enable you to monitor your servers, sites, and applications and to diagnose problems if they occur. You can use these features as follows:

- **Custom Logging** Enables support for custom logging. Typically, custom logging uses the ILogPlugin interface of the Component Object Model (COM). Rather than using this feature, Microsoft recommends that you create a managed module and subscribe to the RQ_LOG_REQUEST notification.
- **HTTP Logging** Enables support for logging Web site activity. You can configure IIS 7.0 to use one log file per server or one log file per site. Use per-server logging when you want all Web sites running on a server to write log data to a single log file. Use per-site logging when you want to track access separately for each site on a server.
- **Logging Tools** Allows you to manage server activity logs and automate common logging tasks using scripts.
- **ODBC Logging** Enables support for logging Web site activity to ODBC-compliant databases. In IIS 7.0, ODBC logging is implemented as a type of custom logging.
- **Request Monitor** Allows you to view details on currently executing requests, the run state of a Web site or the currently executing application domains, and more.
- **Tracing** Supports tracing of failed requests. Another type of tracing that you can enable after configuration is HTTP tracing, which allows you to trace events and warnings to their sources through the IIS server core.

Security features make it possible to control access to a server and its content. You can use these features as follows:

- **Basic Authentication** Requires a user to provide a valid user name and password to access content. All browsers support this authentication mechanism, but they transmit the password without encryption, making it possible for a malicious individual to intercept the password as the browser is transmitting it. If you want to require Basic Authentication for a site or directory, you should disable Anonymous Authentication for the site or directory.

- **Client Certificate Mapping Authentication** Maps client certificates to Active Directory accounts for the purposes of authentication. When you enable certificate mapping, this feature performs the necessary Active Directory certificate mapping for authentication of authorized clients.
- **Digest Authentication** Uses a Windows domain controller to authenticate user requests for content. Digest Authentication can be used through firewalls and proxies.
- **IIS Client Certificate Mapping Authentication** Maps SSL client certificates to a Windows account for authentication. With this method of authentication, user credentials and mapping rules are stored within the IIS configuration store.
- **IP and Domain Restrictions** Allows you to grant or deny access to a server by IP address, network ID, or domain. Granting access allows a computer to make requests for resources but doesn't necessarily allow users to work with resources. If you require authentication, users still need to authenticate themselves. Denying access to resources prevents a computer from accessing those resources, meaning that denied users can't access resources even if they could have authenticated themselves.
- **Request Filtering** Allows you to reject suspicious requests by scanning URLs sent to a server and filtering out unwanted requests. By default, IIS blocks requests for file extensions that could be misused and also blocks browsing of critical code segments.
- **URL Authorization** Supports authorization based on configuration rules. This allows you to require logon and to allow or deny access to specific URLs based on user names, .NET roles, and HTTP request method.
- **Windows Authentication** Supports Windows-based authentication using NTLM, Kerberos, or both. You'll use Windows Authentication primarily in internal networks.

For enhancing performance, IIS supports both static compression and dynamic compression. With static compression, IIS performs an in-memory compression of static content upon first request and then saves the compressed results to disk for subsequent use. With dynamic content, IIS performs in-memory compression every time a client requests dynamic content. IIS must compress dynamic content every time it is requested because dynamic content changes.

When you are trying to improve server performance and interoperability, don't overlook the value of these extended features:

- **File Cache** Caches file handles for files opened by the server engine and related server modules. If IIS does not cache file handles, IIS has to open the files for every request, which can result in performance loss.
- **Managed Engine** Enables IIS integration with the ASP.NET runtime engine. When you do not configure this feature, ASP.NET integration also is disabled, and no managed modules or ASP.NET handlers will be called when pooled applications run in Integrated mode.
- **Token Cache** Caches Windows security tokens for password based authentication schemes, including Anonymous Authentication, Basic Authentication, and Digest Authentication. Once IIS has cached a user's security token, IIS can use the cached security token for subsequent requests by that user. If you disable or remove this feature, a user must be logged on for every request, which can result in multiple logon user calls that could substantially reduce overall performance.
- **HTTP Trace** Supports request tracing for whenever a client requests one of the traced URLs. The way IIS handles tracing for a particular file is determined by the trace rules that you create.
- **URI Cache** Caches the Uniform Resource Identifier (URI)–specific server state, such as configuration details. When you enable this feature, the server will read configuration information only for the first request for a particular URI. For subsequent requests, the server will use the cached information if the configuration does not change.

You use Web management tools for administration and can divide the available tools into two general categories: those required for managing IIS 7.0 and those required for backward compatibility with IIS 6. You can use the related setup features as follows:

- **IIS Management Console** Installs the Internet Information Services (IIS) Manager, the primary management tool for working with IIS 7.0.
- **IIS Management Scripts and Tools** Installs the IIS command line administration tool and related features for managing Web servers from the command prompt.
- **IIS Management Service** Installs the Web Management Service (WMSVC), which provides a hostable Web core that acts as a standalone Web server for remote administration.
- **IIS Metabase Compatibility** Provides the necessary functionality for backward compatibility with servers running IIS 6 Web sites by installing a component that translates IIS 6 metabase changes to the IIS 7.0 configuration store.

- **IIS 6 WMI Compatibility** Provides the necessary functionality for scripting servers running IIS 6 Web sites by installing the IIS 6 Windows Management Instrumentation (WMI) scripting interfaces.
- **IIS 6 Scripting Tools** Provides the necessary functionality for scripting servers running IIS 6 Web sites by installing the IIS 6 Scripting Tools.
- **IIS 6 Management Console** Installs the Internet Information Services (IIS) 6.0 Manager, which is required to remotely manage servers running IIS 6 sites and to manage FTP servers for IIS 6.

Role Services for Servers Running SharePoint Services

You use servers running Windows SharePoint Services to enable team collaboration by connecting people and information. A server running SharePoint Services is essentially a Web server running a full installation of IIS and using managed applications that provide the necessary collaboration functionality. When you deploy SharePoint Services on a server, you can manage the server by using IIS 7.0 modules and administration tools and several SharePoint-specific tools, including SharePoint Central Administration and the SharePoint Products And Technologies Configuration Wizard. After installation, both management tools will be available on the Administrative Tools menu.

On a SharePoint site, you can host lists and libraries. A *list* is a collection of information on a site that you share with team members, including announcements, contacts, discussion boards, tasks, and team calendars. A *library* is a location on a site where you can create, store, and manage the files used by a team. SharePoint sites can host Web pages in addition to lists and libraries, and your Web pages can use static content, dynamic content, or both.

In your deployment planning for servers running SharePoint Services, you must consider several additional issues including the additional security and connectivity requirements that may be necessary for team collaboration. You'll want to ensure that you carefully protect access to a server running SharePoint Services. You'll also want to ensure that team members can access the server from remote locations as appropriate for the potential sensitivity of the information they are sharing.

As part of your planning, you'll need to consider the additional workload produced by SharePoint applications running on the server in addition to resources used by user connections. Windows SharePoint Services has a number of standard applications that run on a server running SharePoint Services, and these applications place an additional burden on the server's physical resources. Each user connection to a server will place an additional workload on the server, as will the requests and modifications users make.

Setting Up IIS 7.0

The way you set up IIS 7.0 depends on the role and operating system you are using. As discussed previously, you can configure IIS 7.0 to support one of three server roles: application server, Web server, and server running SharePoint Services. You can also configure IIS 7.0 as part of a desktop installation. I discuss deploying IIS 7.0 in each of these situations in the sections that follow.

Installing Application Servers

You can install an application server with or without Web server support by following these steps:

1. Start Server Manager by clicking the Server Manager icon on the Quick Launch toolbar or by clicking Start, Administrative Tools, Server Manager.
2. In Server Manager, select the Roles node in the left pane, and then, under Roles Summary, click Add Roles. This starts the Add Roles Wizard. If the wizard displays the Before You Begin page, read the Welcome page, and then click Next. You can avoid seeing the Welcome page the next time you start this wizard by selecting the Do Not Show Me This Page Again check box before clicking Next.
3. On the Select Server Roles page, select the Application Server role. You'll then see the Add Features Required For Application Server dialog box. This dialog box lists the features that are required in order to install an application server. Click Add Required Features to close the dialog box and add the .NET Framework 3.0 components and the Windows Process Activation Service components to the application server installation.
4. When you are deploying an application server with Web Server support, you can elect to accept the default common Web features or configure the exact features you'd like to use. If you have not installed Web Server (IIS) components previously and want to select the Web server (IIS) components for installation, select Web Server (IIS), and then click Next twice. Otherwise, just click Next twice to continue.
5. You should now see the Select Role Services page. If not previously installed, select Web Server (IIS) Support to install the application server with Web server support in the standard default configuration. You'll then see a dialog box listing the additional required roles. After you review the required roles, click Add Required Role Services to close the dialog box.

> **Note** The required roles are the same as those listed in Table 2-1 as Web Common. I recommend selecting Web Server (IIS) Support if the application server will host Web sites or Web services. This will ensure that Setup selects the required Web Common features by default, and this will be helpful later in the setup process.

6. Select other role services to install as appropriate, and then click Next. If you select a role service with additional required features, you'll see a dialog box listing the additional required roles. After you review the required roles, click Add Required Role Services to close the dialog box.
7. If you selected the WS-Atomic Transactions feature, you'll see the Choose A Certificate For SSL Encryption page next. You have the following options:
   - ❑ **Choose An Existing Certificate For SSL Encryption** Select this option if you previously obtained a certificate from a certification authority (CA) and want to install it for use with the WS-AT site that Setup will configure on the server. If you've previously imported certificates using the Certificate snap-in or the Import Certificate Wizard, you'll see a list of available certificates, and you can click the certificate you want to use. Otherwise, click Import to start the Certificate Import Wizard, and then follow the prompts to import the certificate.
   - ❑ **Create A Self-Signed Certificate For SSL Encryption** Select this option if you are using WS-Atomic transactions with a limited number of clients or for testing/development purposes and want to create and then automatically install a self-signing certificate for use with the WS-AT site that Setup will configure on the server. You will need to install the same certificate manually on all clients that need to be able to authenticate with the server.
   - ❑ **Choose A Certificate For SSL Encryption Later** Select this option if you haven't obtained a certificate from a CA yet but plan to later. When you choose this option, IIS disables SSL on the WS-AT site until you import the certificate, as discussed in Chapter 10.
8. If you selected Web Server (IIS) on the Select Server Roles page, as discussed in Step 5, click Next twice to display the Select Role Services page for Web server features. You can then select the Web server features to install. In most cases, you'll want to select additional features rather than trying to remove features. When selecting or clearing role services, keep the following in mind before you click Next to continue:
   - ❑ If you select a role service with additional required features, you'll see a dialog box listing the additional required roles. After you review the required roles, click Add Required Role Services to accept the additions and close the dialog box. If you click Cancel instead, Setup clears the feature you previously selected.
   - ❑ If you try to remove a role service that is required based on a previous selection, you'll see a warning prompt about dependent services that Setup must also remove. In most cases, you'll want to click Cancel to preserve the previous selection. If you click Remove Dependent Role Services, Setup

will remove the previously selected dependent services, which could cause the Web server to not function as expected

9. Click Next. On the Confirm Installation Selections page, click the Print, E-mail, Or Save This Information link to generate an installation report and display it in Windows Internet Explorer. You can then use standard Windows Internet Explorer features to print or save the report. After you've reviewed the installation options and saved them as necessary, click Install to begin the installation process.

10. When Setup finishes installing the application server with the features you've selected, you'll see the Installation Results page. Review the installation details to ensure that all phases of the installation completed successfully. If any portion of the installation failed, note the reason for the failure, and then use these troubleshooting techniques:

    a. Click the Print, E-mail, Or Save The Installation Report link to create or update the installation report and display it in Windows Internet Explorer.

    b. Scroll down to the bottom of the installation report in Windows Internet Explorer, and then click Full Log (For Troubleshooting Only) to display the Server Manager log in Notepad.

    c. In Notepad, press Ctrl+F, type the current date in the appropriate format for your language settings, such as 2007-08-30, and then click Find Next. Notepad will then move through the log to the first Setup entry from the current date.

    d. Review the Server Manager entries for installation problems, and take corrective actions as appropriate.

Installing Web Servers

You can install a Web server by following these steps:

1. Start the Server Manager by clicking the Server Manager icon on the Quick Launch toolbar or by clicking Start, Administrative Tools, Server Manager.

2. In Server Manager, select the Roles node in the left pane and then, under Roles Summary, click Add Roles. This starts the Add Roles Wizard. If the wizard displays the Before You Begin page, read the Welcome page, and then click Next. You can avoid seeing the Welcome page the next time you start this wizard by selecting the Do Not Show Me This Page Again check box before clicking Next.

3. On the Select Server Roles page, select the Web Server (IIS) role. You'll then see the Add Features Required For Web Server dialog box. This dialog box lists the features that are required to install a Web server. Click Add Required Features to close the dialog box and add the Windows Activation Service components to the Web server installation. Click Next twice to continue.

4. On the Select Role Services page, Setup selects the core set of standard features by default. When selecting or clearing role services, keep the following in mind before you click Next to continue:

   ❑ If you select a role service with additional required features, you'll see a dialog box listing the additional required roles. After you review the required roles, click Add Required Role Services to accept the additions and close the dialog box. If you click Cancel instead, Setup will clear the feature you previously selected.

   ❑ If you try to remove a role service that is required based on a previous selection, you'll see a warning prompt about dependent services that Setup must also remove. In most cases, you'll want to click Cancel to preserve the previous selection. If you click Remove Dependent Role Services, Setup will also remove the previously selected dependent services, which could cause the Web server to not function as expected.

5. Click Next. On the Confirm Installation Options page, click the Print, E-mail, Or Save This Information link to generate an installation report and display it in Windows Internet Explorer. You can then use standard Windows Internet Explorer features to print or save the report. After you've reviewed the installation options and saved them as necessary, click Install to begin the installation process.

6. When Setup finishes installing the application server with the features you've selected, you'll see the Installation Results page. Review the installation details to ensure that all phases of the installation completed successfully. If any portion of the installation failed, note the reason for the failure and then use these troubleshooting techniques:

   a. Click the Print, E-mail, Or Save The Installation Report link to create or update the installation report and display it in Windows Internet Explorer.

   b. Scroll down to the bottom of the installation report in Windows Internet Explorer and then click Full Log (For Troubleshooting Only) to display the Server Manager log in Notepad.

   c. In Notepad, press Ctrl+F, type the current date in the appropriate format for your language settings, such as 2007-08-30, and then click Find Next. Notepad will then move through the log to the first Setup entry from the current date.

   d. Review the Server Manager entries for installation problems and take corrective actions as appropriate.

Installing Windows SharePoint Services

Windows SharePoint Services uses one of two distinctly different configurations: independent server configuration and dependent load-balanced configuration. With an independent server configuration, you install Windows SharePoint Services on a single server that has its own database for storing application and user information. With a dependent load-balanced configuration, you install SharePoint Services on a computer as part of a Web farm where all servers share a Microsoft SQL Server 2000 or SQL Server 2005 database. Although both types of installations are configured using a similar initial setup process, if you want to connect to the SQL Server database and use load balancing, you must configure a server that is part of a Web farm.

> **Note** Windows SharePoint Services 2008 is a supplement to the Windows Server 2008 operating system. As such, Windows SharePoint Services 2008 is not included in Windows Server 2008 and must be installed separately. Once you've downloaded the installer packages from Microsoft and double-clicked each one to install it, you can configure this role using Server Manager, as discussed in this section. However, because SharePoint is a supplement, the wizard pages and related setup options may be different.

You can install Windows SharePoint Services on a computer by following these steps:

1. Start Server Manager by clicking the Server Manager icon on the Quick Launch toolbar or by clicking Start, Administrative Tools, Server Manager.
2. In Server Manager, select the Roles node in the left pane, and then, under Roles Summary, click Add Roles. This starts the Add Roles Wizard. If Setup displays the Before You Begin page, read the Welcome page and then click Next. You can avoid seeing the Welcome page the next time you start this wizard by selecting the Do Not Show Me This Page Again check box before clicking Next.
3. On the Select Server Roles page, select the Windows SharePoint Services role. You'll then see the Add Role Services And Features Required For Windows SharePoint Services dialog box. As listed previously in Table 2-1, this dialog box lists the features that are required in order to install SharePoint Services. Click Add Required Features to close the dialog box and add the Web Server (IIS), Windows Activation Service, and .NET Framework 3.0 components to the SharePoint installation. Click Next.
4. Read the introduction to Windows SharePoint Services. As necessary, click the links provided to learn more about the features offered with Windows SharePoint Services. Click Next when you are ready to continue.
5. On the Select Configuration Type page, choose the type of installation. If you are deploying a single-server solution, select Install Only On This Server and then click Next. If you are deploying a server that is part of a Web farm, select Install As Part Of A Server Farm, and then click Next.

6. Although individual SharePoint sites can use different languages, the administration site for Windows SharePoint Services can use only the language chosen during Setup, and you cannot change this language later. On the Select The Language For The Administration Site page, use the selection drop-down list provided to choose the desired language for the administration site, such as English, German, or Korean, and then click Next.
7. If you are installing a single-server configuration of Windows SharePoint Services, on the Specify E-mail Settings page, configure the default e-mail settings that SharePoint will use to send e-mail notifications to administrators. You can use the options provided as follows:
   - **Outbound SMTP Server** Sets the fully qualified domain name of the e-mail server that will send notifications to administrators, such as mail.adatum.com.
   - **From E-mail Address** Sets the e-mail address that will appear in the From field of notification messages, such as wss-admin@adatum.com.
   - **Reply-To E-mail Address** Sets the reply-to e-mail address for notification messages, such as wss-incoming@adatum.com.
8. If you have not previously installed Web Server (IIS), click Next twice to display the Select Role Services page for Web server features. You can then select the Web server features to install. In most cases, you'll want to select additional features rather than trying to remove features. When selecting or clearing role services, keep the following in mind before you click Next to continue:
   - If you select a role service with additional required features, you'll see a dialog box listing the additional required roles. After you review the required roles, click Add Required Role Services to accept the additions and close the dialog box. If you click Cancel instead, Setup will clear the feature you previously selected.
   - If you try to remove a role service that is required based on a previous selection, you'll see a warning prompt about dependent services that Setup must also remove. In most cases, you'll want to click Cancel to preserve the previous selection. If you click Remove Dependent Role Services, Setup will also remove the previously selected dependent services, which could cause the Web server to not function as expected.
9. On the Confirm Installation Selections page, click the Print, E-mail, Or Save This Information link to generate an installation report and display it in Windows Internet Explorer. You can then use standard Windows Internet Explorer features to print or save the report. After you've reviewed the installation options and saved them as necessary, click Install to begin the installation process.

10. If you are setting up a server that is part of a Web farm, you must configure a connection to the shared SQL Server database and perform other preliminary setup tasks by using the Windows SharePoint Services Central Administration tool.
11. When Setup finishes installing the application server with the features you've selected, you'll see the Installation Results page. Review the installation details to ensure that all phases of the installation completed successfully. If any portion of the installation failed, note the reason for the failure and then use these troubleshooting techniques:
    a. Click the Print, E-mail, Or Save The Installation Report link to create or update the installation report and display it in Windows Internet Explorer.
    b. Scroll down to the bottom of the installation report in Windows Internet Explorer and then click Full Log (For Troubleshooting Only) to display the Server Manager log in Notepad.
    c. In Notepad, press Ctrl+F, type the current date in the appropriate format for your language settings, such as 2007-08-30, and then click Find Next. Notepad will then move through the log to the first Setup entry from the current date.
    d. Review the Server Manager entries for installation problems and take corrective actions as appropriate.

Adding or Removing Web Server Features on Windows Vista

In earlier versions of Windows, you use Add/Remove Windows Components in the Add or Remove Programs application to add or remove operating system components. In Windows Vista, you configure operating system components as Windows features that you can turn on or off rather than add or remove.

You can configure Web server features on a computer running Windows Vista by completing these steps:

1. Click Start, and then click Control Panel.
2. In Control Panel, click Programs.
3. Under Programs And Features, click Turn Windows Features On Or Off. This displays the Windows Features dialog box.
4. You'll find Windows features for Web servers under the following nodes:
    - **Internet Information Services/FTP Publishing Service** Includes the FTP Management Console and the FTP Server
    - **Internet Information Services/Web Management Tools** Includes the IIS 6 Management and IIS 7.0 Management components

- ❑ **Internet Information Services/World Wide Web Services** Includes the Application Development, Common HTTP, Health and Diagnostics, Performance, and Security features
- ❑ **Microsoft .NET Framework 3.0** Includes the XPS View and the HTTP Activation and Non-HTTP Activation components for WCF
- ❑ **Microsoft Message Queue (MSMQ) Server** Includes the MSMQ Core server components in addition to support and integration components for message queuing
- ❑ **Windows Process Activation Service** Includes the .NET Environment, Configuration APIs, and Process Model

To turn features on, select feature check boxes. To turn features off, clear feature check boxes. As you select features, Windows Vista selects any required related features automatically without a warning prompt.

5. When you click OK, Windows Vista reconfigures components as appropriate for any changes you've made. You may need your original installation media.

Managing Installed Roles and Role Services

When you are working with Web and application servers and servers running SharePoint Services, Server Manager is the primary tool you'll use to manage roles and role services. Not only can you use Server Manager to add or remove roles and role services, you can also use Server Manager to view the configuration details and status for roles and roles services.

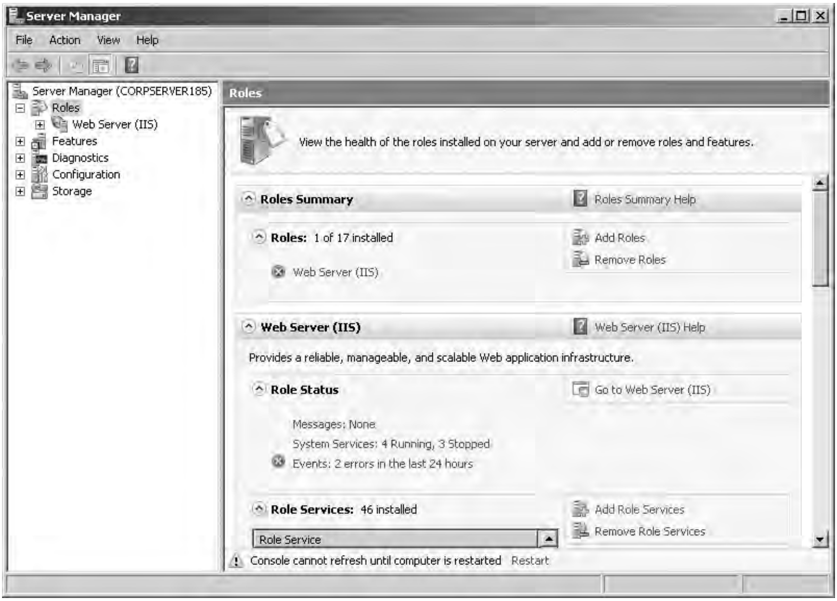
Viewing Configured Roles and Role Services

On Windows Server, Server Manager lists roles you've installed when you select the Roles node in the left pane. As Figure 2-3 shows, the main view of the Roles node displays a Roles Summary section that lists the number of roles and the names of the roles installed. When there are error-related events for a particular server role, Server Manager displays a warning icon to the left of the role name.

In the Roles window, the name of the role is a clickable link that accesses the related role details. The role details provide the following:

- Summary information about the status of related system services. If applicable, Server Manager lists the number of related services that are running or stopped, such as "System Services: 3 Running, 2 Stopped."
- Summary information about events the related services and components have generated in the last 24 hours, including details on whether any errors have occurred, such as "Events: 1 error(s), 6 warning(s), 2 informational in the last 24 hours."

- Summary information about the role services installed including the number of role services installed and the installed or not installed status of each individual role service that you can use with the role.



View the health of the roles installed on your server and add or remove roles and features.

Roles: 1 of 17 installed

Web Server (IIS)

Provides a reliable, manageable, and scalable Web application infrastructure.

Messages: None
System Services: 4 Running, 3 Stopped
Events: 2 errors in the last 24 hours

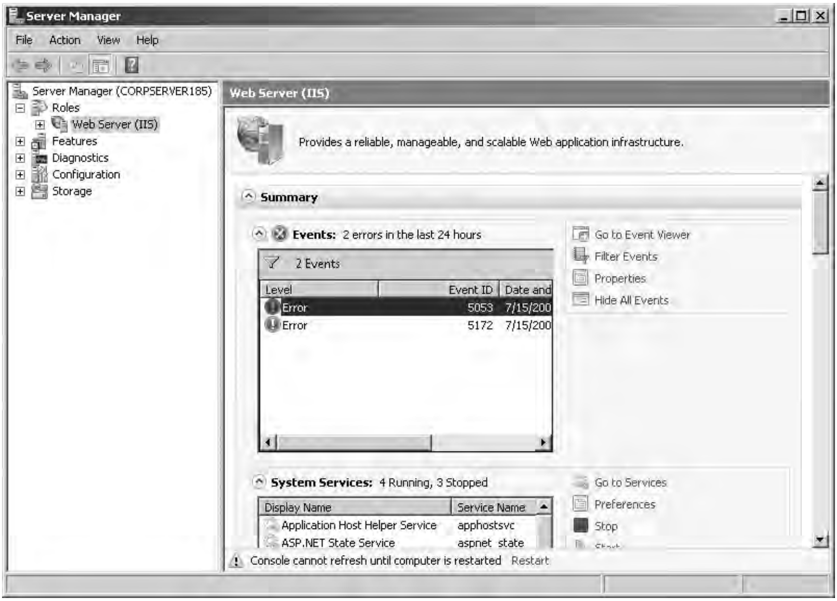
Role Services: 46 installed

Console cannot refresh until computer is restarted

Figure 2-3 View the status details for installed roles.

> **Tip** By default, Server Manager refreshes the details once an hour. You can refresh the details manually by selecting Refresh on the Action menu. If you want to set a different default refresh interval, click Configure Refresh at the bottom of the Summary window, use the options provided to set a new refresh interval, and then click OK.

In Server Manager's main window, if you click a role under Roles Summary or click the Go To Manage Roles link under Roles Summary section or click a role under Roles Summary, Server Manager displays expanded summary details on the events and services for the related role. As shown in Figure 2-4, Server Manager lists all events in the last 24 hours. If you click an event and then click View Event Properties, you can get detailed information about the event. Additionally, Server Manager provides details regarding the system services used by the role and their status. You can manage a service by clicking it and then clicking the related Stop, Start, or Restart links provided. In many cases, if a service isn't running as you think it should, you can click Restart to resolve the issue by stopping and then starting the service.



Provides a reliable, manageable, and scalable Web application infrastructure.

Events: 2 errors in the last 24 hours

2 Events

Level	Event ID	Date and
Error	5053	7/15/200
Error	5172	7/15/200

System Services: 4 Running, 3 Stopped

Display Name	Service Name
Application Host Helper Service	apphostsvc
ASP.NET State Service	aspnet_state

Console cannot refresh until computer is restarted Restart

Figure 2-4 View recent events and manage system services.

Adding or Removing Roles on Servers

When you select the Roles node in Server Manager, the Roles Summary pane section details on the current roles that you've installed. In the Roles Summary section, you'll find options for adding and removing roles. You can add a role as discussed previously in the "Setting Up IIS 7.0" section of this chapter. The roles you can remove depend on the type of server. The roles are as follows:

- On application servers, you can remove the application server role, the Web server role, or both.
- On a Web server, you can remove the Web server role.
- On a server computer running SharePoint Services, you can remove the Windows SharePoint Services role or both the Windows SharePoint Services role and the Web server role.

You can remove a server role by completing the following steps:

1. Start Server Manager by clicking the Server Manager icon on the Quick Launch toolbar or by clicking Start, Administrative Tools, Server Manager.
2. In Server Manager, select the Roles node in the left pane, and then click Remove Roles. This starts the Remove Roles Wizard. If Setup displays the Before You

Begin page, read the Welcome page and then click Next. You can avoid seeing the Welcome page the next time you start this wizard by selecting the Do Not Show Me This Page Again check box before clicking Next.

3. On the Remove Server Roles page, clear the check box for the role you want to remove, and then click Next. If you try to remove a role that another role depends on, you'll see a warning prompt stating that you cannot remove the role unless you also remove the other role as well. If you click Remove Dependent Role Services, Setup will remove both roles.
4. On the Confirm Removal Selections page, review the related role services that Setup will remove based on your previous selections, and then click Remove.
5. When Setup finishes modifying the server configuration, you'll see the Removal Results page. Review the modification details to ensure that all phases of the removal process completed successfully. If any portion of the removal process failed, note the reason for the failure and then use the previously discussed troubleshooting techniques to help resolve the problem.

Viewing and Modifying Role Services on Servers

In Server Manager, you can view the role services configured for a role by selecting Roles in the left pane and then scrolling down to the Role Services section for the role you want to work with. In the details section, you'll find a list of role services that you can install in addition to their current Installed or Not Installed status. You can manage role services for application servers and Web servers by using the Add Role Services and Remove Role Services functions provided for the related role details entry. The Windows SharePoint Services role, however, does not have individual role services that you can manage in this way. With a server computer running SharePoint Services, you can modify the Web server role or remove only the Windows SharePoint Services role.

You can add role services by completing the following steps:

1. Start Server Manager by clicking the Server Manager icon on the Quick Launch toolbar or by clicking Start, Administrative Tools, Server Manager.
2. In Server Manager, select the Roles node in the left pane, and then scroll down until you see the Roles Services section for the role you want to manage. In the Roles Services section for the role, click Add Role Services. This starts the Add Role Services Wizard.
3. On the Select Role Services page, Setup makes the currently selected roles unavailable so that you cannot select them. To add a role, select it in the Role Services list. When you are finished selecting roles to add, click Next, and then click Install.

You can remove role services by completing the following steps:

1. Start Server Manager by clicking the Server Manager icon on the Quick Launch toolbar or by clicking Start, Administrative Tools, Server Manager.
2. In Server Manager, select the Roles node in the left pane and then scroll down until you see the Roles Services section for the role you want to manage. In the Roles Services section for the role, click Remove Role Services. This starts the Remove Role Services Wizard.
3. On the Select Role Services page, Setup selects the currently installed roles. To remove a role, clear the related check box. When you are finished selecting roles to remove, click Next, and then click Remove.

Index

About the Author

William R. Stanek (*http://www.williamstanek.com/*) has over 20 years of hands-on experience with advanced programming and development. He is a leading technology expert, an award-winning author, and an excellent and popular instructional trainer. Over the years, his practical advice has helped millions of technical professionals all over the world. He has written more than 65 books, including *Microsoft Exchange Server 2007 Administrator's Pocket Consultant, Windows Vista Administrator's Pocket Consultant, Windows Server 2008 Administrator's Pocket Consultant,* and *Windows Server 2008 Inside Out*.

Stanek has been involved in the commercial Internet community since 1991. His core business and technology experience comes from over 11 years of military service. He has substantial experience in developing server technology, encryption, and Internet solutions. He has written many technical white papers and training courses on a wide variety of topics. He frequently serves as a subject matter expert and consultant.

Stanek has an MS degree with distinction in information systems and a BS degree (magna cum laude) in computer science. He is proud to have served in the Persian Gulf War as a combat crewmember on an electronic warfare aircraft. He flew on numerous combat missions into Iraq and was awarded nine medals for his wartime service, including the Air Force Distinguished Flying Cross, one of the highest flying honors one can receive in the U.S. military. Currently, he resides in the Pacific Northwest with his wife and children.