

Introducing Windows Server® 2008 R2



Charlie Russel and Craig Zacker
with the Windows® Server Team at Microsoft

PUBLISHED BY

Microsoft Press

A Division of Microsoft Corporation

One Microsoft Way

Redmond, Washington 98052-6399

Copyright © 2010 by Microsoft Corporation

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Control Number: 2009938603

Printed and bound in the United States of America.

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.

Microsoft, Microsoft Press, Access, Active Directory, Aero, BitLocker, DirectX, ESP, Forefront, Hyper-V, MS, SQL Server, Windows, Windows NT, Windows PowerShell, Windows Server, and Windows Vista are either registered trademarks or trademarks of the Microsoft group of companies. Other product and company names mentioned herein may be the trademarks of their respective owners.

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: nSight, Inc.

Technical Reviewer: Bob Hogan, Technical Review services by Content Master, a member of CM Group, Ltd.

Cover: Tom Draper Design

For Sharon—you are truly the love of my life and my boon companion.

—CHARLIE RUSSEL

Contents at a Glance

	<i>Introduction</i>	<i>xvii</i>
CHAPTER 1	What's New in Windows Server R2	1
CHAPTER 2	Installation and Configuration: Adding R2 to Your World	9
CHAPTER 3	Hyper-V: Scaling and Migrating Virtual Machines	25
CHAPTER 4	Remote Desktop Services and VDI: Centralizing Desktop and Application Management	47
CHAPTER 5	Active Directory: Improving and Automating Identity and Access	65
CHAPTER 6	The File Services Role	91
CHAPTER 7	IIS 7.5: Improving the Web Application Platform	109
CHAPTER 8	DirectAccess and Network Policy Server	129
CHAPTER 9	Other Features and Enhancements	147
	<i>Index</i>	<i>163</i>

Contents

Introduction

xvii

Chapter 1 What's New in Windows Server R2 1

What Is R2?	1
Release Cadence	1
Licensing and Packaging Changes	2
The Focus for R2	2
Virtualization	3
Management	3
Scalability	4
Web	4
Networking and Access	5
Better Together with Windows 7	5
Top Reasons to Upgrade	5
Themes Visited Throughout the Book	7
Best Practice Analyzers	7
Windows PowerShell 2.0	8

Chapter 2 Installation and Configuration: Adding R2 to Your World 9

System Requirements and Scalability	11
Processors and Memory	12
Power Consumption	13

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

Clustering	13
Scalability	13
Installation	14
Schema Updates	15
Installation Process	16
Configuration.	16
Role-Based Configuration	19
Server Manager	19
Adding Roles, Role Services, and Features	20
Windows Server Core.	21
Configuring Server Core	21
Managing Server Core	23
Chapter 3 Hyper-V: Scaling and Migrating Virtual Machines	25
The Strategic Role of Virtualization	25
Licensing.	26
Deploying and Managing Virtual Machines	27
Hyper-V Manager Console	27
Configuring Settings for a VM	30
Windows PowerShell Cmdlets	31
SCVMM 2008 R2	33
Managing Virtual Machine Storage.	35
Live Migration	37
Live Migration Compared to Quick Migration	37
Configuring a VM for Live Migration	38
Optimizing Virtual Machine Performance	45
Chapter 4 Remote Desktop Services and VDI: Centralizing Desktop and Application Management	47
(Re)introducing Remote Desktop Services and VDI	47
Providing a Rich Remote Desktop	48
Remote Desktop Administration and Management	49
Windows 7 and RDS (Better Together)	54

Enabling VDI	55
Integrating Remote and Local Applications with RemoteApp	58
Working Over the Web: Web Access	59
Licensing	60
License Server Assignment and Activation	61
Virtual Desktop Licensing	62
Chapter 5 Active Directory: Improving and Automating Identity and Access	65
Using Windows PowerShell with Active Directory	66
Using Active Directory Module for Windows PowerShell	66
Active Directory Administrative Center: Better Interactive Administration	69
Introducing Active Directory Web Services	73
Remote Active Directory Administration with Windows PowerShell Cmdlets	75
Selecting Functional Levels in Windows Server 2008 R2	78
Using the Windows Server 2008 R2 Forest Functional Level	79
Using the Windows Server 2008 R2 Domain Functional Level	80
Active Directory Recycle Bin: Recovering Deleted Objects	82
Understanding Windows Server 2008 R2 Object Recovery	82
Enabling the Active Directory Recycle Bin	83
Using the Active Directory Recycle Bin	84
Offline Domain Join: Securing and Facilitating Deployment	86
Service Accounts	87
Best Practices Analyzer	88
Chapter 6 The File Services Role	91
Using the File Classification Infrastructure	91
Introducing the FCI Components	92
Creating FCI Classification Properties	93
Creating FCI Classification Rules	96
Performing File Management Tasks	99

Using BranchCache	101
Understanding BranchCache Communications	102
Configuring a BranchCache Server	104
Configuring BranchCache Clients	106
Configuring a Hosted Cache Mode Server	107
Introducing Distributed File System Improvements	108
Chapter 7 IIS 7.5: Improving the Web Application Platform	109
Installing IIS 7.5	109
Using Microsoft Web Platform Installer	110
Using the IIS Web Deployment Tool	111
Using New IIS Services	113
Using IIS WebDAV	113
Using FTP Server	114
Hosting Applications with IIS 7.5	115
Running ASP.NET Applications	116
FastCGI Support in IIS 7.5	117
Using Managed Service Accounts	118
Managing IIS 7.5	118
Automating IIS Administration with Windows PowerShell	118
Using IIS Administration Pack Extensions	122
Creating IP Address Restrictions	125
Using Configuration Tracing	126
Using Best Practices Analyzer	127
Using New Performance Counters	128
Accessing IIS Resources on the Internet	128
Chapter 8 DirectAccess and Network Policy Server	129
Introducing DirectAccess	129
IPv6 and IPsec	131
Understanding the DirectAccess Connection Process	132
Deploying DirectAccess	133
Choosing an Access Model	133

DirectAccess Server Requirements	135
DirectAccess Client Requirements	135
DirectAccess Infrastructure Requirements	136
Configuring DirectAccess	136
Using VPN Reconnect	140
New Features in Network Policy Server	142
Configuring NPS Logging	143
Using NPS Templates	144
Migrating IAS Configuration Settings	146
Chapter 9 Other Features and Enhancements	147
Using Windows Server Backup	147
Backing Up Selected Files and Folders	147
Selecting a Backup Destination	150
Creating Incremental Backups	152
Backing Up the System State	153
Backing Up Hyper-V	154
Backing Up from the Command Line	155
BitLocker ToGo	158
 <i>Index</i>	 163

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

Acknowledgments

As always with a book like this, the cast of characters involved can be pretty long, and all of them play a critical role in making the book possible. For us as authors, it almost always starts with the Product Planner, Martin DelRe. Martin gave us a very tight schedule, but then gave us the team to make it possible, including Karen Szall, our Content Development Manager, and Maureen Zimmerman, our Content Project Manager. Both are consummate professionals and a pleasure to work with. When Maureen was on vacation near the end of the project, Melissa von Tschudi-Sutton jumped in and did her usual superb job.

Bob Hogan was our Technical Reviewer, and did a thorough review while providing useful comments that were very much appreciated. Our indexer, Lucie Haskins, and desktop publisher, Terrie Cundiff, did an excellent and much appreciated job. The editorial team, Teresa Horton, Mandy Hagee, and Chris Norton, performed a careful and sensitive edit for which we're very grateful. And last but absolutely not the least, we thank the production and support people at Microsoft Press, without whom this book would not exist. It is a pleasure to work with a team of professionals of this caliber. Thank you.

Charlie would like once again to thank Roger Benes, from Microsoft Canada, who played a crucial and very much appreciated role in helping to make critical connections—plus he's a good and valued friend. Also from Microsoft Canada, I'm indebted to Mark Dickinson, who took that connection to the next step; and to Sasha Krsmanovic and Simran Chaudhry, Charlie's super MVP Leads, for always being there when needed.

Charlie is indebted to Hewlett-Packard Canada for their generous loan of an excellent ML350G5 server to use while writing this book. It's both powerful and quiet. I'd especially like to thank Gordon Pellose and Alan Rogers at HP Canada, and Sharon Fernandez and David Chin of Hill & Knowlton, HP's public relations firm in Canada.

All Charlie's screen captures were made using HyperSnap from Hyperionics, as has been the case for more than 15 years now. It is a great application that I couldn't live without.

Finally, Charlie would like to thank Sharon Crawford, who went way beyond the norm this time!

Introduction

Windows Server 2008 R2, or simply R2 for short, is the second release of Windows Server 2008. It isn't a completely new release, but rather adds additional features and refinements to the existing release. In this book, we focus on the new features and refinements in R2. We assume you have at least a general knowledge of Windows Server, and that you have some familiarity with Windows Server 2008, although we don't assume you're actively running Windows Server 2008. Where an R2 feature is a refinement of a feature that was new in Windows Server 2008, we provide background on the Windows Server 2008 feature to provide context.

Who This Book Is For

This book is targeted primarily at Windows server administrators who are responsible for hands-on deployment and day-to-day management of Windows-based servers for large organizations. Windows server administrators manage file and print servers, network infrastructure servers, Web servers, and IT application servers. They use graphical administration tools as their primary interface but also use Windows PowerShell commandlets and occasionally write Windows PowerShell scripts for routine tasks and bulk operations. They conduct most server management tasks remotely by using Terminal Server or administration tools installed on their local workstation.

What This Book Is About

Covering every aspect of Windows Server 2008 R2 in nine chapters and approximately 200 pages is clearly an impossible task. Rather than try to cover everything, we've focused on what is new and important, while giving you the context from Windows Server 2008.

Chapter 1, "What's New in Windows Server R2" Provides a brief overview of all the new features and capabilities of Windows Server 2008 R2.

Chapter 2, "Installation and Configuration: Adding R2 to Your World" Covers minimum system requirements, basic installation and configuration of R2, and what is involved in adding an R2 server to an existing Windows Server network. Configuration of the Windows Server Core installation option, added in Windows

Server 2008, is covered, along with the steps required to add a Windows Server 2008 R2 domain controller to an existing Windows Server network.

Chapter 3, “Hyper-V: Scaling and Migrating Virtual Machines” Covers the new Hyper-V features of Windows Server 2008 R2, including dynamic storage management and Quick Migration of clustered virtual machines (VMs). Covers creation and management of virtual machines using the Hyper-V Manager console, Windows PowerShell, and the Failover Cluster Manager console and discusses the features of System Center Virtual Machine Manager 2008 R2.

Chapter 4, “Remote Desktop Services and VDI: Centralizing Desktop and Application Management” Covers Remote Desktop Services (the new name for Terminal Services) and the enhancements of Windows Server 2008 R2, including Virtual Desktop Infrastructure (VDI), which uses the new RD Virtualization Host capability of R2 to provide desktop virtualization. R2 also includes an all-new Remote Desktop Services provider for Windows PowerShell.

Chapter 5, “Active Directory: Improving and Automating Identity and Access” Covers the new features of Active Directory (AD), including an AD Recycle Bin, a new set of Active Directory Windows PowerShell cmdlets, and improvements in daily AD administration.

Chapter 6, “The File Services Role” Covers the new File Services features, including BranchCache, Distributed File System–ReadOnly (DFS-R), and the File Classification Infrastructure (FCI).

Chapter 7, “IIS 7.5: Improving the Web Application Platform” Covers the features of the new version of Internet Information Services (IIS), including the new Windows PowerShell management features.

Chapter 8, “DirectAccess and Network Policy Server” Covers the Network Policy Server (NPS) and the new DirectAccess feature that allows Windows 7 computers to be transparently connected to internal network resources from anywhere without requiring a virtual private network (VPN) connection.

Chapter 9, “Other Features and Enhancements” Covers the enhanced version of Windows Server Backup included in R2, including the Windows PowerShell commands for backing up. Also covered is the new BitLocker To Go capability, which provides an important new protection for removable volumes such as backup disks.

Support for This Book

Every effort has been made to ensure the accuracy of this book. As corrections or changes are collected, they will be added to a Microsoft Knowledge Base article accessible via the Microsoft Help and Support site. Microsoft Press provides support for books, including instructions for finding Knowledge Base articles, at the following Web site:

<http://www.microsoft.com/learning/support/books/>

If you have questions regarding the book that are not answered by visiting the site above or viewing a Knowledge Base article, send them to Microsoft Press via e-mail to *mspinput@microsoft.com*.

Please note that Microsoft software product support is not offered through these addresses.

We Want to Hear from You

We welcome your feedback about this book. Please share your comments and ideas via the following short survey:

<http://www.microsoft.com/learning/booksurvey>

Your participation will help Microsoft Press create books that better meet your needs and your standards.

We hope that you will give us detailed feedback via our survey. If you have questions about our publishing program, upcoming titles, or Microsoft Press in general, we encourage you to interact with us via Twitter at *<http://twitter.com/MicrosoftPress>*. For support issues, use only the e-mail address shown above.

What's New in Windows Server R2

- What Is R2? 1
- The Focus for R2 2
- Top Reasons to Upgrade 5
- Themes Visited Throughout the Book 7

In this chapter we cover what is new in Windows Server 2008 R2, and what has changed since the release of Windows Server 2008, along with some basic information about how the book is organized.

What Is R2?

Windows Server 2008 R2, or simply “R2” for short, is the second release of Windows Server 2008. It isn’t a completely new release, but rather adds additional features and refinements to the existing release.

Release Cadence

Beginning with Windows Server 2003, Microsoft moved to a server release cycle that was designed to have a major release every three to five years (Windows Server 2003, Windows Server 2008), with a minor release at the approximate midpoint of the major release cycle (Windows Server 2003 R2, Windows Server 2008 R2). This change allowed Microsoft to move away from including new functionality in service packs (SPs), while providing customers with a more stable and predictable server environment.

An R2 release is more than an SP, but less than a full major release. Windows Server 2008 R2 includes Windows Server 2008 SP2, but it also adds many new features and functionality that were not part of Windows Server 2008.

Licensing and Packaging Changes

There are some minor licensing changes included in Windows Server 2008 R2, and one completely new edition since the original release of Windows Server 2008. The new edition is Windows Server 2008 R2 Foundation, an original equipment manufacturer (OEM)-only edition that is an entry-level small-business solution limited to a maximum of 15 users, which has several other restrictions as well.

MORE INFO For more information on Windows Server 2008 R2 editions, including Windows Server 2008 R2 Foundation, and full details and edition comparisons for all Windows Server 2008 R2 editions, see: <http://www.microsoft.com/windowsserver2008/en/us/R2-editions.aspx>.

The licensing of Windows Server 2008 R2 is very similar to that of Windows Server 2008, and you can use Windows Server 2008 Client Access Licenses (CALs) for Windows Server 2008 R2 without having to upgrade your license. There is, however, one important difference that is introduced with Windows Server 2008 R2—there is no requirement to upgrade to Windows Server 2008 CALs when you install Windows Server 2008 R2 on a physical server that is only used with the Hyper-V role.

Another difference between Windows Server 2008 and Windows Server 2008 R2 licensing is caused by the name change from Terminal Services (TS) in Windows Server 2008 to Remote Desktop Services (RDS) in Windows Server 2008 R2. This is more than just a name change, and we cover the new features and functionality in depth in Chapter 4, “Remote Desktop Services and VDI: Centralizing Desktop and Application Management.” However, for the licensing, it really is just a name change—Windows Server 2008 R2 RDS CALs and Windows Server 2008 TS CALs can both be used for the full functionality of Windows Server 2008 R2 RDS.

There are also new license suite options in Windows Server 2008 R2, with the introduction of the new Virtual Desktop Infrastructure (VDI) Standard and Virtual Desktop Infrastructure Premium suites. We cover these new suite licenses in Chapter 4 when we talk about the new VDI functionality that R2 makes possible.

The Focus for R2

It would be presumptuous of us to talk about the “vision” that Microsoft had for Windows Server 2008 R2, but we can certainly see a pattern in where the major improvements are:

- Virtualization
- Management
- Scalability
- Web

- Networking and access
- “Better Together” with Windows 7

We take a look at each of these areas throughout this book, but let’s start with a quick high-level look at what has changed in each area.

Virtualization

Direct support for server virtualization, in the form of the Hyper-V hypervisor, was one of the most important and highly anticipated improvements in Windows Server 2008. With the release of Windows Server 2008 R2, Microsoft extends Hyper-V virtualization to include support for client desktop virtualization, and adds important new capabilities for dynamic disk allocation, live migration, and improved scalability and redundancy. We cover the improvements in Hyper-V server virtualization capabilities in Chapter 3, “Hyper-V: Scaling and Migrating Virtual Machines.”

Virtualization, however, isn’t limited to machine virtualization, but also includes presentation virtualization (RDS), application virtualization (App-V), and client desktop virtualization (VDI).

Windows Server 2008 R2 adds improvements in RDS that provide a more seamless integration with Windows 7 clients, including full support for Windows Aero and multiple monitors. Application virtualization support in R2 is improved, and the addition of the Remote Desktop Virtualization Host (RD Virtualization Host) role service enables full desktop virtualization. We cover VDI and RDS in greater detail in Chapter 4.

Management

There are substantial improvements in the way Windows Server 2008 R2 can be managed, both graphically and from the command line. A new version of Windows PowerShell provides enhanced remote capabilities and is now available as an installation option for Windows Server Core. Graphical management is also improved, with Server Manager now fully supported remotely, and many of the management consoles are better integrated into Server Manager, enabling remote management. The improvements in Windows PowerShell are covered throughout the book, and we cover the specifics of setting up remote Server Manager, installing Windows PowerShell in Server Core, and many of the changes to role-based administration in Chapter 2, “Installation and Configuration: Adding R2 to Your World.”

Windows Server 2008 R2 includes a new Active Directory (AD) schema that enables an AD Recycle Bin, a new set of Active Directory Windows PowerShell cmdlets, and improvements in daily AD administration.

Improvements in storage management and file server management are part of Windows Server 2008 R2. The new Windows File Classification Infrastructure (FCI) provides insight into your data by automating classification processes so that you can manage your data more effectively and economically. BranchCache improves bandwidth utilization of wide area

network (WAN) connections by enabling local caching of data on Windows Server 2008 R2 and Windows 7 computers at branch offices. Improvements in processor utilization, startup speed, and input/output (I/O) performance make the centralization of storage on iSCSI storage area networks (SANs) easier and more efficient. We cover the details of file system and storage improvements in Chapter 6, “File Server Role.”

Scalability

Windows Server 2008 R2 is the first version of Windows Server to support *only* 64-bit processors. Further, Windows Server 2008 R2 now supports up to 256 logical processor cores for a single operating system instance. Hyper-V virtual machines are able to address up to 64 logical cores in a single host. With the improvements in storage performance and efficiency, and reduced graphical user interface (GUI) overhead, this gives Windows Server 2008 R2 the ability to scale up to larger workloads. Additionally, the R2 version of Hyper-V also adds performance enhancements that increase virtual machine performance and reduce power consumption. Hyper-V now supports Second Level Address Translation (SLAT), which uses new features on today’s CPUs to improve virtual machine (VM) performance while reducing processing load on the Windows Hypervisor. These improvements increase your ability to consolidate workloads and servers onto fewer physical servers, reducing administration overhead, power consumption, and rack costs. Chapters 2 and 3 cover these improvements.

Network Load Balancing (NLB) allows Windows Server 2008 R2 to scale out across multiple servers. Windows Server 2008 R2 includes improvements in support for applications and services that require persistent connections and also improves the health monitoring of NLB clusters and the applications and services running on them.

Web

Windows Server 2008 R2 includes Internet Information Services (IIS) 7.5, an improved and updated version of the IIS 7 that was included in Windows Server 2008. Windows Server 2008 R2 also includes a new Windows PowerShell provider for IIS to facilitate the automation of management tasks. This Windows PowerShell provider is available on Server Core installations of Windows Server 2008 R2 as well as full installations. IIS 7.5 also includes a new File Transfer Protocol (FTP) server that supports Internet Protocol version 6 (IPv6), Secure Sockets Layer (SSL), and Unicode characters.

Server Core can now include the Microsoft .NET Framework, giving administrators the ability to manage IIS from Windows PowerShell or IIS Manager. As with many other areas of R2, IIS 7.5 includes a Best Practices Analyzer (BPA) to simplify troubleshooting and configuration of IIS. For full details on the new version of IIS, see Chapter 7, “IIS 7.5: Improving the Web Application Platform.”

Networking and Access

One of the most exciting new features in Windows Server 2008 R2 is DirectAccess, a new way to securely connect remote clients to the corporate network. The most common method has been virtual private networks (VPNs), which often require third-party client software running on the client, and can be time-consuming to configure and troubleshoot. With Windows Server 2008 R2 and DirectAccess, if the client is running Windows 7, the remote user has seamless, always-on remote access to corporate resources that does not compromise the secure aspects of remote connectivity.

DirectAccess works with the Network Access Protection (NAP) of Windows Server 2008 R2 to ensure that client computers meet your system health requirements, such as having security updates and antimalware definitions installed, before allowing a DirectAccess connection.

Clients that are connected via DirectAccess can be remotely managed by internal IT staff, allowing you to ensure that they are kept current with critical updates. DirectAccess is covered in Chapter 8, “DirectAccess and Network Policy Server.”

Better Together with Windows 7

Many of the enhancements of Windows Server 2008 R2 are independent of the client operating system being used, but others, such as DirectAccess, only work with Windows 7 clients. Others, as is the case with the new RDS features, work better with a Windows 7 client, but are still important improvements even if you’re running Windows Vista or Windows XP.

Some of the things that make Windows 7 and Windows Server 2008 R2 work better together (and the technologies that enable them) are the following:

- Simplified remote connectivity for remote users (DirectAccess)
- Secure remote connectivity, even from public computers (Remote Workplace plus RD Gateway and RD Session Host)
- Improved branch office performance and security (BranchCache and read-only Distributed File System Replication [DFS-R])
- More efficient power management where the hardware supports it (Group Policy)
- Virtualized desktops (VDI)
- Improved removable drive security (BitLocker To Go)

Top Reasons to Upgrade

Windows Server 2008 R2 is not a free update to Windows Server 2008 unless you have Software Assurance (SA). So should you upgrade? And why?

Well, the short answer is yes, you should upgrade. The why is what this book is all about in many ways, but here are our top 10 reasons to upgrade:

- **Powerful hardware and scaling features** Windows Server 2008 R2 supports up to 256 logical processors. R2 also supports SLAT, which enables R2 to take advantage of the Enhanced Page Tables feature found in the latest AMD CPUs as well as the similar Nested Page Tables feature found in Intel's latest processors. The combination enables R2 servers to run with much improved memory management.
- **Improved Hyper-V** Hyper-V in Windows Server 2008 R2 can now access up to 64 logical CPUs on host computers—twice Hyper-V's initial number of supported CPUs. Live migration enables a highly fault-tolerant virtualization infrastructure, and dynamic addition and removal of disks simplifies backup scenarios and overall management of virtualized resources.
- **Reduced power consumption** Windows Server 2008 R2 supports Core Parking, which dynamically turns off unused processor cores when they aren't needed, reducing power consumption.
- **Reduced desktop costs** Windows Server 2008 R2 enables VDI technology, which extends the functionality of RDS to provide full desktop virtualization or application virtualization of key applications.
- **Improved server management** Windows Server 2008 R2 includes a new version of Windows PowerShell, which is now available on Server Core as well. Server Manager can now also be used remotely.
- **Improved branch office performance and security** Windows Server 2008 R2 includes BranchCache and read-only DFS-R, which extends the branch office scenarios introduced in Windows Server 2008.
- **Improved Web server** Windows Server 2008 R2 includes IIS 7.5 as well as a new FTP server. IIS 7.5 includes a new Windows PowerShell provider for IIS management.
- **Windows PowerShell v2** Windows Server 2008 R2 includes an improved and more powerful version of Windows PowerShell that has cmdlet support for remote management. Windows PowerShell is now available on Server Core in Windows Server 2008 R2.
- **Improved Remote Desktop Services** The new RDS features provide an improved and more seamless user experience, especially when the client is running Windows 7.
- **Improved mobile user experience** Mobile users running Windows 7 have seamless and continuous access to corporate resources through DirectAccess. And RD Web Access, shown in Figure 1-1, provides users running at least Windows XP SP3 with full access to published applications or desktops.

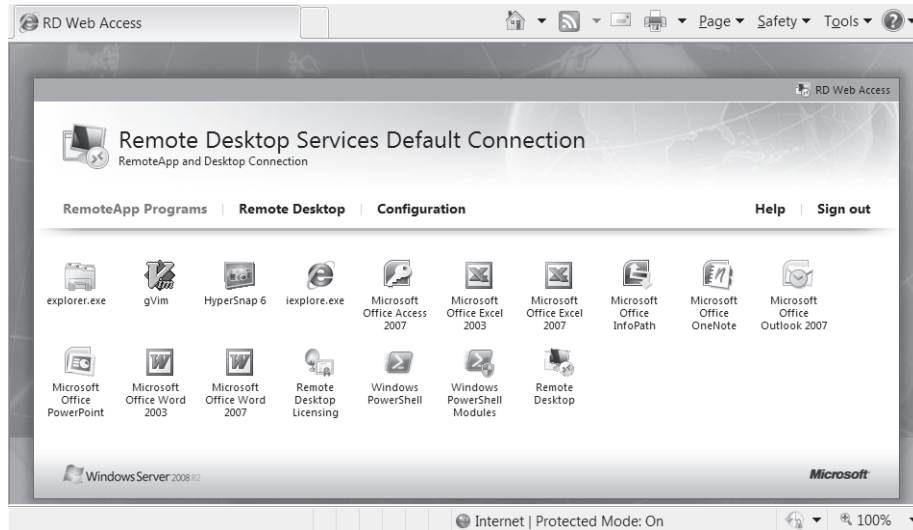


FIGURE 1-1 Remote Desktop Web Access requires at least Windows XP SP3.

Themes Visited Throughout the Book

Throughout this book, we focus on what is new and different in Windows Server 2008 R2, and we assume that you have at least some familiarity with Windows Server 2008. Inevitably, there will be some overlap between the features that were introduced in Windows Server 2008, and the improvements or changes in R2. We try to keep from telling you what you already know about Windows Server 2008, but in some cases we need to set the stage as we go, so bear with us, please.

Two important additions in Windows Server 2008 R2 that we use throughout the book are the many new BPAs, and the new version of Windows PowerShell.

Best Practice Analyzers

BPAs have been around for a while, but usually focused on server applications, such as Microsoft Exchange, or on suite products such as Windows Small Business Server. New in Windows Server 2008 R2 are several new BPAs that are directly integrated into Server Manager. These BPAs are part of the role-based management of Server Manager, and they scan for deviations from known best practices for the particular role. A typical error is shown in Figure 1-2.

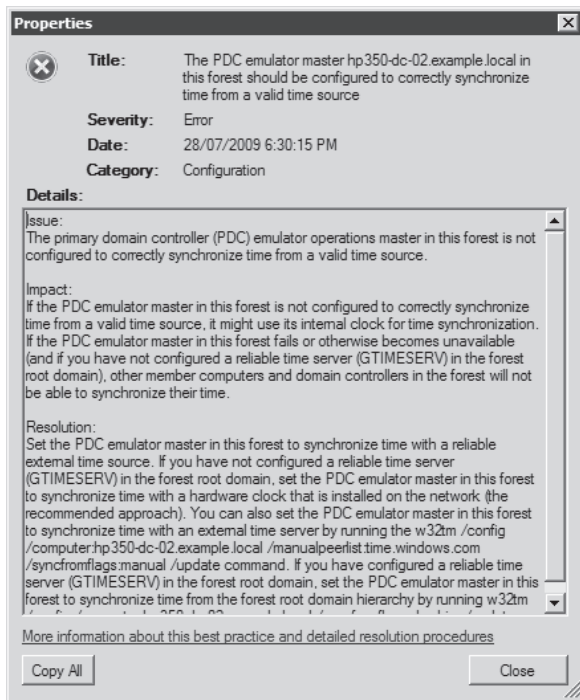


FIGURE 1-2 The Active Directory BPA.

The BPAs are an important new feature in Windows Server 2008 R2, and we cover them as we go through each area.

Windows PowerShell 2.0

The other new feature in Windows Server 2008 R2 that crosses just about every chapter is Windows PowerShell 2.0. This new version of Windows PowerShell adds many new cmdlets, and has built-in support for running commands remotely. It is available for earlier versions of Windows operating systems, but it is installed by default in Windows Server 2008 R2. We use it to provide simple scripts or command-line ways of doing tasks throughout the book. An important design criterion for Windows PowerShell 2.0 was that it run Windows PowerShell 1.0 commands and scripts seamlessly. This protects your existing investment in Windows PowerShell scripting and makes it easy for you to extend your existing Windows PowerShell knowledge to encompass the new capabilities of 2.0.

Installation and Configuration: Adding R2 to Your World

- System Requirements and Scalability 11
- Installation 14
- Configuration 16
- Windows Server Core 21

Windows Server 2008 R2 uses the same basic installation and configuration methods as Windows Server 2008. The installer, originally introduced in Microsoft Windows Vista, is an image-based install that is noticeably quicker than earlier versions of Windows Server. Configuration continues the role-based model introduced in Windows Server 2008, now with a new ServerManager module for Windows PowerShell as an option for adding and removing roles and features. This new capability is also available on Server Core installations, a change from Windows Server 2008 where Windows PowerShell was not supported on Server Core.

Additionally, for Server Core, the command-line utility used to add and remove roles has changed. In Windows Server 2008, the utility is Ocsetup.exe, but in Windows Server 2008 R2, it is Dism.exe.

Windows Server Core

If you're coming to Windows Server 2008 R2 from Microsoft Windows Server 2003, a brief explanation of Server Core is probably in order here. With the release of Windows Server 2008, Microsoft added a new installation option called Server Core. This installs a version of Windows Server that has a limited subset of available roles and functionality, and no graphical interface, as shown in Figure 2-1.

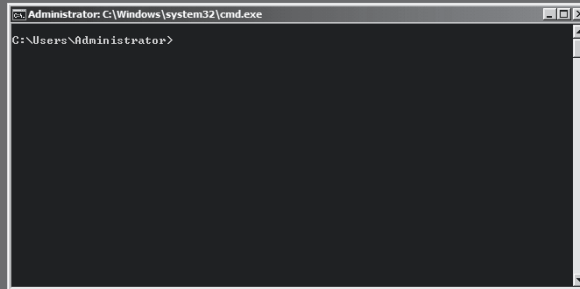


FIGURE 2-1 The console interface of Windows Server 2008 R2 Enterprise Core.

Server Core is not a separate edition of Windows Server 2008 R2, it is an *installation option* that has a reduced footprint and reduced overhead, but still provides all the underlying server functionality of the roles that are available on it. You can't go out and buy a copy of Windows Server Core. Instead, you buy whatever version of Windows Server you need for your network, and when you install Windows Server, you choose a Server Core installation, as shown in Figure 2-2.

Management of server roles can be done from the command line, or from remote management tools running on other computers in the network.

So, why choose Server Core? After all, most Windows Server administrators are a good deal more comfortable with the familiar Windows graphical interface than they are with the command line, and even an experienced administrator can find the single Cmd.exe window shown in Figure 2-1 a bit daunting. The two reasons we find most compelling are the reduction in resource usage—a Server Core installation is physically smaller and uses less RAM—and the improved security footprint—because there are fewer services and features installed, there is a smaller attack surface. This also has the added benefit of requiring fewer security-related updates and potentially fewer server restarts.

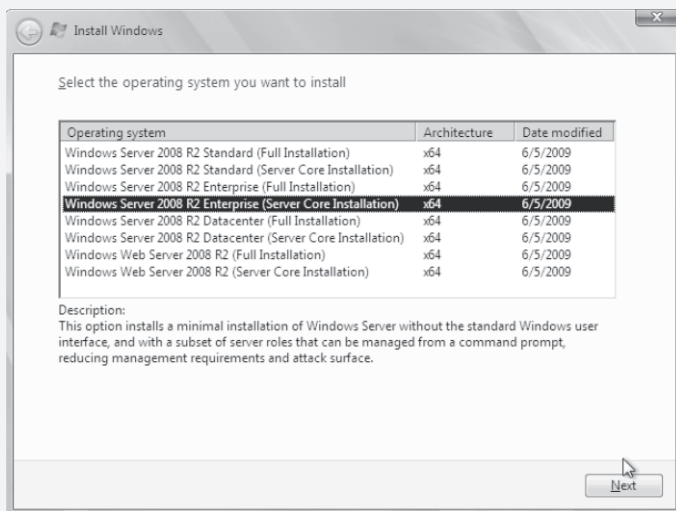


FIGURE 2-2 Server Core is an installation option, not a separate product.

System Requirements and Scalability

The system requirements for Windows Server 2008 R2 are essentially the same as for Windows Server 2008, with one very important exception: There is no 32-bit version of Windows Server 2008 R2. There are only 64-bit versions. The minimum system requirements are shown in Table 2-1.

TABLE 2-1 Minimum System Requirements for Windows Server 2008 R2

COMPONENT	REQUIREMENT
Processor	Minimum: 1.4 GHz x64 processor
	Note: An Intel Itanium 2 processor is required for Windows Server 2008 for Itanium-Based Systems
Memory	Minimum: 512 MB RAM (384 MB for Server Core installation)
	Maximum: 8 GB (Foundation) or 32 GB (Standard) or 2 TB (Enterprise, Datacenter, and Itanium-Based Systems)
Disk Space Requirements	Minimum: 32 GB or greater (3.5 GB for Server Core installation)
	Foundation: 10 GB or greater
	Note: Computers with more than 16 GB of RAM will require more disk space for paging and dump files
Display	Super VGA (800 × 600) or higher resolution monitor
Other	Keyboard and Microsoft Mouse or compatible pointing device

Processors and Memory

The Windows Server 2008 R2 editions support the same number of physical processors and RAM as Windows Server 2008 did, up to 64 processor sockets, and up to 2 terabytes (TB) of RAM, for Windows Server 2008 R2 Datacenter and Itanium versions. Table 2-2 shows the breakdown by edition.

TABLE 2-2 Windows Server 2008 R2 Memory and Processors by Edition

EDITION	MAXIMUM # OF CPUS	MAXIMUM RAM
Web	4	32 GB
Standard	4	32 GB
Enterprise	8	2 TB
Datacenter	64	2 TB
Itanium	64	2 TB
Foundation	1	8 GB

Microsoft counts processor sockets, not logical processors, for most licensing purposes and for the consideration of maximum number supported. The exception to this is the Hyper-V role of Windows Server 2008 R2, which supports a maximum of 64 logical processors for a single physical server.

Second Level Address Translation

Windows Server 2008 R2 adds support for the enhanced memory management capabilities of the newest Intel and AMD processors. AMD calls this Rapid Virtualization Indexing (RVI) and Intel calls it Enhanced Page Tables. In both cases, it allows the Hyper-V hypervisor to manage memory, especially of large-memory virtual machines (VMs), more effectively and with less overhead in the parent partition. Second Level Address Translation (SLAT) works by providing two levels of address translation. The additional page table is used to translate guest “physical” addresses to system physical addresses. Guest operating systems can now be allowed to directly manage their own page tables, without the need for the hypervisor to intercept those calls, reducing the overhead required for the Hyper-V parent to maintain shadow page tables in software.

Power Consumption

Power consumption, and the carbon footprint it generates, is an ever increasing concern for most information technology (IT) managers these days. The cost of the power itself, along with the resulting cost of cooling to remove the excess heat generated, adds significantly to the overall cost of running a datacenter. Modern server processors have helped improve this by using less actual power per CPU, but this has been offset to some extent by the increasing need for more RAM and more CPUs. Windows Server 2008 R2 helps manage the overall power consumption of datacenters in several ways, including the following:

- **Server consolidation** Windows Server 2008 R2 supports more logical processors per physical Hyper-V host, giving you the ability to consolidate more workloads onto fewer physical servers.
- **Core parking** Windows Server 2008 R2 is able to take advantage of the ability of modern processors to dynamically enable and disable processor cores. When Windows Server recognizes that processors are being underutilized, it turns off or parks processor cores that aren't needed, reducing power consumption. When processor demand increases, Windows Server 2008 R2 reenables cores as necessary to maintain system performance.
- **Group Policy management of P-states** Windows Server 2008 R2 utilizes Group Policy to change the Advanced Configuration and Power Interface (ACPI) power-performance states (P-states) of the processors to manage the speed and power consumption of the processors.
- **Storage consolidation** Windows Server 2008 R2 is able to better utilize storage area networks (SANs), including booting directly from an SAN, allowing you to centralize and consolidate storage more effectively.

Clustering

Windows Server 2008 R2 adds a new Cluster Shared Volume (CSV) feature to failover clustering to enable live migration of VMs. CSV volumes enable multiple nodes in the same failover cluster to concurrently access the same logical unit number (LUN). By storing the VHD files for a virtual machine on the CSV, migration of a VM happens without interruption of service. Also new in failover clustering is improved connectivity fault tolerance, and an enhanced cluster validation tool. More on clustering is discussed in Chapter 3, "Hyper-V: Scaling and Migrating Virtual Machines," when we talk about Hyper-V.

Scalability

A key design goal was to provide higher performance for Windows Server 2008 R2 on similar hardware. Windows Server 2008 R2 features that improve performance and scalability for applications and services include the following:

- Support for larger workloads by adding more servers to a workload (scaling out)
- Support for larger workloads by utilizing or increasing system resources (scaling up)

Increased Workload Support by Scaling Out

The Network Load Balancing (NLB) feature in Windows Server 2008 R2 allows you to combine two or more computers into a cluster. You can use NLB to distribute workloads across the cluster nodes to support a larger number of simultaneous users. NLB feature improvements in Windows Server 2008 R2 include the following:

- Improved support for applications and services that require persistent connections using the new IP Stickiness feature in NLB clusters
- Improved health monitoring and awareness for applications and services running on NLB clusters

Installation

Installation of Windows Server 2008 R2 uses the same general steps as Windows Server 2008, with the exception that you won't be prompted for a license key during the installation, as you are with some distributions of Windows Server 2008. We do not cover the detailed step-by-step of Windows Server installation here—that's adequately covered in many places, including [http://technet.microsoft.com/en-us/library/dd540768\(Ws.10\).aspx](http://technet.microsoft.com/en-us/library/dd540768(Ws.10).aspx).

We focus in this section on the various upgrade scenarios and what is and isn't supported. The general rule is that upgrades of the same architecture, language, edition, and platform from Windows Server 2003 SP2, Windows Server 2003 R2, and Windows Server 2008 are supported. Upgrades from beta builds of Windows Server 2008 R2 are not supported, and upgrades from versions of Windows Server prior to Windows Server 2003 SP2 are not supported.

If you are running a 32-bit version of Windows Server, even if the underlying hardware is 64-bit, there is no upgrade available.

The specific supported upgrade scenarios are shown in Table 2-3.

TABLE 2-3 Supported Upgrade Scenarios for Windows Server 2008 R2

SOURCE VERSION	SUPPORTED TARGET VERSION OF WINDOWS SERVER 2008 R2
FROM WINDOWS SERVER 2003 (SP2, R2)	
Datacenter	Datacenter
Enterprise	Enterprise, Datacenter
Standard	Standard, Enterprise

FROM WINDOWS SERVER 2008	
Datacenter	Datacenter
Datacenter Core	Datacenter Core
Enterprise	Enterprise, Datacenter
Enterprise Core	Enterprise Core, Datacenter Core
Foundation (SP2 only)	Standard
Standard	Standard, Enterprise
Standard Core	Standard Core, Enterprise Core
Web	Standard, Web
Web Core	Standard Core, Web Core

There are a couple of omissions in the upgrade paths that are worth pointing out. There is no upgrade path for Itanium versions of Windows Server—the expectation is that a full, clean install will be performed. There is also no way to upgrade to Windows Server 2008 R2 Foundation. If you have Windows Server 2008 Foundation, which shipped at the SP2 level, you can upgrade to Windows Server 2008 R2 Standard only.

Also, Microsoft does support upgrades from both the Release Candidate (RC) and Interim Development Server (IDS) builds of Windows Server 2008.

Even where it is technically possible and supported to upgrade, in our experience it's always worth considering a clean installation. This is especially true if the server being upgraded has already gone through one or more upgrades to get to its current level.

Schema Updates

Joining a computer running Windows Server 2008 R2 to an existing Active Directory domain doesn't require an update to the Active Directory schema. However, before you can make a computer running Windows Server 2008 R2 a domain controller, you do need to prepare the forest and the domain that will have an R2 domain controller. To prepare the forest, follow these steps:

1. Log on to the domain controller that holds the Schema Master flexible single master operations (FSMO) role with an account that is a member of the Schema Admins group.
2. Copy the contents of the \Support\Adprep folder on the Windows Server 2008 R2 DVD to a local folder.
3. Open a command prompt *as administrator* and change to the directory where you copied the files.
4. Run the following command:

```
Adprep /forestprep
```

5. Allow the changes to replicate before preparing the domain.

If you're installing Windows Server 2008 R2 into an existing forest, but a new domain, you don't need to do anything else, but if you're installing into an existing domain, you'll need to prepare that domain using the following steps:

1. Log on to the domain controller that holds the Infrastructure Master FSMO role with an account that is a member of the Domain Admins group.
2. Copy the contents of the \Support\Adprep folder on the Windows Server 2008 R2 DVD to a local folder.
3. Open a command prompt *as administrator* and change to the directory where you copied the files.
4. Run the following command:

```
Adprep /domainprep /gpprep
```

5. Allow the changes to replicate before installing the new Windows Server 2008 R2 domain controller.

MORE INFO See [http://technet.microsoft.com/en-us/library/cc731728\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc731728(WS.10).aspx) for more information on Adprep.exe.

Installation Process

The installer for Windows Server 2008 R2 is the same installer that was introduced with Windows Vista. Before you start the installation on x64 systems, however, you need to verify that you have *digitally signed* drivers for any hardware that will be used on the server. Starting with Windows Server 2008, all drivers for x64 versions of Windows Server must be digitally signed or they will not load during the boot process. This can cause the server to fail to boot, or to have hardware unavailable, so it's a good idea to make sure you have all the drivers you need before you start.

Windows Server 2008 R2 doesn't require a license key to install, but you will need to provide one within 60 days to continue to use the software. As you can see in Figure 2-2, you must choose the edition of R2 you want to install. This choice must match the license key you use to activate the software or activation will fail.

Configuration

The final step of the Windows Server 2008 R2 installation is setting the password on the Administrator account, as shown in Figure 2-3.

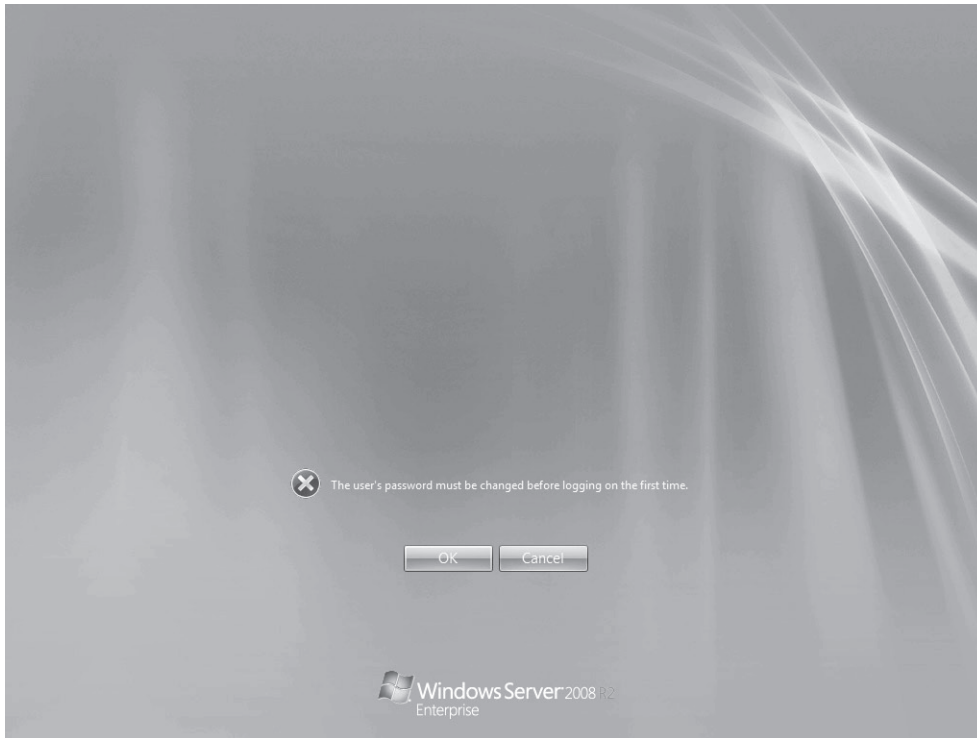


FIGURE 2-3 Setting the initial Administrator password.

The default password policy for Windows Server 2008 R2 is to require a minimum of six characters, with at least three of the four categories of characters: lowercase, uppercase, numbers, and nonalphanumeric characters. Passwords expire in 42 days, by default. Once a server is joined to a domain, the policies of the domain will apply for domain accounts, but the local security policy will still apply for local accounts, as shown in Figure 2-4.

Once the password is set, you'll see the Initial Configuration Tasks Wizard, as shown in Figure 2-5. This wizard is also known as the Out of Box Experience (OOBE) and is similar to the one from Windows Server 2008, with the addition of the Activate Windows option. The OOBE is a useful wizard for the initial configuration of a server, providing easy access on a single page to most of the tasks you need to get your server up and running.

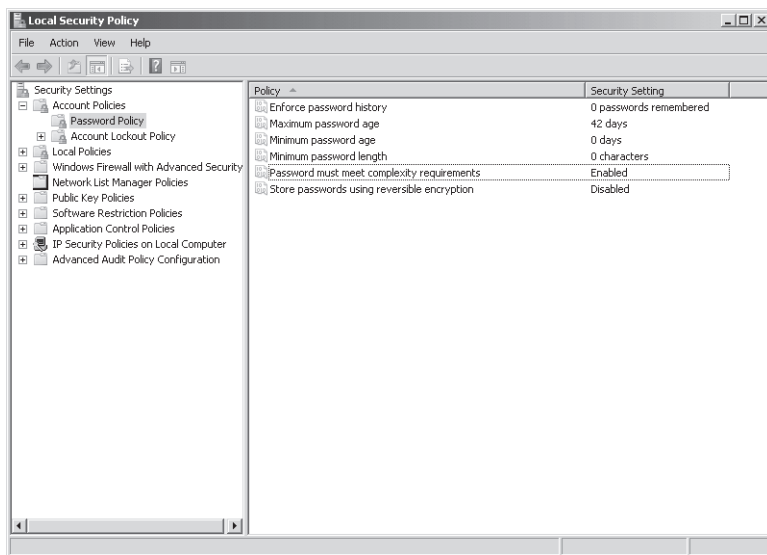


FIGURE 2-4 The Local Security Policy controls password policies for local accounts.

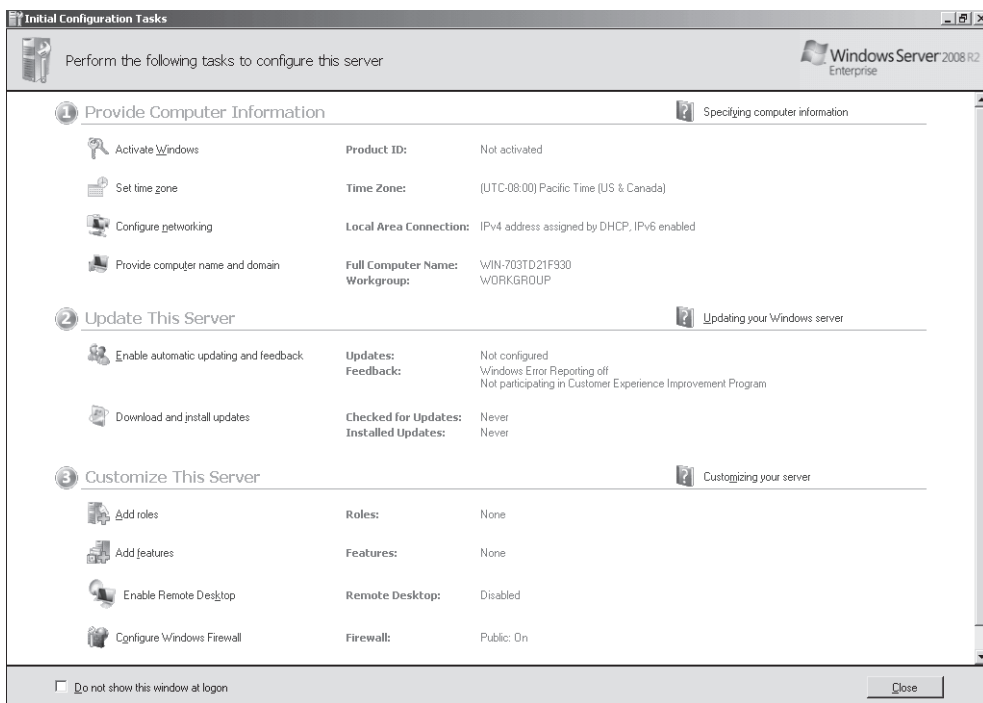


FIGURE 2-5 The Initial Configuration Tasks Wizard, or OOBE.

Role-Based Configuration

If you're familiar with Windows Server 2008, you'll already have a good start on the role-based configuration that is used in Windows Server 2008 R2, but if you're new to Windows Server 2008, then a quick overview should help. Windows Server 2008 and Windows Server 2008 R2 both use role-based configuration. All the features and roles that are available to the server are physically installed on the server's hard drive, as part of the image-based install. You don't ever have to worry about finding the right DVD for your server if there's an update or you need to add a new feature or role because all the necessary files are already on the hard drive.

When you want to enable specific functionality on the server, you add the *role*, *role service*, or *feature* that includes that functionality. This is an important change that ensures that each role gets only the services and features enabled that are required by the role and no others, limiting the overall attack surface of the server. Enabling the role also configures the Windows Firewall for that role, enabling the role or feature to work without opening up unnecessary ports that could create an unintended security risk.

There are 17 possible roles and 42 different features that can be enabled on Windows Server 2008 R2 Enterprise Edition.

Server Manager

The primary graphical interface for server management in Windows Server 2008 R2 is the Server Manager console, shown in Figure 2-6.

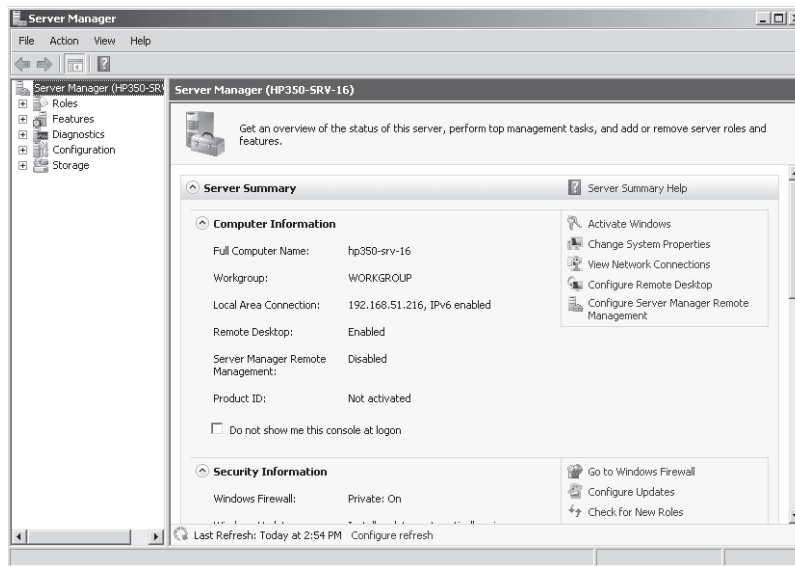


FIGURE 2-6 The Server Manager console.

The Server Manager console includes integrated management consoles for the roles and features that are enabled on the server. New in Windows Server 2008 R2 is the ability to run the Server Manager remotely without having to open a Remote Desktop session to the remote server.

Also new in the R2 version of Server Manager are Best Practice Analyzers (BPAs) that are directly integrated into the Server Manager for those roles that have them.

Adding Roles, Role Services, and Features

Adding a role, role service, or feature in Windows Server 2008 R2 can be done from Server Manager, from the Initial Configuration Tasks Wizard, or from Windows PowerShell. The Server Manager and Initial Configuration Tasks Wizard experience is essentially the same as it was in Windows Server 2008, but the option to use Windows PowerShell is new.

To use Server Manager to add a role or feature, select Server Manager (<servername>) in the tree pane and then, from the Action menu, select Add Roles (or Add Features). To add a role service for an already installed role, highlight that role in the tree pane and, from the Action menu, select Add Role Service. The Add Role Wizard, Add Role Services Wizard, or Add Feature Wizard will open. All three wizards are essentially the same. The Add Role Wizard is shown in Figure 2-7.

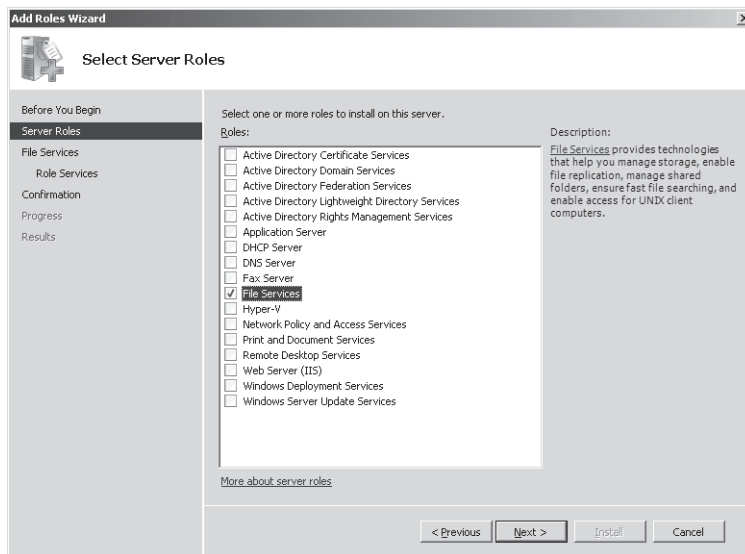
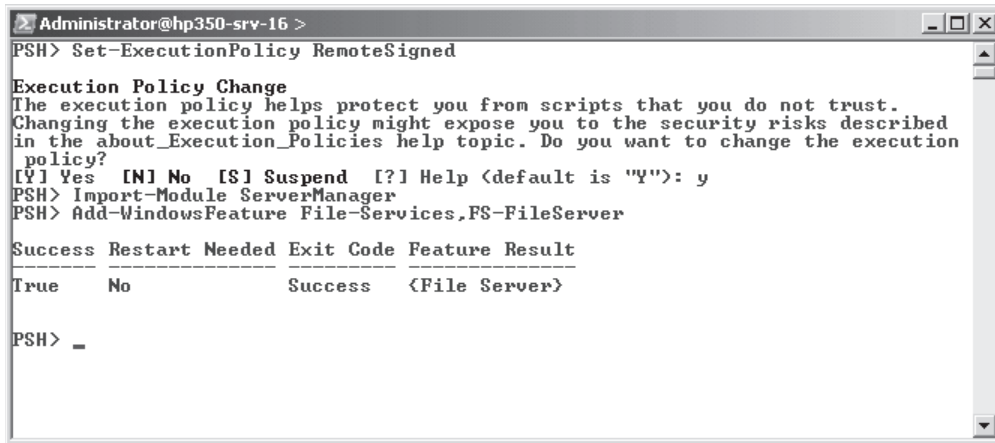


FIGURE 2-7 The Add Role Wizard, with the File Services role selected.

A new alternative that makes it easier to script and automate the configuration of servers is Windows PowerShell. Windows Server 2008 R2 has a new ServerManager module that can be used to add or remove roles, role services, or features. Figure 2-8 shows a Windows PowerShell session that sets the execution policy to only require signing for scripts that originate

remotely, then imports the ServerManager module, and finally adds the File Services role, along with the File Server role service.



```
Administrator@hp350-srv-16 >
PSH> Set-ExecutionPolicy RemoteSigned

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust.
Changing the execution policy might expose you to the security risks described
in the about_Execution_Policies help topic. Do you want to change the execution
policy?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): y
PSH> Import-Module ServerManager
PSH> Add-WindowsFeature File-Services,FS-FileServer

Success Restart Needed Exit Code Feature Result
-----
True      No              Success    {File Server}

PSH> _
```

FIGURE 2-8 Adding the File Services role using Windows PowerShell.

Windows Server Core

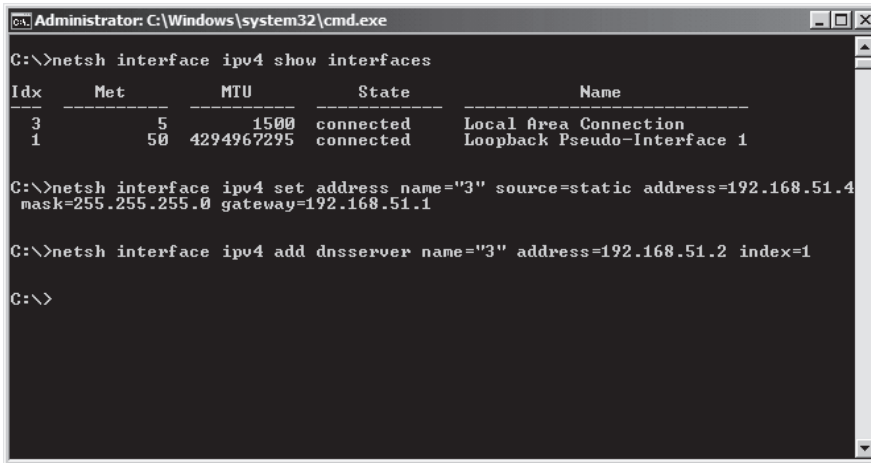
The option to choose a minimal environment for running specific server roles was a very new installation option in Windows Server 2008, and that option has been enhanced in Windows Server 2008 R2 with the addition of Active Directory Certificate Services as a role, and the inclusion of Windows PowerShell as a supported feature.

Server Core is an installation option, not a separate edition of Windows Server 2008 R2. You can install Server Core regardless of which edition you are installing—it's really just a decision about interface and functionality.

Configuring Server Core

Configuring and managing a Server Core installation is a bit different than a full installation of Windows Server 2008 R2. The initial configuration is especially different because the Initial Configuration Tasks Wizard isn't available. Once the server is configured, however, you can use standard remote management tools to manage the roles and features on the server, including using Server Manager.

The following steps outline how to perform a basic Server Core configuration to give the server a fixed Internet Protocol (IP) address and join it to the domain. These instructions assume you've completed the basic installation and set the default administrator password, and you are now staring at the blank Cmd.exe prompt shown earlier in Figure 2-1. Use the commands shown in Figure 2-9 to configure the network adapter for a fixed IP address of 192.168.51.4 with a Domain Name System (DNS) server at 192.168.51.2.



```
Administrator: C:\Windows\system32\cmd.exe

C:\>netsh interface ipv4 show interfaces

Idx      Met      MTU      State      Name
-----
3        5        1500     connected  Local Area Connection
1        50       4294967295 connected  Loopback Pseudo-Interface 1

C:\>netsh interface ipv4 set address name="3" source=static address=192.168.51.4
mask=255.255.255.0 gateway=192.168.51.1

C:\>netsh interface ipv4 add dnsserver name="3" address=192.168.51.2 index=1

C:\>
```

FIGURE 2-9 Setting a fixed IP address.

Now, join the server to the example.local domain using the following command:

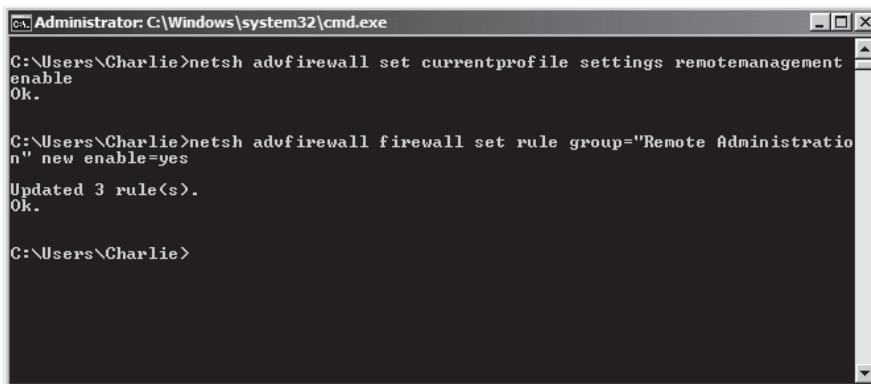
```
Netdom join %computername% /domain:example.local /userd:example\Charlie /passwordd:*
```

Restart the server using **shutdown -r** and log back in with a domain administrator account to confirm that the domain join went as expected. Once you're back at the inspiring Server Core command line, you need to rename the computer something a bit more memorable than the random name given it during the initial install. The command to do this is Netdom again:

```
Netdom renamecomputer %computername% /newname:<yournamehere>
```

Answer Yes at the prompt, and then restart the computer after the rename and log back in with a domain administrator account.

Now, configure the firewall for remote administration and enable remote management through the firewall, using the commands shown in Figure 2-10.



```
Administrator: C:\Windows\system32\cmd.exe

C:\Users\Charlie>netsh advfirewall set currentprofile settings remotemanagement
enable
Ok.

C:\Users\Charlie>netsh advfirewall firewall set rule group="Remote Administration" new enable=yes

Updated 3 rule(s).
Ok.

C:\Users\Charlie>
```

FIGURE 2-10 Enabling remote management.

Finally, use the new Server Configuration utility, Sconfig.exe, to configure the rest of the settings, as shown in Figure 2-11.

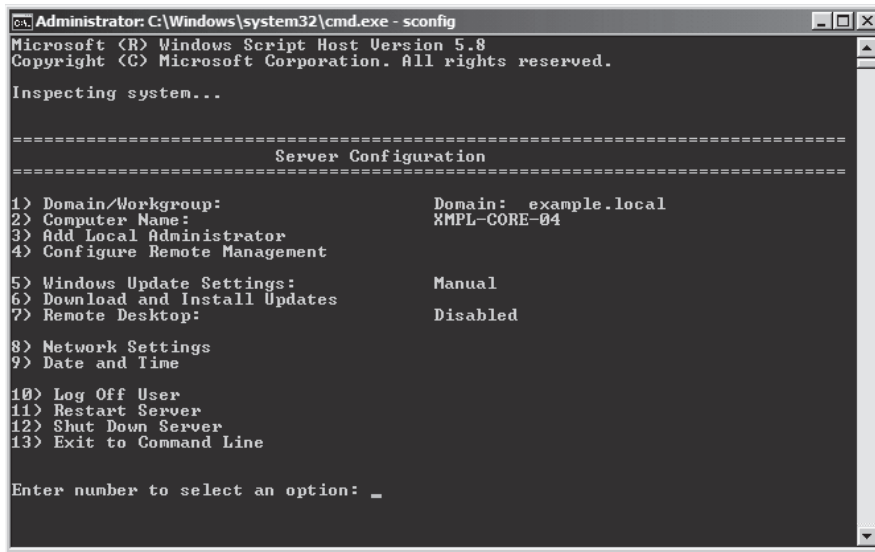


FIGURE 2-11 Sconfig.exe makes configuring some options much easier.

Sconfig.exe is new in Windows Server 2008 R2 and allows you to configure most of the settings you need to get up and running with Server Core. This includes enabling remote Server Manager, remote management consoles, and Windows PowerShell, which are critical steps to getting your Server Core installation ready to use.

Managing Server Core

Once Windows PowerShell is installed, and you have remote management and Remote Desktop enabled, you are in a position to manage the server comfortably using familiar tools. You'll need to use Windows PowerShell or the Dism.exe command-line utility to add or remove roles, role services, and features because you can't use the remote management tools or Server Manager to add roles remotely. You can manage a server running Server Core installation in the following ways:

- **Locally and remotely using a command prompt** By using the Windows command-line tools at a command prompt, you can manage servers running a Server Core installation.
- **Remotely using Terminal Server** By using another computer running Windows, you can use the Terminal Server client to connect to a server running a Server Core installation and manage it remotely. The shell in the Terminal Server session will be the command prompt.

- **Remotely using Windows Remote Shell** By using another computer running Windows Vista, Windows 7, Windows Server 2008, or Windows Server 2008 R2, you can use Windows Remote Shell to run command-line tools and scripts on a server running a Server Core installation.
- **Locally or remotely using Windows PowerShell** By using Windows PowerShell locally on a computer running a Server Core installation of Windows Server 2008 R2 or remotely from a computer running Windows Server 2008 R2, you can connect to a server running a Server Core installation in the same way that you would connect to any computer running Windows.
- **Remotely using a Microsoft Management Console (MMC) snap-in** By using an MMC snap-in from a computer running Windows Vista, Windows 7, Windows Server 2008, or Windows Server 2008 R2, you can connect to a server running Server Core installation in the same way that you would connect to any computer running Windows.
- **Remotely using Server Manager** By using Server Manager from a computer running Windows Server 2008 R2 you can connect to a server running a Server Core installation of Windows Server 2008 R2 and manage it.

Figure 2-12 shows Server Manager connecting to a Server Core computer.

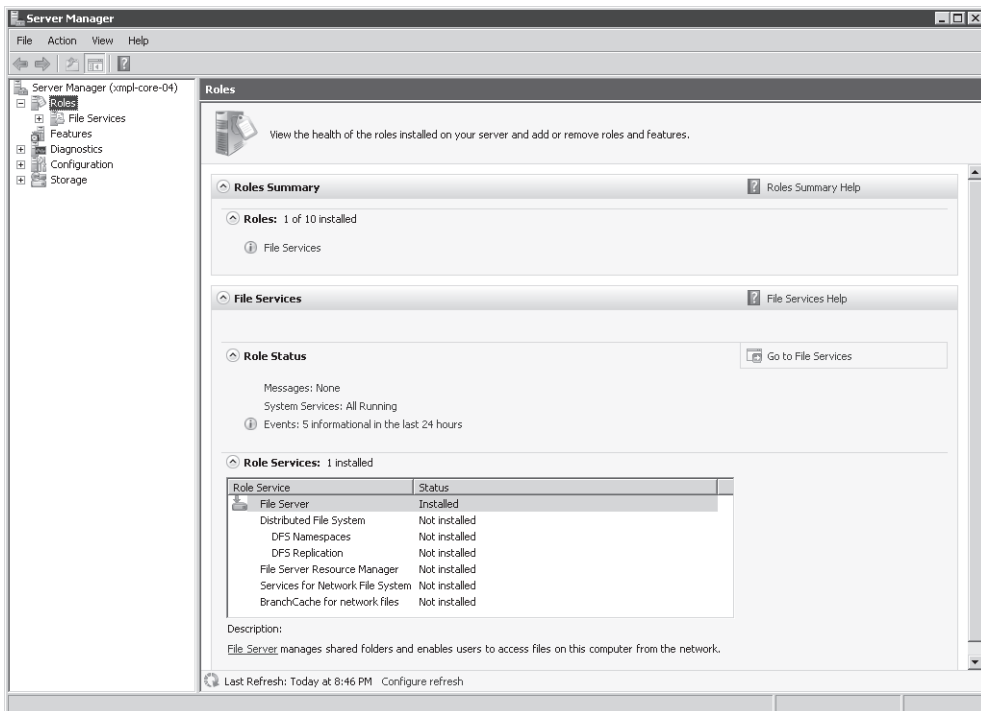


FIGURE 2-12 Server Manager connected to a remote Server Core computer.

Hyper-V: Scaling and Migrating Virtual Machines

- The Strategic Role of Virtualization 25
- Licensing 26
- Deploying and Managing Virtual Machines 27
- Managing Virtual Machine Storage 35
- Live Migration 37

One of the most highly anticipated and far-reaching changes that was part of Windows Server 2008 was the inclusion of virtualization in the form of the Hyper-V role. Hyper-V provides a fast, fully supported, hypervisor-based virtualization solution that gives you the flexibility to manage your IT resources more effectively and efficiently. In this chapter we look at the changes to Hyper-V for Windows Server 2008 R2 and how to most effectively use them in your environment.

The key areas of improvement in Windows Server 2008 R2 Hyper-V are as follows:

- **Scalability** Hyper-V now supports up to 64 logical processors per physical host.
- **Availability** Hyper-V now supports live migration of virtual machines (VMs) using Clustered Shared Volumes (CSVs).
- **Efficiency** Hyper-V now supports improved networking.
- **Flexibility** Hyper-V now supports dynamic addition or removal of storage.

The Strategic Role of Virtualization

Even as recently as a few years ago, virtualization was something that IT people talked about, but it simply wasn't a significant part of most IT infrastructures. Most virtualization either was very expensive or was based on using an emulation layer of software that ran on top of the operating system. This meant that performance was less than ideal, and most applications weren't supported running in a virtualized environment.

With the release of Hyper-V, however, that changed dramatically, and nearly everyone in IT is actively investigating or already deploying virtualized servers and applications.

Microsoft fully supports virtualization for most products (see the Microsoft Knowledge Base article 957006 at <http://support.microsoft.com/kb/957006/> for the current virtualization support policies for Microsoft server software).

So why the rush to virtualize? There are several different reasons why organizations choose virtualization, but the two we hear as the most common drivers are the following:

- Flexibility
- Server consolidation and utilization

Virtualization gives you the flexibility to quickly create test environments, to move workloads from one server to another, and to rapidly deploy additional VMs to meet changing requirements. It also gives you a far greater degree of hardware independence, as the virtualized workload sees a consistent virtualized hardware across a wide range of physical hardware.

Virtualization plays an important role in enabling organizations to make the most effective use of their hardware resources. By virtualizing multiple workloads onto a single physical server, each in its own VM, you can take advantage of underutilized computer resources while simplifying the overall management of your infrastructure.

Virtualization also helps you save money. By having fewer physical computers, you reduce your energy consumption, datacenter space requirements, and hardware support costs, while also reducing your carbon footprint, a not insignificant consideration these days.

Licensing

Windows Server 2008 R2 Hyper-V requires no additional licensing to use on those editions in which it is available. There are no additional Client Access Licenses required, either. The Hyper-V role is not available for Windows Server 2008 R2 Itanium, Windows Server 2008 R2 Web, or Windows Server 2008 R2 Foundation.

When using the Hyper-V role to virtualize other workloads, you have additional “virtual use rights” that vary by which edition of Windows Server 2008 R2 you are using. If you’re using Windows Server 2008 R2 Standard on the physical host computer, and you don’t enable any roles other than the Hyper-V role, you have a license to run a second copy of the Windows Server software virtualized on that physical server. This is sometimes referred to as “1+1” licensing.

With Windows Server 2008 R2 Enterprise, the licensing is 1+4 licensing—you can run four instances of the Windows Server software virtualized on the physical computer that is licensed with Windows Server 2008 R2 Enterprise, so long as the physical instance is only used to manage the virtual instances.

With Windows Server 2008 R2 Datacenter, you have an unlimited license to run virtualized instances of the Windows Server software. This can make the price of Windows Server 2008 R2 Datacenter a compelling bargain in heavily virtualized environments.

NOTE If you enable additional roles beyond the Hyper-V role, or use the instance of Windows Server 2008 R2 running on the physical computer for additional workloads such as file or print serving, you no longer have the “1+” use rights just described.

Deploying and Managing Virtual Machines

As with most things Windows, there are multiple ways to deploy, manage, and configure VMs running on Hyper-V. You can use the Hyper-V Manager console, either locally (if you’re running a full server installation, not Server Core) or remotely. You can use Windows PowerShell cmdlets. You can use the Failover Cluster Manager if the Hyper-V server is part of a Windows failover cluster, or you can use System Center Virtual Machine Manager (SCVMM) 2008 R2, as shown in Figure 3-1.

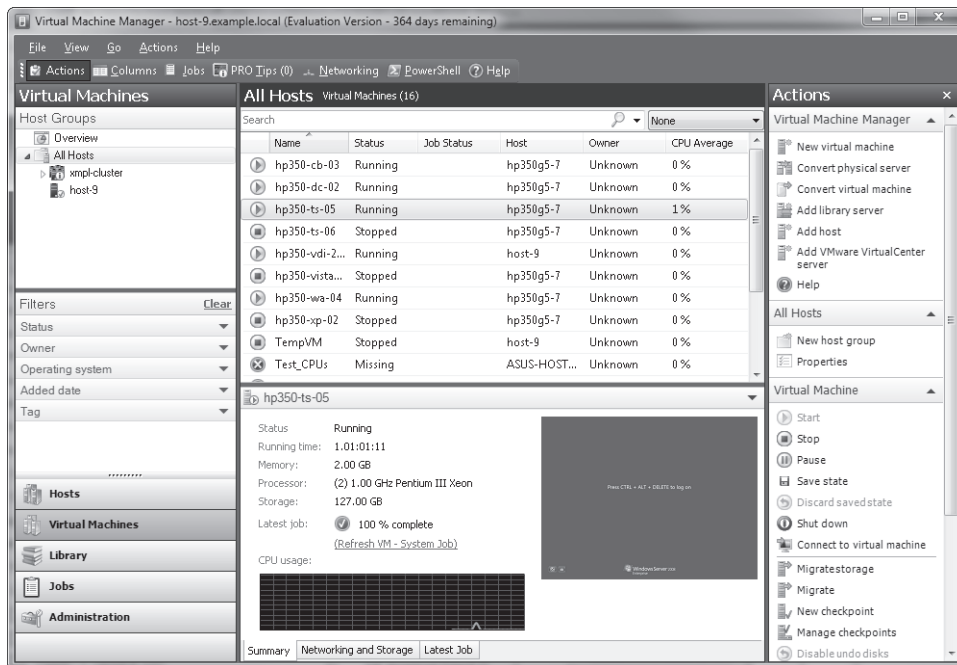


FIGURE 3-1 System Center Virtual Machine Manager 2008 R2.

Hyper-V Manager Console

The Hyper-V Manager console is the default way to manage nonclustered Hyper-V nodes. It integrates into the Server Manager console or runs stand-alone as shown in Figure 3-2.

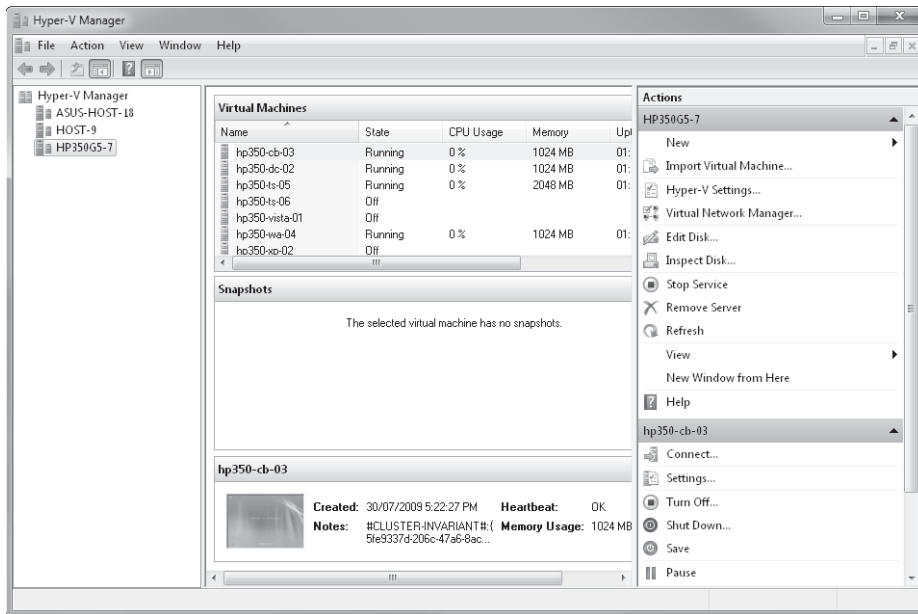


FIGURE 3-2 The Hyper-V Manager console.

With the Hyper-V console, you can manage all aspects of a VM except clustering. You can add or delete VMs, add networks, change the settings on a VM, export it, take a snapshot, and all the other things you need to do to a VM. In a book like this, we can't show all the steps for managing VMs, but for those new to Hyper-V, we'll give you the highlights.

When you add the Hyper-V role to a computer running Windows Server 2008 R2, the Add Roles Wizard includes the basic networking setup. One key requirement is to leave one network interface card (NIC) exclusively for managing the server. If you're also using iSCSI to support failover clustering, you should have an additional NIC exclusively for the iSCSI traffic. In a production environment, unlike our test environment here, you should also plan on having at least one dedicated NIC for each VM on the server. As you can see, planning for virtualization means configuring your servers with multiple NICs.

Creating a New Virtual Machine

To create a new VM, right-click the server in the tree pane of the Hyper-V Manager, and select **New** and then **Virtual Machine**, as shown in Figure 3-3.

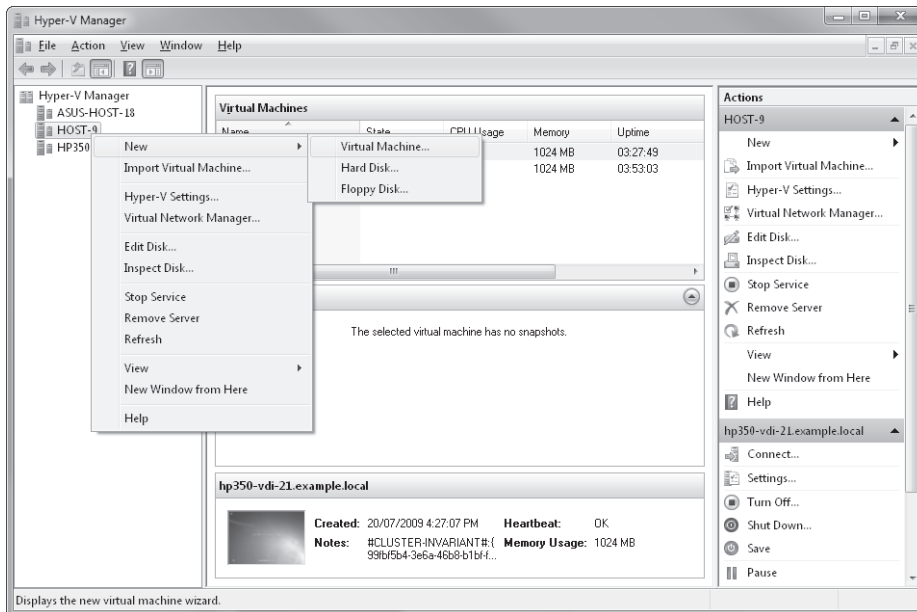


FIGURE 3-3 Using the shortcut menu to create a new virtual machine.

This launches the New Virtual Machine Wizard, shown in Figure 3-4.

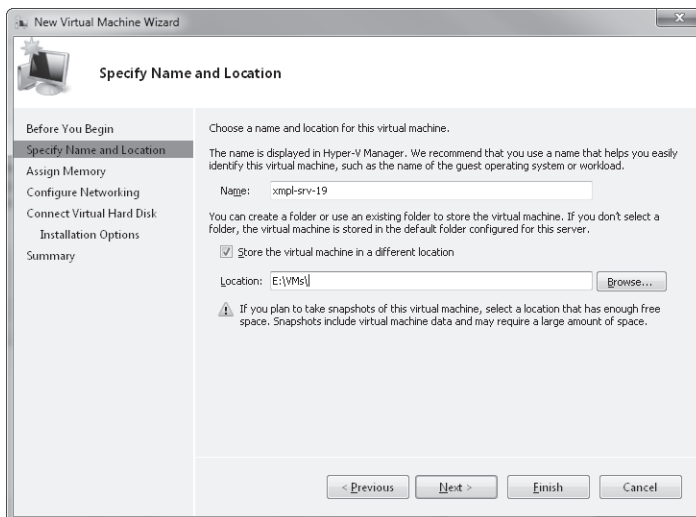


FIGURE 3-4 The New Virtual Machine Wizard.

The New Virtual Machine Wizard can create a “typical” VM, but the defaults aren’t appropriate for production environments. If you use the wizard to automatically create a new virtual hard disk (VHD), it will create a dynamically expanding VHD file, which is nice for only using the space you really need, but imposes a performance penalty as it has to periodically expand the disk space. A better option is to use the New Virtual Hard Disk Wizard, shown in Figure 3-5, to create the VHD file before you create the VM, allowing you to specify a fixed size or pass-through disk for optimal performance.

NOTE One of the areas where Windows Server 2008 R2 improves on the performance of Windows Server 2008 is dynamically expanding disks, which have been optimized in R2 to reduce the performance penalty. However, fixed-size VHDs are still recommended for production servers.

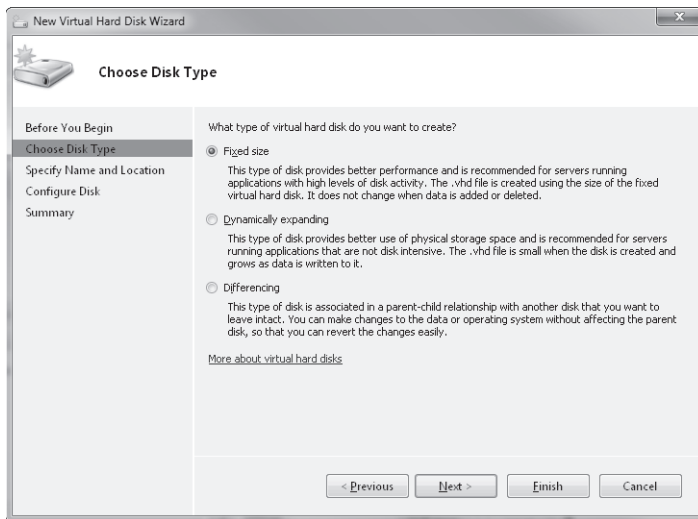


FIGURE 3-5 The New Virtual Hard Disk Wizard.

Another consideration when using the New Virtual Machine Wizard is that it automatically assigns only a single processor and a single disk to the VM it creates. Personally, we wish it would let you choose a template for the new VM, but if you do need that capability, SCVMM is a great solution.

Configuring Settings for a VM

Once you’ve created the VM with the New Virtual Machine Wizard, you’ll often need to adjust the settings for the VM. To modify them, right-click the VM in the center pane of the Hyper-V Manager, and select Settings from the drop-down menu to open the Settings dialog box

shown in Figure 3-6. In the Settings dialog box, you can change the virtual hardware that is used by the VM, along with management settings for the VM.

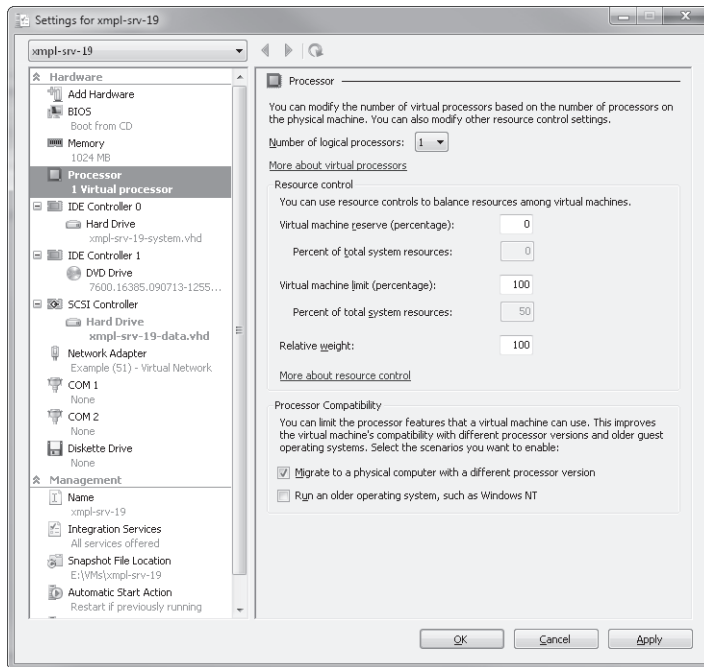


FIGURE 3-6 The Settings dialog box for the xmpl-srv-19 VM.

Only a limited subset of the virtual hardware of a VM can be modified while the VM is running or in a saved state. These include changing the connection of a network adapter (but not the number of network adapters), changing the DVD or Diskette Drive connections, and adding or removing a hard disk. This last feature is an important change in Windows Server 2008 R2 that allows you to dynamically manage the storage used by a VM.

Windows PowerShell Cmdlets

First, the bad news: Windows Server 2008 R2 does *not* include any new Windows PowerShell cmdlets for managing VMs. We're seriously disappointed by that, but fortunately, there is some good news here, too. First, if you're using Hyper-V in a Windows failover cluster, you'll get a bit of help from the new Windows PowerShell Module for failover clusters, which includes cmdlets for creating, moving, and updating clustered VMs.

Second, the PowerShell Management Library for Hyper-V, available on Codeplex at <http://pshyperv.codeplex.com/> is quite useful. The current release as of this writing is still version 1.00b, but the project is being actively maintained and updated, and if you're at all comfortable with Windows PowerShell, you should definitely be using this library.

To use the Codeplex project, download the file and unzip it to someplace where you can easily find it and where it's convenient to load it whenever you want to manage Hyper-V. We like to use the \$profile directory, which is, by default, at C:\Users\<username>\Documents\WindowsPowerShell. But here's an easy way to handle the whole thing, assuming you've unzipped the file to the C:\Users\Public\Downloads directory.

```
PSH> cd C:\Users\Public\Downloads
PSH> cp hyperv.ps1 (split-path $profile)
PSH> cd (split-path $profile)
PSH> . .\hyperv.ps1
```

This assumes, of course, that you already have a customized \$profile. If you've never made a custom profile for Windows PowerShell, the directory for it might not exist yet. Adjust the code by adding a line at the beginning:

```
PSH> mkdir (split-path $profile)
```

This will, of course, error out harmlessly if the directory already exists.

Because this project was designed to work with Windows PowerShell 1.0, the commands are implemented as functions instead of cmdlets in a module. That will likely change with the next version of PSHyperV.

PSHyperV includes more than 70 functions for managing and manipulating Hyper-V objects, including VMs, network adapters, and VHDs. A simple example of using Windows PowerShell to start a stopped VM is shown in Figure 3-7.

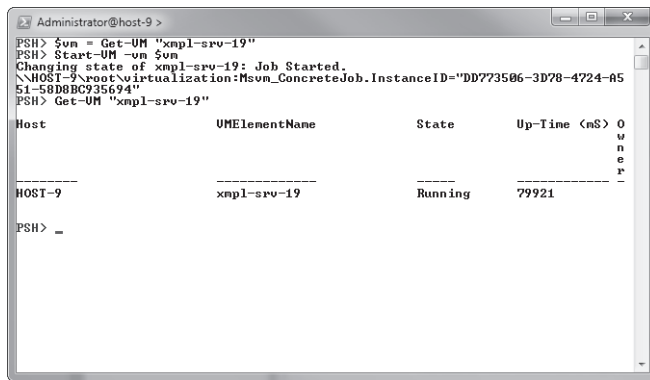


FIGURE 3-7 Using the PSHyperV project to start a virtual machine with Windows PowerShell.

Finally, if you're using SCVMM, it includes a full set of Windows PowerShell cmdlets, and even lets you easily save the underlying Windows PowerShell code from the Virtual Machine Manager console.

SCVMM 2008 R2

System Center Virtual Machine Manager 2008 R2 is the companion release of System Center for managing the v2 release of Hyper-V that is part of Windows Server 2008 R2. SCVMM 2008 R2 supports the new features of Hyper-V, including live migration, CSVs, and hot addition and removal of storage. SCVMM is more than just a tool for managing Hyper-V, however, with support for Microsoft Virtual Server 2005 R2 and VMware, including vSphere 4. New features in the R2 release of SCVMM include the following:

- Storage migration
- Queuing of live migrations
- Rapid provisioning
- Host-compatibility checks
- Third-party storage support

Storage Migration

Storage migration in SCVMM allows you to easily migrate the storage of a running VM, enabling you to migrate existing VMs to support the new CSV volumes and live migration. You can migrate the storage within a host, or across hosts, with short downtimes—on the order of a couple of minutes, depending on the speed of the network and the speed of the storage. Also, SCVMM 2008 R2 supports VMware's vMotion.

Queuing of Live Migrations

One limitation of live migration is that you can only do one at a time on a given host, either as source or target. SCVMM adds the ability to queue live migrations on a host by detecting that a migration fails because another one is in process, and relaunching the migration in the background after waiting. The wait between tries increases after each failure, up to a maximum.

Rapid Provisioning

In SCVMM 2008, creating a new VM meant copying the VHD from the library to the host over the network, using Background Intelligent Transfer Service (BITS). This could be a slow process on a busy network with large VHDs. In SCVMM 2008 R2, using Windows PowerShell, you can rapidly deploy new VMs using a local VHD file instead of the template VHD file from the library.

Host-Compatibility Checks

In migrations between hosts, the CPU and other host hardware needs to be compatible for the migration to succeed. Part of this is handled by the Processor Compatibility settings for the VM in Hyper-V, but SCVMM 2008 R2 does deep checks for compatibility using Hyper-V

and VMware compatibility check application programming interfaces (APIs). This enables users to check if a VM is compatible without having to do the migration, only to discover that the VM cannot start or run on the host.

Third-Party Storage Support

SCVMM 2008 R2 adds support for the Veritas Volume Manager as a cluster disk resource, and for third-party clustered file systems that have similar functionality to CSV.

Built on Windows PowerShell

SCVMM is built on Windows PowerShell and all operations that are available from the Virtual Machine Manager console can also be done from the Windows PowerShell command line. But an even better feature is that you can use the graphical console to help build a library of scripts that you can then modify for repeat use. When you perform an action in the console, as the last step before you execute it, you have an option to click View Script, which opens a Notepad window with the Windows PowerShell script that will be executed. For example, Figure 3-8 shows the Summary page for adding a new filesystem share to the library, and Figure 3-9 shows the Windows PowerShell code that was displayed with the View Script button.

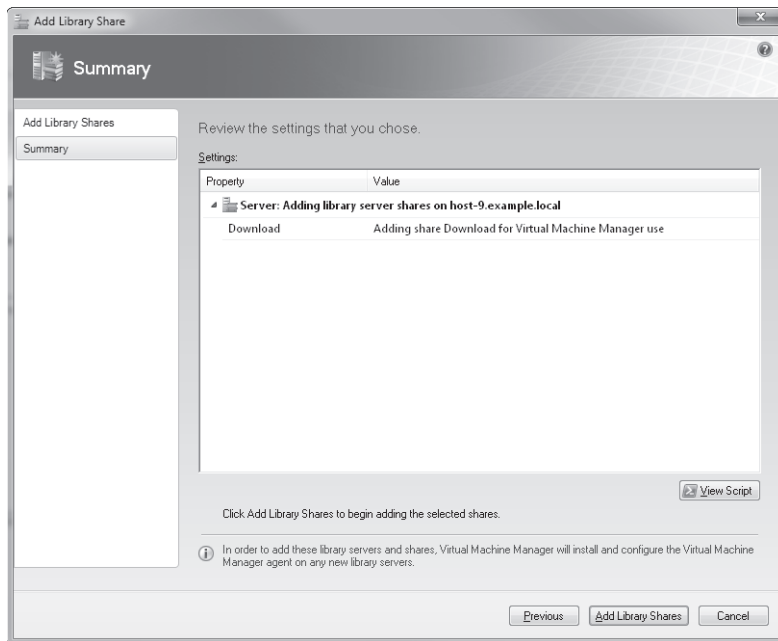
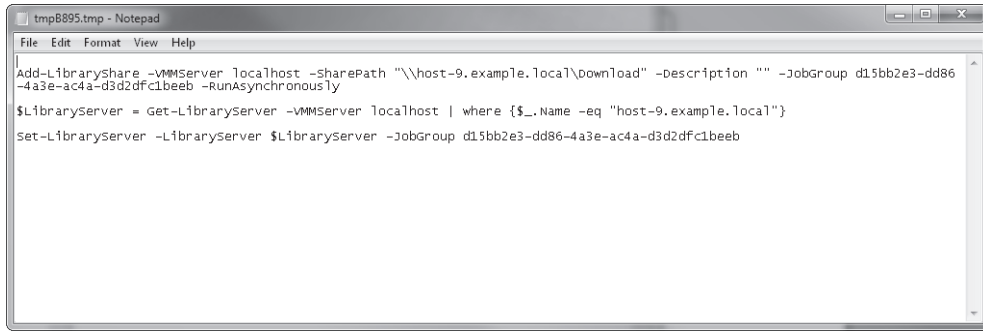


FIGURE 3-8 The Add Library Share Wizard for adding a file share to the library.



```
File Edit Format View Help
Add-LibraryShare -VMMServer localhost -SharePath "\\host-9.example.local\download" -Description "" -JobGroup d15bb2e3-dd86-4a3e-ac4a-d3d2dfc1beeb -RunAsynchronously
$LibraryServer = Get-LibraryServer -VMMServer localhost | where {$_.Name -eq "host-9.example.local"}
Set-LibraryServer -LibraryServer $LibraryServer -JobGroup d15bb2e3-dd86-4a3e-ac4a-d3d2dfc1beeb
```

FIGURE 3-9 The Windows PowerShell script for adding a share to the SCVMM library.

Managing Virtual Machine Storage

One of the big changes in Windows Server 2008 R2 Hyper-V is the increased flexibility of storage. In Windows Server 2008, you were pretty limited in your options for storage, and even if you used failover clusters, you still had limited flexibility. R2 changes that dramatically with the addition of CSVs in Windows failover clustering. Now, instead of having to have dedicated LUNs for each virtual machine, you can use CSV volumes that allow more flexible use of storage area network (SAN) resources.

Another problem in Windows Server 2008 Hyper-V was the inability to dynamically change the storage on a running VM. Unlike in the physical world where you can easily add or remove Universal Serial Bus (USB), eSATA, or iSCSI drives without shutting down a server, the only way to add or remove VHDs from a VM was to shut the VM down. In Windows Server 2008 R2 Hyper-V, this is changed and you can add or remove VHDs on a running VM. Figure 3-10 shows the Settings dialog box for the server xmpl-srv-19, which is currently running on the Hyper-V parent "host-9.example.local."

The other big change in Windows Server 2008 R2 Hyper-V is support for the new CSV volumes when using failover clustering. These volumes don't get added as drive letters, but are shown as mount points off the system drive, as shown in Figure 3-11. CSV volumes can hold multiple VHDs from multiple VMs in the cluster, greatly simplifying storage management and improving utilization.

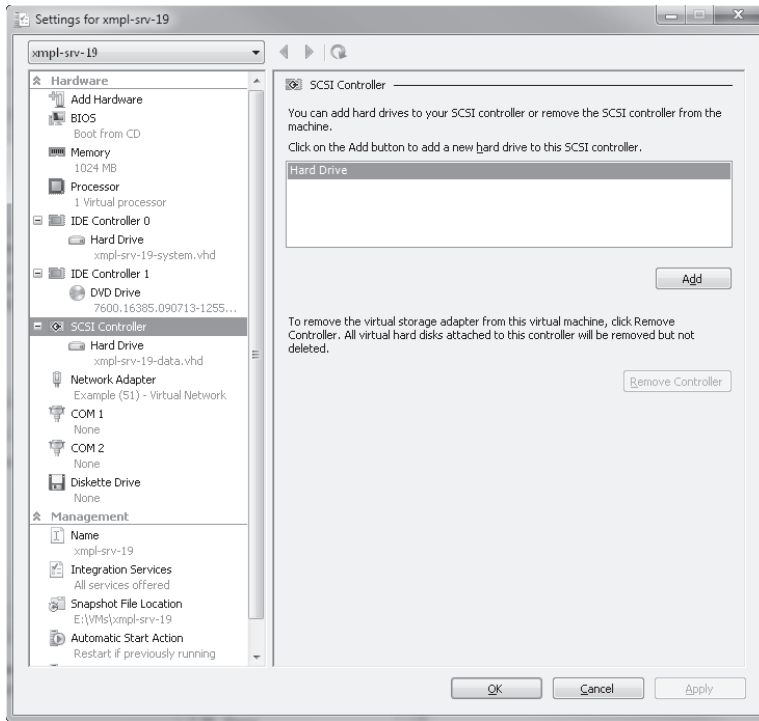


FIGURE 3-10 Adding a new virtual hard drive to the currently running xmpl-srv-19 VM.

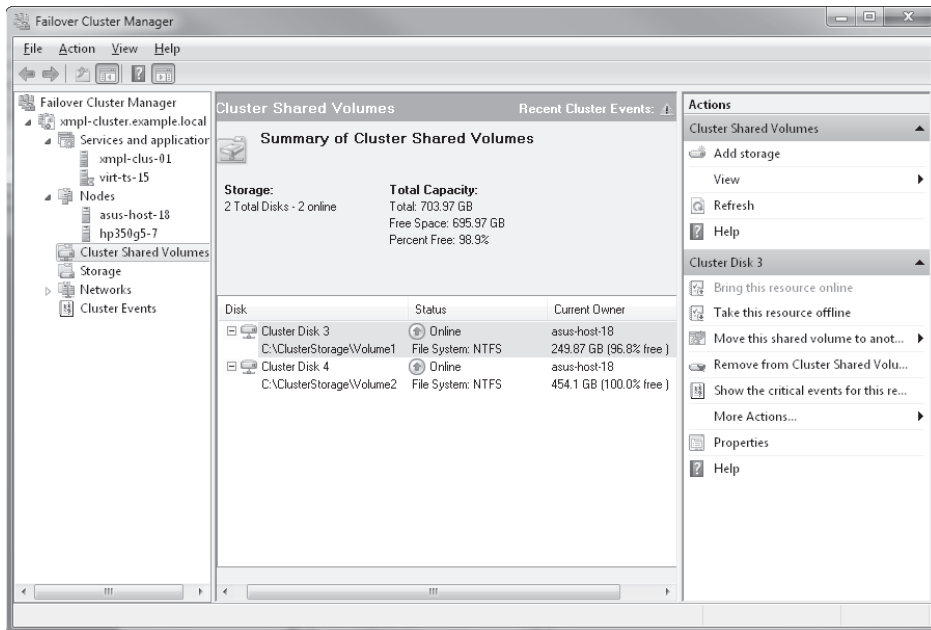


FIGURE 3-11 Cluster Shared Volumes are mounted on the system drive to ensure a common namespace across the cluster.

Live Migration

Hyper-V live migration is integrated with Windows Server 2008 R2 Hyper-V and enables running VMs to be moved from one Hyper-V physical host to another without any disruption of service or perceived downtime.

Live migration has the following benefits:

- **Provides better agility** Datacenters with multiple Hyper-V physical hosts can move running VMs to the best physical computer for performance, scaling, or optimal consolidation without affecting users.
- **Reduces costs and increases productivity** Data centers with multiple Hyper-V physical hosts can service those systems in a controlled fashion, scheduling maintenance during regular business hours. Live migration makes it possible to keep VMs online, even during maintenance, increasing productivity for users and server administrators. Datacenters can now also reduce power consumption by dynamically increasing consolidation ratios and powering off unused physical hosts during lower demand times.

Live Migration Compared to Quick Migration

Quick Migration was introduced with Windows Server 2008 Hyper-V and continues to be available with Windows Server 2008 R2 Hyper-V. Live migration and Quick Migration both move running VMs from one Hyper-V physical computer to another, but with an important difference: Quick Migration saves, moves, and restores VMs, resulting in some downtime, whereas live migration uses a different mechanism for moving the running VM to the new physical computer. Briefly, live migration uses the following process:

1. A snapshot of the running VM's memory pages is taken and the pages are transferred from the source Hyper-V physical host to the target Hyper-V physical host. During this process, any VM modifications to the VM's memory pages are tracked.
2. Any page modifications that occurred during step 1 are transferred to the destination physical computer.
3. Hyper-V moves the storage handle for the VM's VHD files to the destination physical computer.
4. The destination VM is brought online on the destination Hyper-V server.

A live migration results in significantly less downtime for the VM being migrated compared to a Quick Migration or a simple move, making it preferable when users need uninterrupted access to the migrating VM. Because a live migration completes in less time than the Transmission Control Protocol (TCP) timeout for the migrating VM, users experience no outage for the migrating VM during steps 3 and 4 of the migration.

Processor Compatibility Mode: Migrating Between Hosts with Different Processors

In a cluster where all the nodes of the cluster are exactly the same, hardware migration is fairly straightforward. There are no concerns about differences in hardware, and especially no concerns about different capabilities of the CPUs. Because Hyper-V can take advantage of the processor capabilities in the newest Intel and AMD processors to improve the overall speed and efficiency of the VMs running on the physical host, the default is to use whatever processor features are available on the original host when the VM is created. With identical processors, both live migration and Quick Migration work as expected.

When a cluster includes nodes with different processors, the capabilities of the processors can be different. Because a migration occurs with a running machine, this can cause a failure when the VM tries to run after migrating to a different processor. Applications use the x86 CPUID processor instruction to determine processor type and processor features. When Processor Compatibility Mode is used, Hyper-V hides processor features by intercepting a VM's CPUID instruction and clearing the returned bits corresponding to the hidden features.

Use the Processor Compatibility Mode only in cases where VMs will migrate from one Hyper-V-enabled processor type to another within the same vendor processor family. Processor Compatibility Mode does not enable migrations between AMD and Intel-based hosts. Processor Compatibility Mode is not needed for VM moves that involve a stop and restart of the VM. This includes unplanned failovers and manual VM moves between hosts.

To enable Processor Compatibility Mode on existing VMs, you need to shut down the VM and change the Processor setting for the VM, selecting the Migrate To A Physical Computer With A Different Processor Version check box, as shown earlier in Figure 3-6.

Configuring a VM for Live Migration

The process of configuring a VM to enable live migration involves multiple steps and requires that Windows Failover Clustering be up and running. We don't have the space in this book to cover all the steps required to set up a live migration, but it's worth looking at what's involved and walking through the parts of the process that are new to Windows Server 2008 R2. The basic steps of the process are as follows:

- Create a failover cluster of two or more nodes.
- Enable CSVs on the cluster.
- Assign cluster storage to be CSV.
- Create a new VM using the Failover Cluster Manager with all storage on the CSV volumes.

Failover clusters in Windows Server 2008 R2 can include nodes with different processors, and even processors from different manufacturers. Live migration, however, requires that the processors at least be from the same manufacturer. This does not mean that you can't create a clustered VM if your cluster includes both AMD and Intel-based nodes, but those clustered VMs can't be configured to do live migration except to nodes that are of the same manufacturer.

Create a Failover Cluster

Before you can do live migration (or Quick Migration), you first have to configure two or more servers as a failover cluster. The basic minimum hardware requirements for a two-node cluster are the following:

- Two physical servers capable of running Windows Server 2008 R2. Ideally, these should be identical or very similar servers.
- One NIC on each node dedicated to cluster communications, on a separate subnet from other networks.
- Fibre Channel or iSCSI storage. For iSCSI, this should be on a dedicated network that uses its own NICs and subnet.
- At least two LUNs on the iSCSI or Fibre Channel shared storage, one for the "witness" or quorum disk, and one or more for CSV volumes.

The minimum number of Gigabit NICs for a supported live migration scenario is three per node. The recommended configuration uses four Gigabit NICs.

Follow the steps in the Microsoft TechNet article "Hyper-V: Using Hyper-V and Failover Clustering" at [http://technet.microsoft.com/en-us/library/cc732181\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc732181(WS.10).aspx) to get the basic Failover Clustering configured.

NOTE Microsoft supports a failover cluster solution only if all the hardware features are marked as "Certified for Windows Server 2008 R2." In addition, the complete configuration (servers, network, and storage) must pass all tests in the Validate a Configuration Wizard, which is included in the Failover Cluster Manager snap-in.

Enable Cluster Shared Volumes on the Cluster

Once the cluster has been created, you need to enable CSVs, which are not enabled by default. You should have already created at least two storage disks, one of which will be used for the witness disk.

When you use iSCSI disks, they are initially offline and not initialized even after you have them assigned to a node. You need to use the Disk Management console or Diskpart.exe to change the disks to online, initialize them, and format them with an NTFS file system.

To enable CSVs for the cluster, open the Failover Cluster Manager and connect to the cluster you've created. Right-click the cluster in the tree pane, and select Enable Cluster Shared Volumes. You'll be greeted with a clear warning as shown in Figure 3-12.

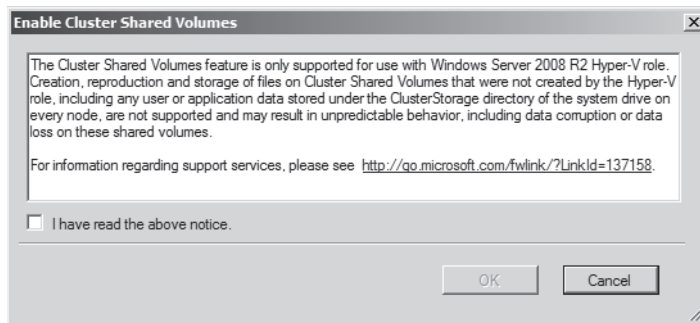


FIGURE 3-12 Cluster Shared Volumes are *only* for use with Hyper-V.

The warning about using CSVs for anything other than Hyper-V is for a good reason. Read it and understand what it means before you go any further.

Assign Cluster Storage to be CSV

Once you've enabled CSV storage, you can assign existing cluster storage to be CSV volumes, or add new storage to the cluster and move it to CSV storage. To connect to an iSCSI LUN and use it for CSV on a two-node cluster, follow these steps:

1. Create the LUN on the iSCSI SAN. The steps for this will vary depending on your SAN hardware or software.
2. From Control Panel, open the iSCSI Initiator Properties dialog box.
3. Click Refresh to show any new targets, as shown in Figure 3-13.

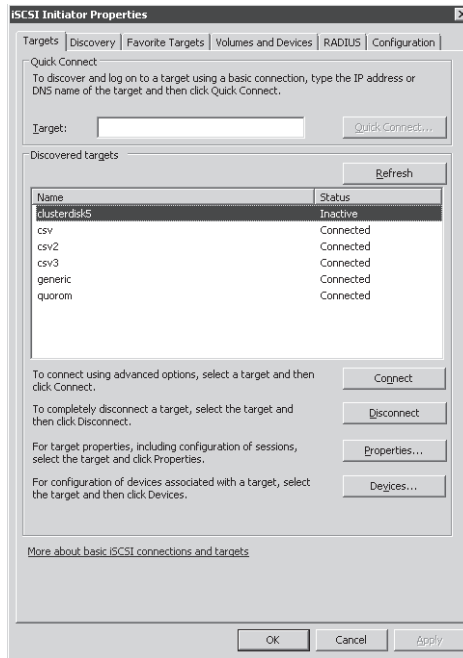


FIGURE 3-13 ClusterDisk5 is new and not yet connected.

4. Select the target you want to add to the cluster and click Connect.
5. Click OK and then click OK again to exit the iSCSI Initiator Properties dialog box.
6. Repeat steps 2 through 5 on the second node in the cluster. The iSCSI target must be connected to all nodes that will be using the disk.
7. Open the Disk Management console (Diskmgmt.msc) on either node of the cluster, and select the iSCSI disk just added, right-click, and then select Online, as shown in Figure 3-14.

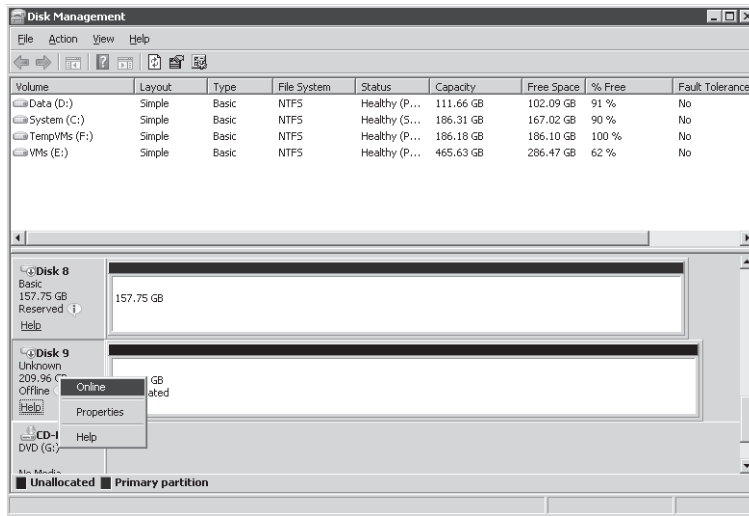


FIGURE 3-14 New iSCSI disks are offline and uninitialized.

8. Right-click again and select Initialize Disk.
9. Create a New Simple Volume and format the disk as NTFS. You don't need to assign a drive letter to the new volume.
10. Exit the Disk Management console and open the Failover Cluster Manager.
11. Select Storage in the tree pane. In the Actions pane, click Add A Disk to open the Add Disks To A Cluster dialog box shown in Figure 3-15.

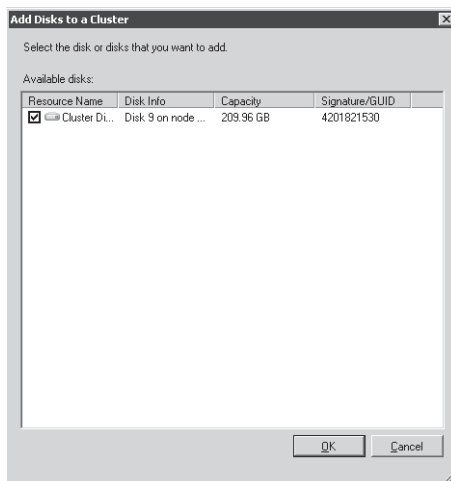


FIGURE 3-15 The Add Disks To A Cluster dialog box showing the new disk.

12. Select the disk to add and click OK. The disk will be added to the cluster in the Storage node.
13. In the tree pane, select Cluster Shared Volumes. In the Actions pane, click Add Storage to open the Add Storage dialog box shown in Figure 3-16.

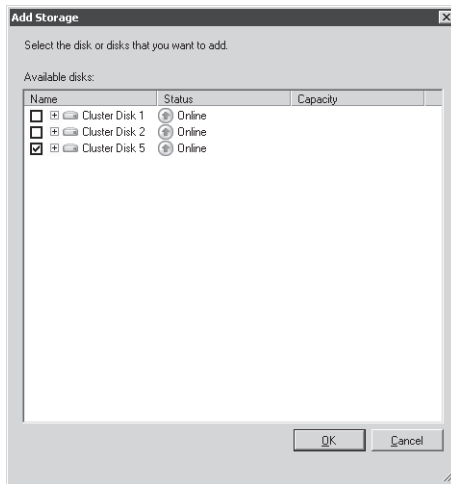


FIGURE 3-16 Adding a cluster disk to CSV storage.

14. Select the iSCSI disk just added, and click OK to move it from general cluster storage to CSV storage. The disk is now available for Hyper-V storage.

Create a New VM Using the Failover Cluster Manager

The final stage of the process is to create a new VM that is configured for live migration. To do this, you need to start the process from the Failover Cluster Manager, or use Windows PowerShell with the FailoverClusters module loaded. The basic steps are as follows:

- Create the VM.
- Assign CSV storage.
- Set Automatic Start Action to None.
- Enable High Availability for the VM.

Use the following steps to create the new VM and make it available for live migration:

1. Open the Failover Cluster Manager and connect to the cluster to which you want to add the VM.
2. In the tree pane, right-click Services And Applications, and select Virtual Machines, New Virtual Machine, and then select the initial node that will host the VM, as shown in Figure 3-17.

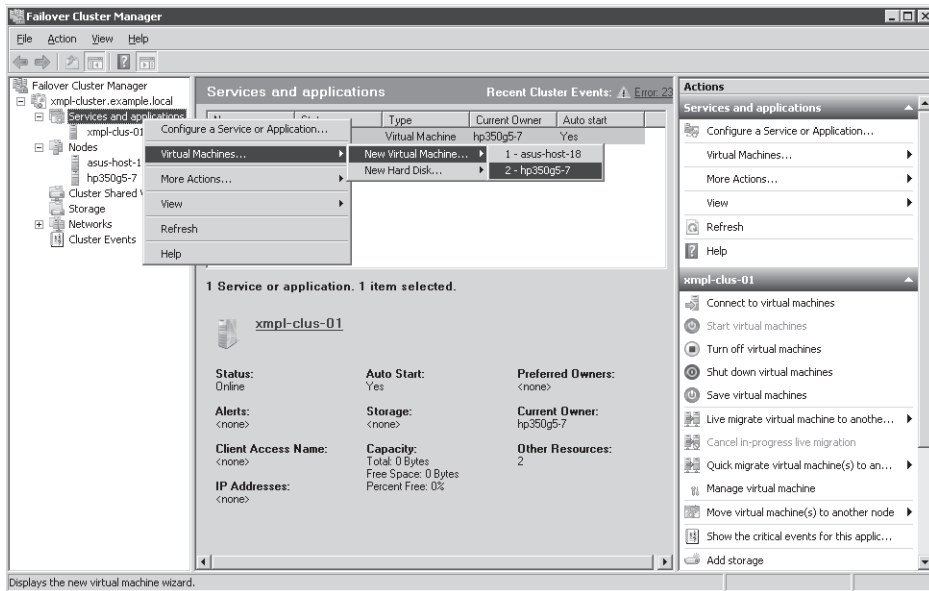


FIGURE 3-17 Creating a new virtual machine using the Failover Cluster Manager.

3. In the New Virtual Machine Wizard, specify a name and then browse to a location to store the VM files. Specify a CSV location, as shown in Figure 3-18.

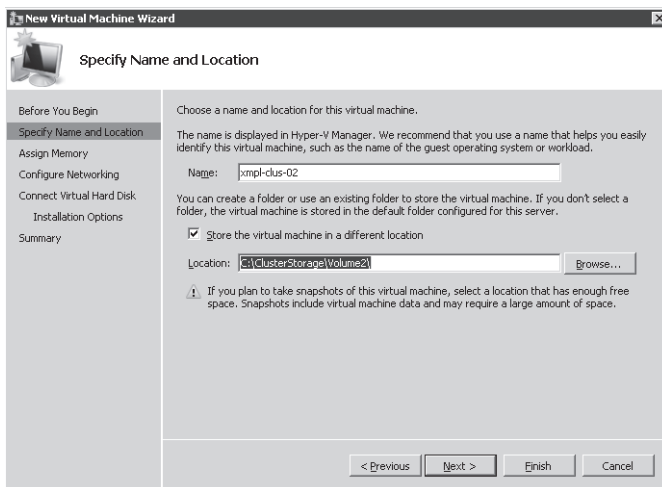


FIGURE 3-18 Specify a CSV location for the virtual machine.

4. Complete the rest of the New Virtual Machine Wizard. The VHD you specify must also reside on CSV.

5. When the New Virtual Machine Wizard completes, it will launch the High Availability Wizard, as shown in Figure 3-19. If everything worked correctly, you'll have a success report, as shown. If not, click View Report to identify the problem and correct it.

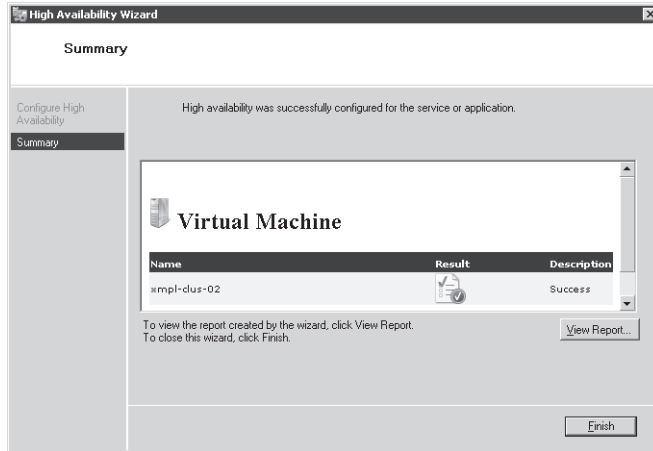


FIGURE 3-19 The High Availability Wizard.

6. Open the Hyper-V Manager and configure any additional settings for the new VM. Ensure that Automatic Start actions are set to None and that all storage is on CSV storage.
7. Start the VM and install an operating system as you would normally. The VM is configured to support live migration.

Once a VM is configured to support live migration, the process is simple. Open the Failover Cluster Manager, select the VM to migrate in the tree pane, and in the Actions pane click Live Migrate Virtual Machine To Another Node. Specify the target node, and the migration proceeds. When it's completed, you'll see that the new owner of the VM is the target node.

Optimizing Virtual Machine Performance

Windows Server 2008 R2 improves overall Hyper-V performance significantly compared to Windows Server 2008, while reducing power consumption and allowing greater VM density per physical host. The two main areas of performance improvement are scalability and networking.

Scalability Improvements in VM Performance

Windows Server 2008 R2 supports up to 64 logical processors on the physical host computer. This allows greater VM density per physical host, reducing costs and power consumption, and gives IT administrators greater flexibility in assigning CPU resources to VMs. Also,

Hyper-V now supports Second Level Address Translation (SLAT), which uses new features on today's CPUs to improve VM performance while reducing processing load on the Windows Hypervisor.

Power consumption of the VM physical host is also reduced because of Windows Server 2008 R2's support for core parking, which allows unused processor cores to be dynamically turned off and on according to the processor requirements and load.

Networking Improvements in VM Performance

Hyper-V in Windows Server 2008 R2 uses several new networking technologies to improve overall VM networking performance. The three key areas of improvement are the following:

- New VM Chimney (also called TCP Offload)
- Support for Jumbo Frames
- Support for the Virtual Machine Queue (VMQ)

VM Chimney allows a VM to dump its network processing load onto the NIC of the host computer. This works the same as in a physical TCP Offload scenario; Hyper-V now simply extends this functionality into the virtual world. This benefits both CPU and overall network throughput performance and is fully supported by live migration.

VM Chimney is disabled by default in Windows Server 2008 R2. VM Chimney requires compatible networking hardware but can significantly reduce the host server's CPU burden when dealing with VM network traffic. This translates into better host system performance and a simultaneous boost to VM network throughput.

Support for Jumbo Frames was introduced with Windows Server 2008. Hyper-V in Windows Server 2008 R2 simply extends this capability to VMs. Jumbo Frames support in Hyper-V adds the same basic performance enhancements to virtual networking, including up to six-times-larger payloads per packet, which not only improves overall throughput but also reduces CPU utilization for large file transfers.

VMQ essentially allows the host's single NIC card to appear as multiple NICs to the VMs by allowing the host's NIC to direct memory access (DMA) packets directly into individual VM memory stacks. Each VM device buffer is assigned a VMQ, which avoids needless packet copies and route lookups in the virtual switch. The result is less data in the host's buffers and an overall performance improvement in input/output (I/O) operations.

Remote Desktop Services and VDI: Centralizing Desktop and Application Management

- (Re)introducing Remote Desktop Services and VDI 47
- Providing a Rich Remote Desktop 48
- Enabling VDI 55
- Integrating Remote and Local Applications with RemoteApp 58
- Working Over the Web: Web Access 59
- Licensing 60

Windows Server 2008 R2 includes a major name change: Terminal Services becomes Remote Desktop Services, and all the related services change accordingly. Table 4-1 shows the name changes, but what's really important to understand is that this is much more than just a name change; it is a clear change in focus and functionality. The major technological addition in R2 is full support for Virtual Desktop Infrastructure (VDI) and the ability to have applications and whole desktops virtualized.

(Re)introducing Remote Desktop Services and VDI

Remote Desktop Services includes all the functionality of Windows Terminal Services, and quite a bit more as well. Windows Server 2008 introduced major changes in Terminal Services, especially the ability to integrate a remotely running application into your local desktop with TS RemoteApps. In Windows Server 2008 R2, RemoteApps is extended to provide a more nearly seamless experience, with the ability to have the full Windows Aero experience for remote applications.

Another major change is the addition of the Remote Desktop Virtualization Host (RD Virtualization Host) service, a role service of the Hyper-V role. RD Virtualization Host

works with the rest of the Remote Desktop services to provide virtual desktops to users. For users who need a consistent but personal desktop, Windows Server 2008 R2 can provide a personal virtual desktop regardless of which computer you are using. It can also provide a standard corporate desktop from a pool of virtual desktops to users on demand.

TABLE 4-1 Windows Server 2008 R2 Remote Desktop Services Naming

WINDOWS SERVER 2008 R2 NAME	WINDOWS SERVER 2008 NAME
Remote Desktop Services	Terminal Services
Remote Desktop Session Host (RD Session Host)	Terminal Server
Remote Desktop Virtualization Host (RD Virtualization Host)	No equivalent
Remote Desktop Connection Broker (RD Connection Broker)	Terminal Services Session Broker
Remote Desktop Web Access (RD Web Access)	Terminal Services Web Access
RemoteApp	TS RemoteApp
Remote Desktop Gateway	TS Gateway
Remote Desktop Client Access License (RD CAL)	TS CAL
Remote Desktop Easy Print	Terminal Services Easy Print

Providing a Rich Remote Desktop

Remote Desktop Services provides an improved and more seamlessly integrated remote experience to the user. Remote applications can now take full advantage of multiple monitors, the Windows Aero look and feel, and a full audio experience, while also integrating more seamlessly into the Taskbar, Start menu, and system tray.

Administration and management of RemoteApps and of virtual desktops is improved in Windows Server 2008 R2, with the addition of a Windows PowerShell module (including a Windows PowerShell provider), and an improved RD Web Access Configuration using the RemoteApp and Desktop Connection Web application shown in Figure 4-1.

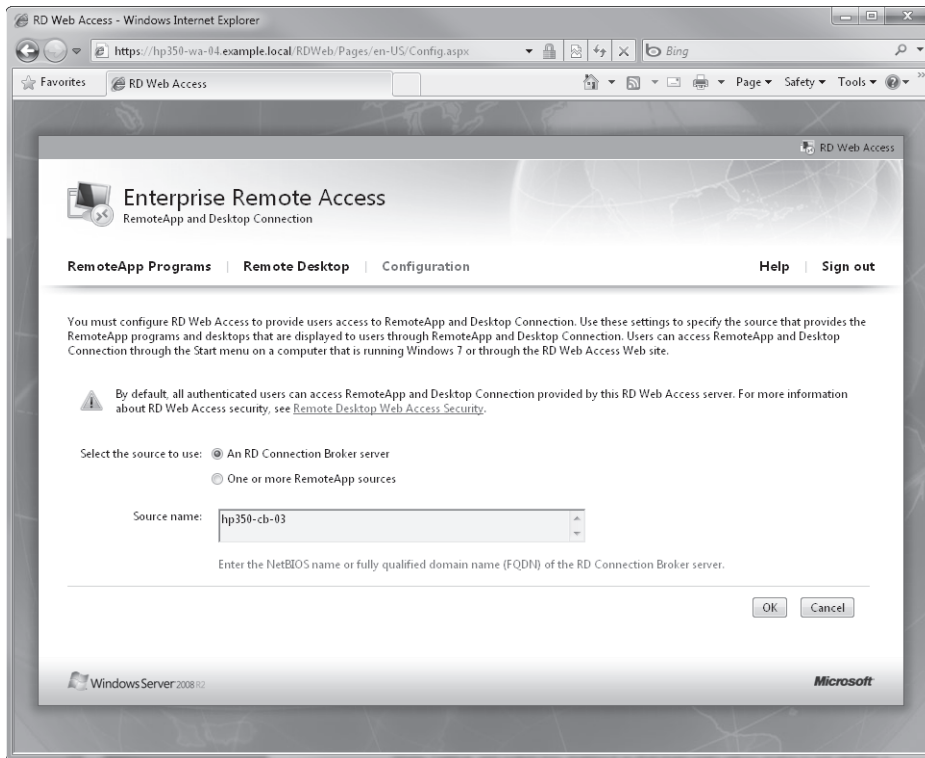


FIGURE 4-1 The RemoteApp and Desktop Connection application.

The RemoteApp and Desktop Connection can be customized to meet your needs, but defaults to a name of Enterprise Remote Access, as shown.

Remote Desktop Administration and Management

The RemoteApp and Desktop Connection Web application gives IT administrators a single place to manage and assign resources for their users. Changes made here are directly reflected in the RemoteApp and Desktop Connection Control Panel for Windows 7 users, and in the applications and virtual desktops that users connecting from earlier versions of Windows see when they log in to the RD Web Access server, as shown in Figure 4-2.

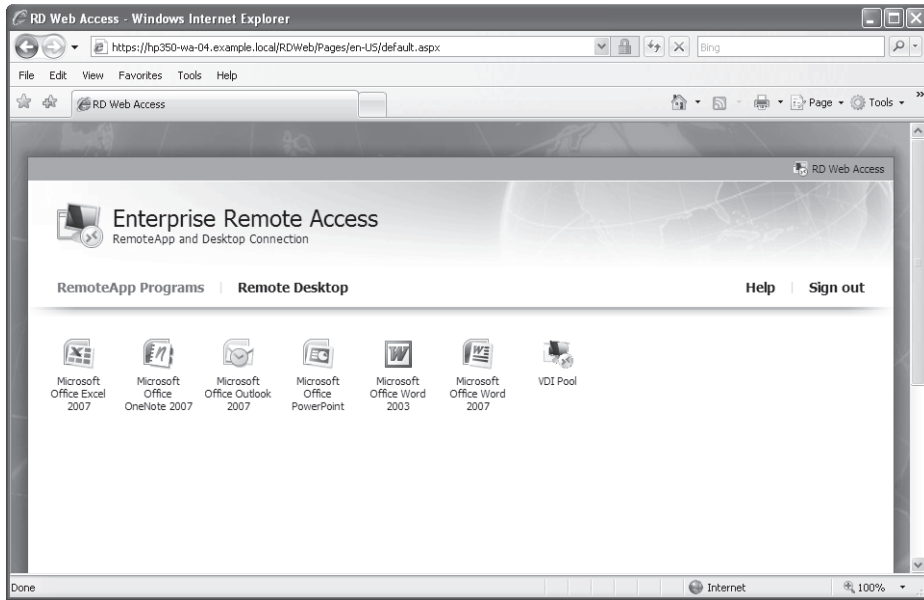


FIGURE 4-2 The RemoteApp and Desktop Connection page from a Windows XP SP3 computer.

The Windows 7 computer of the same user directly integrates these same links into the user's Start menu, as shown in Figure 4-3.

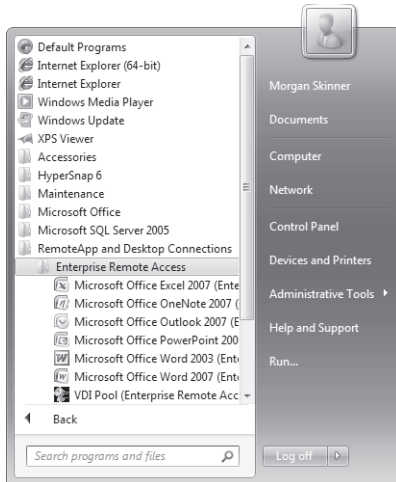
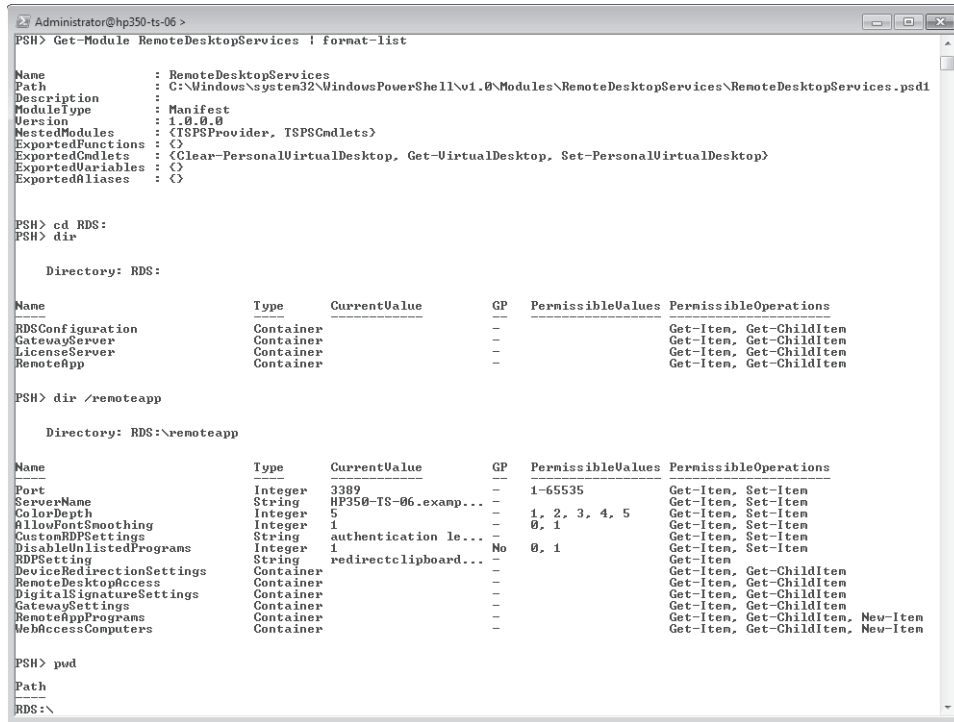


FIGURE 4-3 RemoteApp and Desktop Connections are directly integrated into the Windows 7 Start menu.

Whenever an administrator makes changes to the available programs or virtual desktops, both the RD Web Access page and the user's Start menu are dynamically updated without further intervention.

Windows PowerShell Module

Windows Server 2008 R2 includes a new Windows PowerShell module, the RemoteDesktopServices module, that includes both cmdlets and a full RDS Provider, as shown in Figure 4-4.



```
Administrator@hp350-ts-06 >
PSH> Get-Module RemoteDesktopServices | format-list

Name       : RemoteDesktopServices
Path       : C:\Windows\system32\WindowsPowerShell\v1.0\Modules\RemoteDesktopServices\RemoteDesktopServices.psd1
Description :
ModuleType : Manifest
Version    : 1.0.0.0
NestedModules : (TSPSPProvider, TSPSCmdlets)
ExportedFunctions : {}
ExportedCmdlets : (Clear-PersonalVirtualDesktop, Get-VirtualDesktop, Set-PersonalVirtualDesktop)
ExportedVariables : {}
ExportedAliases : {}

PSH> cd RDS:
PSH> dir

    Directory: RDS:

Name                Type      CurrentValue      GP      PermissibleValues PermissibleOperations
-----
RDSConfiguration    Container
GatewayServer        Container
LicenseServer        Container
RemoteApp            Container

PSH> dir /remoteapp

    Directory: RDS:\remoteapp

Name                Type      CurrentValue      GP      PermissibleValues PermissibleOperations
-----
Port                Integer    3389              -      1-65535           Get-Item, Set-Item
ServerName           String     HP350-TS-06.examp... -      0, 1             Get-Item, Set-Item
ColorDepth            Integer    5                -      1, 2, 3, 4, 5     Get-Item, Set-Item
AllowFontSmoothering Integer    1                -      0, 1             Get-Item, Set-Item
CustomRDPSettings     String     authentication le... -      0, 1             Get-Item, Set-Item
DisableUnlistedPrograms Integer    1                No     0, 1             Get-Item, Set-Item
RDPSetting            String     redirectclipboard... -
DeviceRedirectionSettings Container
RemoteDesktopAccess   Container
DigitalSignatureSettings Container
GatewaySettings       Container
RemoteAppPrograms     Container
WebAccessComputers    Container

PSH> pwd
Path
RDS:\
```

FIGURE 4-4 The RemoteDesktopServices module for Windows PowerShell includes both cmdlets and a provider.

For those new to Windows PowerShell, a brief explanation of providers is in order. In Windows PowerShell, providers are a way to view and navigate information in a hierarchical way as if the providers were drives on the computer. In fact, the FileSystem is implemented as a provider. This means that when you type **dir c:** at the Windows PowerShell prompt, what you're actually doing is asking Windows PowerShell to give you the children of the C drive of the FileSystem provider. (The **dir** command is an alias for **Get-ChildItem**.) Windows PowerShell implements the Windows Registry as a provider as well, so you issue the command **dir HKLM:\System\CurrentControlSet** to see what the HKeyLocalMachine registry hive has in the System\CurrentControlSet container.

With the RemoteDesktopServices provider, the "drive" is RDS:. Beneath that top level we have RDSConfiguration, GatewayServer, LicenseServer, RDSFarms, ConnectionBroker, and RemoteApp containers. With the RDS Windows PowerShell module, you can configure and manage all RDS role services and components using Windows PowerShell. For example, you can do the following:

- View and edit configuration settings of Remote Desktop Server
- Publish RemoteApp applications
- Configure License Server
- Create and configure a Remote Desktop server farm
- Configure and assign virtual Internet Protocol (IP) addresses to either sessions or applications
- Create and manage RDV (VDI) pools
- Create and manage Gateway Resource Access and Client Access policies

For example, with Windows PowerShell, you can quickly get a list of the personal virtual desktop assigned to a particular user:

PSH> import-module RemoteDesktopServices

```
PSH> $cred = Get-Credential
PSH> Get-VirtualDesktop -user example\charlie -credential $cred
```

Name	AssignedTo	Host
----	-----	----
xmpl-vdi-92.example.local	EXAMPLE\Charlie	HOST-9.example.local

Because the RDS team implemented their Windows PowerShell support primarily as a provider, it's easy to navigate and investigate the functionality available, and also easy to get help on how to do tasks. So, for example, if you want to know what the parameters are for creating a new RemoteApp using Windows PowerShell, you just ask Windows PowerShell to tell you, as shown in Figure 4-5.

You can also use Windows PowerShell to quickly get or set the value of various RDS settings, as shown in Figure 4-6.

```

EXAMPLE\Charlie@hp350-ts-06 >
RDS:\remotapp> Get-Help -path .\RemoteAppPrograms New-Item

NAME
    New-Item

SYNOPSIS
    Adds a program to the RemoteApp Programs list.

SYNTAX
    New-Item [-Path] <string[]> [-Credential <PSCredential>] [-Force] [-ItemType <string>] [-Value <Object>] [-Confirm]
    [-WhatIf] [-UseTransaction] [<CommonParameters>]

    New-Item -Name <string> [[-Path] <string[]>] [-Credential <PSCredential>] [-Force] [-ItemType <string>] [-Value <Ob
    ject>] [-Confirm] [-WhatIf] [-UseTransaction] [<CommonParameters>]

    New-Item -Name <string> -ApplicationPath <string> [-ApplicationName <string>] [-ShowInWebAccess <Integer>] [-IconPa
    th <string>] [-IconIndex <Integer>] [-CommandLineSetting <Integer>] [<CommonParameters>]

    New-Item -Name <string> -ApplicationPath <string> [-ApplicationName <string>] [-ShowInWebAccess <Integer>] [-IconPa
    th <string>] [-IconIndex <Integer>] [-CommandLineSetting <Integer>] [-RequiredCommandLine <string>] [<CommonParamet
    ers>]

DESCRIPTION
    The New-Item cmdlet creates a new item and sets its value. The types of items that can be created depend upon the l
    ocation of the item. For example, in the file system, New-Item is used to create files and folders. In the registry
    , New-Item creates registry keys and entries.

    New-Item can also set the value of the items that it creates. For example, when creating a new file, New-Item can a
    dd initial content to the file.

    When the New-Item cmdlet is used with the path 'RemoteAppPrograms', it adds a program to the RemoteApp Programs lis
    t. When you add a program to the RemoteApp Programs list, you make it available to users through Remote Desktop Ser
    vices, and it appears as if it is running on the user's local computer. You can also specify whether the program
    should be shown on the Remote Desktop Web Access portal page. To not show the program on the Remote Desktop Web Ac
    cess portal page, specify a value of 0 for the ShowInPortal parameter. To show the program on the Remote Desktop We
    b Access portal page, specify a value of 1 for the ShowInPortal parameter.

RELATED LINKS
    Online version: http://go.microsoft.com/fwlink/?linkid=143166

REMARKS
    To see the examples, type: "get-help New-Item -examples".
    For more information, type: "get-help New-Item -detailed".
    For technical information, type: "get-help New-Item -full".

RDS:\remotapp> _

```

FIGURE 4-5 Using the Get-Help command with the -path parameter to get specific help on creating RemoteApps.

```

Administrator@hp350-ts-06 >
RDS:\RDSConfiguration> dir connectionbrokersettings

    Directory: RDS:\RDSConfiguration\connectionbrokersettings

Name                Type             CurrentValue      GP    PermissibleValues  PermissibleOperations
-----
ServerPurpose       Integer          2                 No    0, 1, 2, 3         Get-Item, Set-Item
FarmName            String           0                 No    0, 1               Get-Item, Set-Item
LoadBalancingState  Integer          0                 No    0, 1               Get-Item, Set-Item
ServerWeight        Integer          100               No    100-10000          Get-Item, Set-Item
ConnectionBroker    String           hp350-ch-03.examp... No    0, 1               Get-Item, Set-Item
IPAddressRedirection Integer          1                 No    0, 1               Get-Item, Set-Item
CurrentRedirectableAddresses Container         -                 -
RedirectableAddresses Container         -                 -

```

FIGURE 4-6 Getting the ConnectionBrokerSettings.

To change the session settings to disable new connections, the command would be as follows:

```
RDS:\RDSConfiguration> Set-Item -path .\SessionSettings\AllowConnections 0
```

Windows 7 and RDS (Better Together)

Users running Windows 7 will have an enhanced user experience when using Remote Desktop. Not only will they have more direct access to applications and desktops through the RemoteApp and Desktop Connection (RAD) link in the Control Panel, but the overall experience is more natural and integrated. RemoteApps are directly integrated into the Start menu, Taskbar, and system tray, so that many users will be unable to tell whether a program is running locally or remotely.

Improved User Experience

The improved user experience with Remote Desktop Services and Windows 7 clients includes the following features:

- **Multimedia redirection** This feature provides high-quality multimedia by redirecting multimedia files and streams so that audio and video content is sent in its original format from the server to the client and rendered using the client's local media playback capabilities.
- **True multimonitor support** Remote Desktop Services enables support for up to 10 monitors in almost any size, resolution, or layout with RemoteApp and remote desktops. Applications will behave just like they do when running locally in multimonitor configurations.
- **Audio input and recording** VDI supports any microphone connected to a user's local machine and enables audio recording support for RemoteApp and Remote Desktop. This is useful for Voice over Internet Protocol (VoIP) scenarios and also enables speech recognition.
- **Windows Aero support** VDI provides users with the ability to use the Windows Aero user interface for client desktops, ensuring that remote desktop sessions look and feel like local desktop sessions.
- **DirectX redirection** Improvements in DirectX 9, 10, and 11 application rendering, and support for the new DirectX 10.1 application programming interfaces (APIs) that allow DirectX (2D & 3D) graphics to be redirected to the local client to harness the power of the graphical processing unit (GPU) on the user's local device, remove the need for a GPU on the server.
- **Improved audio/video synchronization** Remote Desktop Protocol (RDP) improvements in Windows Server 2008 R2 are designed to provide closer synchronization of audio and video in most scenarios.
- **Language bar redirection** Users can control the language setting of RemoteApp programs using the local language bar.
- **Task Scheduler** Improvements keep scheduled applications from interacting with users running RemoteApps, avoiding confusion.

RAD Control Panel

The RAD Control Panel applet, part of Windows 7, provides a simple way to configure RemoteApp and VDI directly into the user's Start menu. Plus, once the initial connection is made, applications and desktops are automatically updated as the administrator configures the available applications and desktops, simplifying management and deployment.

Configuring RemoteApp and Desktop Connection

To change the settings for RAD, use the Remote Desktop Connection Manager console, as shown in Figure 4-7.

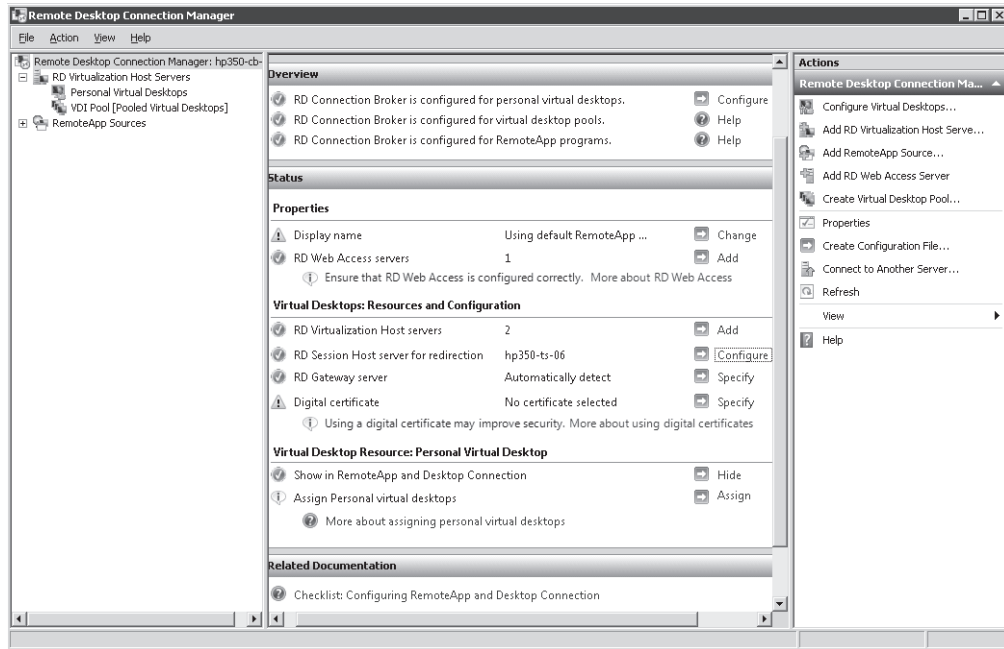


FIGURE 4-7 The Remote Desktop Connection Manager console.

The Remote Desktop Connection Manager connects to an RD Connection Broker, and allows you to configure the RD Virtualization Host servers and the personal and pooled virtual desktops they provide, along with designating the RemoteApp sources that will be available to the RD Connection Broker.

Enabling VDI

Windows Server 2008 R2 adds support for both personal and pooled virtual desktops. Enabling that VDI support requires setting up and configuring an RD Virtualization Host, an RD Session Host, an RD Connection Broker, and an RD Web Access server, although these

different roles can be combined as appropriate for your environment. The basic steps to enabling VDI are as follows:

- Enable the RD Virtualization Host role service of the Remote Desktop Services role. This will also enable the Hyper-V role.

NOTE Enabling the Hyper-V role requires hardware that supports hardware virtualization. This might require an updated BIOS. The BIOS must be configured to support both hardware virtualization and hardware Data Execution Protection.

- Enable an RD Session Host. This is required both for VDI and to provide RemoteApp programs.
- Enable an RD Connection Broker and an RD Web Access server.
- Export the Secure Sockets Layer (SSL) certificate for the RD Web Access computer. This will be imported onto the virtual machines (VMs).
- Create the VMs that will be used, configuring them as appropriate. These will be used either as part of a VDI pool or as personal VMs.
 - Add the SSL machine certificate from the RD Web Access computer to them.
 - Enable Remote Access.
 - Allow Remote RPC for RDS in the registry.
 - Enable Remote Service Management in Windows Firewall.
 - Add RDP protocol permissions to the VMs.
 - Configure the VMs for rollback if they're part of a VDI pool.
- Add the RD Web Access computer to the TS Web Access Computers local group on the RD Connection Broker.
- Add the RD Connection Broker computer as a source for the RD Web Access computer, as shown in Figure 4-8.

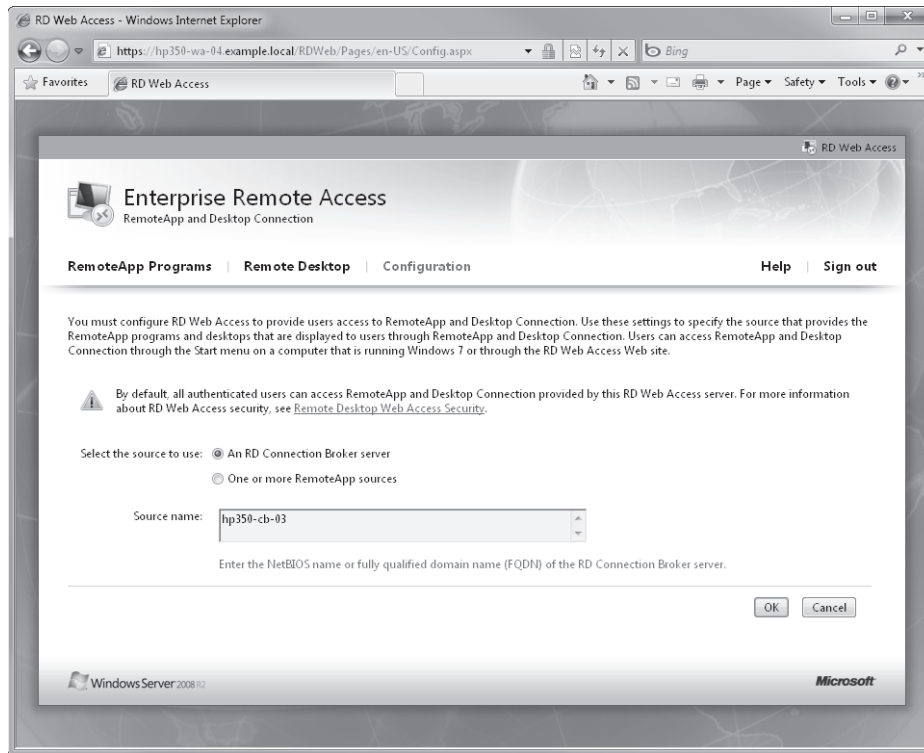


FIGURE 4-8 Configure RD Web Access to use an RD Connection Broker as a source.

- Configure the VDI Pool and assign any Personal Virtual Desktops on the RD Connection Broker as shown in Figure 4-9.

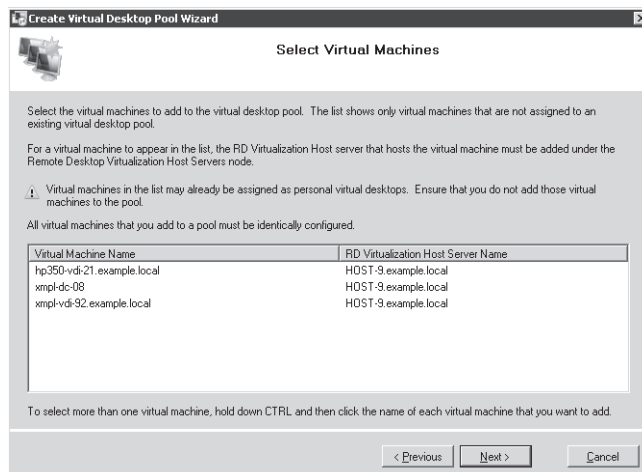


FIGURE 4-9 Adding virtual machines to a Virtual Desktop Pool.

- Add applications to the RemoteApp server and make them available as RemoteApps.
- Log on to the client computer as an administrator and import the machine SSL certificate from the RD Web Access server.
- Log on to the client computer and set up the RemoteApp and Desktop Connection.

Yes, this is fairly complicated, but most of these steps are performed one time only, or can be easily automated.

Integrating Remote and Local Applications with RemoteApp

RemoteApp for Windows Server 2008 R2 gives you the ability to provide your users with an integrated and transparent mixture of local and remote applications. For applications that behave best when run locally, or that are used when not connected to the network, you can install the applications locally, while providing access to other applications using RemoteApp where appropriate. Applications running remotely can even control the file extensions on the client computer, providing a transparent experience for the user.

To configure remote applications to take over the local file extensions, you need to create a Windows Installer (.msi) package for them and install the package locally (or use Group Policy to deploy the resulting .msi package), following these steps:

1. Open RemoteApp Manager and connect to the RD Session Host that hosts the application you want to deploy.
2. Click Add RemoteApp Program to open the RemoteApp Wizard. Click Next and select the program or programs you want to add, as shown in Figure 4-10.

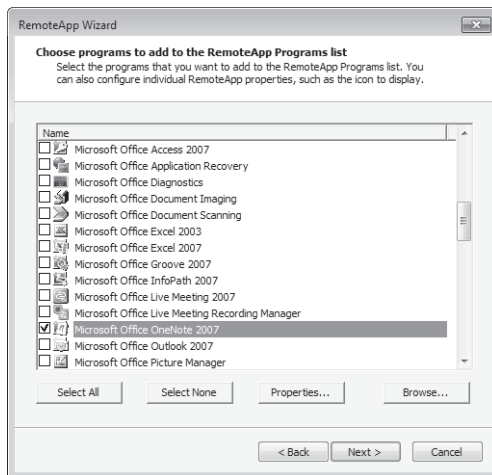


FIGURE 4-10 Adding a program with the RemoteApp Wizard.

3. Click Next and then click Finish to return to the RemoteApp Manager.
4. Select the program in the list of RemoteApp programs and click Create Windows Installer Package in the Actions pane.
5. Click Next to open the Specify Package Settings page. Make any changes here that are appropriate for your environment.
6. Click Next to open the Configure Distribution Package page shown in Figure 4-11.

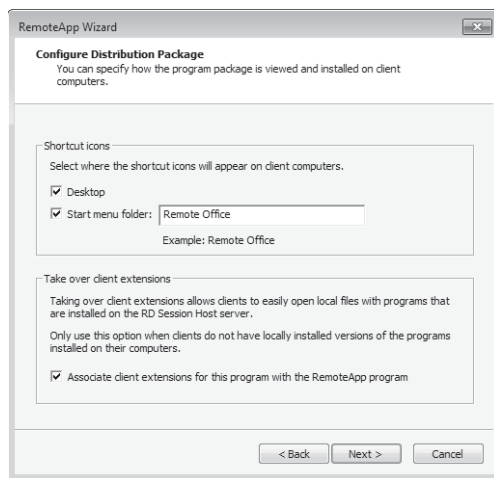


FIGURE 4-11 The Configure Distribution Package page of the RemoteApp Wizard.

7. Select the Associate Client Extensions For This Program With The RemoteApp Program check box. Also select the Desktop check box if you want the user to have a shortcut to this application on his or her desktop.
8. Click Next and then click Finish to create the .msi package, which can be installed on users' computers.

Working Over the Web: Web Access

Windows Server 2008 R2 provides access to RemoteApp programs and desktops using the RD Web Access role for all versions of Windows that support at least RDP version 6.0 or later. This includes Windows Vista SP1 and Windows XP SP3.

Users can connect to the resources of your RDS environment, including virtual desktops, from supported clients using direct RemoteApp and Desktop Connection, or over the Web using the Remote Desktop Gateway. This enables users to have consistent access to corporate resources without having to use a virtual private network (VPN) connection. Figure 4-12 shows the typical RD Web Access connection through an RD Gateway.

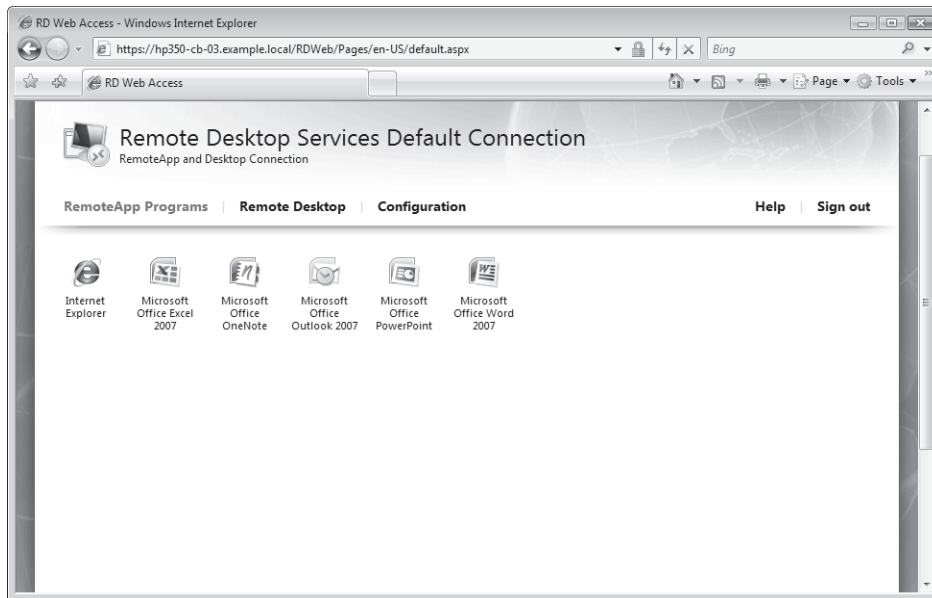


FIGURE 4-12 The RD Web Access connection through an RD Gateway.

RD Web Access can be configured to provide virtual desktops as well as RemoteApps, and also provides a gateway to allow users to connect to their own workstation if corporate policy allows it.

Licensing

The Remote Desktop Services role and its role services are included as part of the standard Windows Server license and do not require any additional licenses. The RD Session Host license is also covered by the Windows Server license with the same considerations as any other Hyper-V host.

Each user or device that directly or indirectly accesses a computer running Windows Server to interact with a remote graphical user interface (using the Windows Server 2008 R2 RDS functionality or other technology) must have a Windows Server 2008 R2 RDS Client Access License (CAL) in addition to the Windows Server CAL. RDS functionality is considered those features or services that are enabled with the RDS role and/or role service(s) in Windows Server 2008 R2. This includes, but is not limited to, RD Gateway, RemoteApp, RD Web Access, and RD Connection Broker.

RDS CALs are available as *Per User* or *Per Device* CALs. RD Session Host servers are configured for Per Device or Per User mode, and require an appropriate RDS CAL for access.

Each Per Device RDS CAL allows one device to connect to the RDS resources, regardless of how many users use the device. Conversely, a Per User RDS CAL allows a single user access to the RDS resources from as many devices as he or she happens to have. Companies should carefully consider their users and the type of devices and access they need to RDS resources before purchasing CALs and deciding what mode RD Session Host servers will use.

Windows Server 2008 R2 RDS CALs and Windows Server 2008 TS CALs are equivalent and can be used interchangeably. However, RDS CALs can only be managed from Windows Server 2008 SP2 Terminal Server License servers or Windows Server 2008 R2 RD License servers. The RD Licensing Manager, shown in Figure 4-13, adds important new capabilities, including the ability to automatically migrate licenses and dynamically activate or deactivate license servers.

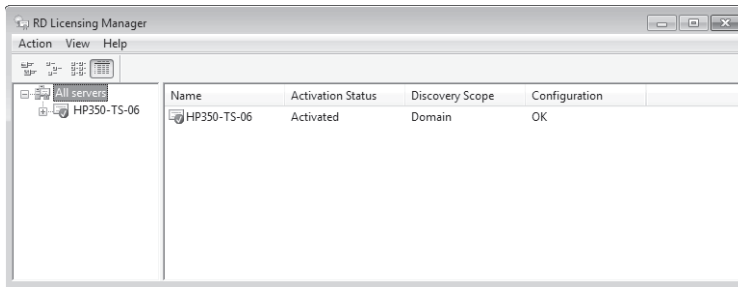


FIGURE 4-13 The RD Licensing Manager.

License Server Assignment and Activation

Windows Server 2008 R2 changes how RDS Session Hosts locate and connect to license servers. In Windows Server 2008, Terminal Servers used a discovery mechanism to find and connect to a license server. This created problems if the license server was unavailable, or if the discovery process encountered problems, and it became the source of a significant number of support calls. In Windows Server 2008 R2, this is changed so that RD Session Hosts explicitly specify the RD License servers they will connect to, as shown in Figure 4-14, and when a specific license server is unavailable, licenses are automatically migrated.

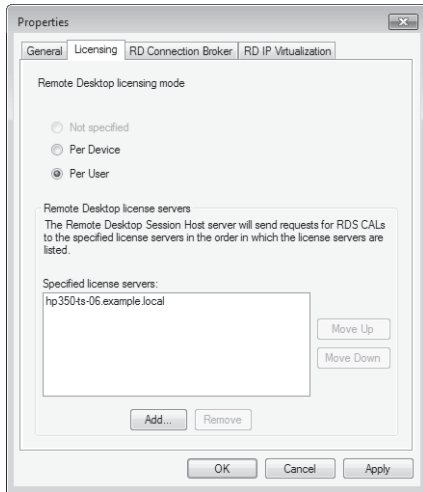


FIGURE 4-14 The RD Licensing Server is specified in the RD Session Host Configuration console.

When an RD Session Host is initially installed, it has a 120-day grace period before a license server needs to be specified. If no license server is specified and available at the end of that grace period, the RD Session Host will stop accepting connections. This grace period cannot be reset and is not an “evaluation” or “temporary” license.

Virtual Desktop Licensing

Complete and correct licensing of a Microsoft VDI environment requires licensing of both the Windows client operating system running in a centralized location and the infrastructure and management components that enable an end-to-end VDI environment.

Virtual Enterprise Centralized Desktop (VECD) is the license for Windows as a guest operating system in the data center. VECD is available for client devices that are covered by Software Assurance (VECD for SA), and those that are not, including devices such as thin clients. VECD or VECD for SA is required for any VDI environment running Windows as the guest operating system.

Most VDI environments also include management components such as System Center Virtual Machine Manager (SCVMM) or System Center Operations Manager to manage the environment. For licensing the infrastructure and management components of a Microsoft VDI environment, there are essentially two options: You can license the infrastructure components (RD Session Host, RD Virtualization Host, RD Connection Broker, etc.) with RDS CALs and license the management components separately; or, if you’re a Volume License customer, you have a pair of new options—the Microsoft Virtual Desktop Infrastructure Standard Suite and the Microsoft Virtual Desktop Infrastructure Premium Suite. These two suites combine the products for an optimum VDI experience in a value package.

- **Microsoft Virtual Desktop Infrastructure Standard Suite (VDI Standard Suite)** Includes the core products and CALs required to enable and manage VDI, including:
 - **Remote Desktop Services (RDS)** The RDS component of the VDI Suite is licensed solely for use in a VDI context; it does not provide a license to use session-based RDS resources.
 - **Microsoft Desktop Optimization Pack (MDOP)** This is a collection of technologies that enable desktop virtualization and management, including App-V.
 - **System Center Virtual Machine Manager (SCVMM) Client Management License** This provides centralized management of the Microsoft® Hyper-V™-based virtualization components of the VDI host.
 - **System Center Configuration Manager Standard Server Management License** This provides centralized configuration management of the (physical) VDI hosts of the VDI Suite.
 - **System Center Operations Manager Standard Server Management License** This provides centralized monitoring and performance management of the physical VDI host of the VDI Suite.
- **Microsoft Virtual Desktop Infrastructure Premium Suite (VDI Premium Suite)** Includes all the components of the VDI Standard Suite, plus the following:
 - **App-V for RDS** This provides application-level virtualization for RDS sessions.
 - **RDS** The RDS license is not use restricted to the VDI scenario only, but can also be used for session-based desktop and applications scenarios.

NOTE The System Center components of the VDI Suites are only licensed for use in a VDI scenario, and can't be used for general management of virtualization hosts with mixed workloads.

Active Directory: Improving and Automating Identity and Access

- Using Windows PowerShell with Active Directory 66
- Selecting Functional Levels in Windows Server 2008 R2 78
- Active Directory Recycle Bin: Recovering Deleted Objects 82
- Offline Domain Join: Securing and Facilitating Deployment 86
- Service Accounts 87
- Best Practices Analyzer 88

For the Windows Server 2008 release, Microsoft consolidated and renamed its various identity and access services to create the following five roles:

- Active Directory Certificate Services (AD CS)
- Active Directory Domain Services (AD DS)
- Active Directory Federation Services (AD FS)
- Active Directory Lightweight Directory Services (AD LDS)
- Active Directory Rights Management Services (AD RMS)

For Windows Server 2008 R2, these five roles remain in place and the visible Active Directory infrastructure is functionally the same. You can still install the roles the same way, by using Server Manager or Windows Optional Component Setup (Ocsetup.exe) from the command line, although the Add Roles Wizard now requires you to install the Microsoft .NET Framework 3.5.1 feature with Active Directory Domain Services, Active Directory Lightweight Directory Services, and Active Directory Rights Management Services roles. This requirement is to support the new Active Directory Web Services module.

NOTE For more information on Active Directory Web Services, see the section “Introducing Active Directory Web Services,” later in this chapter.

Once you have installed the Active Directory roles you need, you will find that all of the familiar Active Directory objects and attributes are still the same, and all of your familiar tools are still there. However, although the Active Directory roles in Windows Server 2008 R2 appear to be the same as those in Windows Server 2008, there are some substantial innovations beneath the surface, particularly in the area of Active Directory administration.

The R2 release includes a new set of tools for managing Active Directory from Windows PowerShell, a new graphical management utility that is based on those same Windows PowerShell cmdlets, and a long-requested mechanism for restoring Active Directory objects that administrators have inadvertently deleted. There is also a new facility for joining workstations to an AD DS domain when they do not have access to a domain controller, and an Active Directory implementation of the Best Practices Analyzer (BPA) technology that should be familiar to administrators of Microsoft Exchange Server.

These are all improvements that administrators can avoid entirely, if they so desire. You can skip right over this chapter if you want to and continue to work with Active Directory the way you always have on your new Windows Server 2008 R2 servers, and everything will function just as it always has. However, if you choose to persevere and examine these new features, you might find yourself approaching your Active Directory management tasks in a completely new and better way. You might even learn to love the command prompt.

Using Windows PowerShell with Active Directory

As in many other areas of its operating system, Windows Server 2008 R2 leverages Windows PowerShell as a major new management tool for Active Directory. Windows Server 2008 R2 includes no fewer than 85 new cmdlets for AD DS and AD LDS, which are designed to replace the existing (non-Windows PowerShell) command prompt tools, such as Dsget.exe, Dsmmod.exe, and Dsadd.exe. For administrators not comfortable working from the command prompt, Windows Server 2008 R2 also includes Active Directory Administrative Center (ADAC), a new management console that provides a graphical interface to the functionality of the Windows PowerShell cmdlets.

Using Active Directory Module for Windows PowerShell

You have already read about the enhanced capabilities of Windows PowerShell 2.0 in Chapter 1, “What’s New in Windows Server 2008 R2,” and you have seen some of what Windows PowerShell can do with Hyper-V and Remote Desktop Services in Chapter 3, “Hyper-V: Scaling and Migrating Virtual Machines,” and Chapter 4, “Remote Desktop Services and VDI: Centralizing Desktop and Application Management.” Another major innovation in Windows Server 2008 R2 is the ability to use Windows PowerShell cmdlets to manage the AD DS and AD LDS roles.

Windows Server 2008 R2 implements the cmdlets for Active Directory management as a Windows PowerShell module called ActiveDirectory. A Windows PowerShell 2.0 module is a self-contained unit consisting of cmdlets, scripts, or other code that you must import into a Windows PowerShell session before you can access its features.

Importing the Active Directory Module

When you add the AD DS or AD LDS role on a computer running Windows Server 2008 R2, the system installs the Active Directory Module for Windows PowerShell and creates a shortcut with the same name in the Administrative Tools program group. This shortcut launches the Windows PowerShell environment and uses the Import-Module cmdlet to load the Active Directory module. You can also import the module manually from a standard Windows PowerShell prompt by using the following command:

```
Import-Module ActiveDirectory
```

Once you have imported the module, the Active Directory cmdlets it contains become available, but only within that Windows PowerShell session. If you open up another Windows PowerShell window (without importing the module), the Active Directory cmdlets are not available in that session.

Using the Active Directory Module Cmdlets

Active Directory Module for Windows PowerShell contains 90 cmdlets not found in a standard Windows PowerShell session. Most (but not all) of the cmdlets in the module include the initials AD as part of their names, so you can list them using the following command:

```
Get-Command *-AD*
```

The Active Directory cmdlets, which you can use individually or combine using the standard PowerShell piping techniques, provide almost universal administrative access to AD DS and AD LDS resources. For example, to create new AD DS objects, you can use any of the following cmdlets:

- New-ADUser
- New-ADComputer
- New-ADGroup
- New-ADOrganizationalUnit
- New-ADObject

Each of these cmdlets supports parameters representing the possible attributes of the new object. For example, the New-ADUser cmdlet has 60 possible parameters, as shown in Figure 5-1, generated by the Get-Help New-ADUser command.

```

Administrator: Active Directory Module for Windows PowerShell
PS C:\Users\Administrator> Get-Help New-ADUser

NAME
    New-ADUser

SYNOPSIS
    Creates a new Active Directory user.

SYNTAX
    New-ADUser [-MobilePhone <string>] [-Initials <string>] [-Fax <string>]
    [-HomePhone <string>] [-POBox <string>] [-PostalCode <string>] [-Office
    <string>] [-ScriptPath <string>] [-ProfilePath <string>] [-LogonWorksta
    <string>] [-SmartcardLogonRequired <bool>] [-Department <string>] [-Company <string>] [-OfficePhone <string>] [-OtherAttribute
    <hashtable>] [-Organization <string>] [-EmployeeNumber <string>] [-Path <string>] [-UserPrincipalName <string>] [-AuthType (Negotiate | Basic)
    <string>] [-DisplayName <string>] [-Title <string>] [-StreetAddress <string>] [-State <string>] [-Division <string>] [-EmployeeID <string>] [-City <string>] [-Country <string>] [-AccountExpirationDate <DateTime>] [-HomePage <string>] [-Description <string>] [-AllowReversiblePasswordEncryption <bool>] [-AccountNotDelegated <bool>] [-Certificates <X509Certificate[]>] [-CannotChangePassword <bool>] [-Instance <ADUser>] [-Credential <PSCredential>] [-Server <string>] [-Type <string>] [-SAMAccountName <string>] [-WhatIf] [-PassThru] [-HomeDirectory <string>] [-Surname <string>] [-GivenName <string>] [-HomeDrive <string>] [-OtherName <string>] [-Manager <string>] [-EmailAddress <string>] [-PasswordNeverExpires <bool>] [-AccountPassword <SecureString>] [-Enabled <bool>] [-PasswordNotRequired <bool>] [-ServicePrincipalNames <string[]>] [-TrustedForDelegation <bool>] [-ChangePasswordAtLogon <bool>] [-Name <string>] [-confirm] [-whatIf] <CommonParameters>
  
```

FIGURE 5-1 Command-line parameters for the New-ADUser cmdlet.

These parameters not only enable you to create a new object, but you can also specify values for many of the object's attributes using a single command, such as in the following example:

```

New-ADUser -Name "Mark Lee" -SamAccountName "MarkLee" -GivenName "Mark"
-Surname "Lee" -DisplayName "Mark Lee" -Path 'CN=Users,DC=example,DC=local'
-OfficePhone "717-555-1212" -Title "Account Manager"
-EmailAddress "mlee@example.com" -ChangePasswordAtLogon $true
  
```

Consider how many different processes you would have to perform and how many screens you would have to access to create the user object for Mark Lee and set all the attributes defined in this example using the Active Directory Users and Computers console. For custom attributes, and those not specifically covered by a cmdlet's parameters, you can use the `-OtherAttributes` parameter, and to create objects not explicitly supported by a cmdlet, you can use `New-ADObject`, and specify the type of object you want to create.

Of course, for any serious Windows PowerShell user, command-line parameters are only one way to specify attribute values when creating a new object with the `New-ADUser` cmdlet. Another possible method is to use an existing object as a template. When you specify the name of the object you want to use as a template on the `New-ADUser` command line, using the `-instance` parameter, the system copies all of the attribute values from the template to the new object, except for those overridden by other parameters on the command line.

Yet another method, suitable for creating multiple Active Directory objects using a single command, is to create a comma-separated value (CSV) file containing a list of the objects you want to create and their attribute values. You can then use the `Import-CSV` cmdlet to pipe the contents of the CSV file to the `New-ADObject` cmdlet, and the system will create each object listed in the file in turn.

In addition to cmdlets for creating Active Directory objects, there are also cmdlets for manipulating them, such as the following examples:

- **Set-ADObject** Modifies the properties of an Active Directory object
- **Get-ADObject** Gets or performs a search to retrieve one or more Active Directory objects
- **Move-ADObject** Moves an Active Directory object or container from one container to another or from one domain to another
- **Restore-ADObject** Restores a deleted Active Directory object
- **Rename-ADObject** Renames an Active Directory object
- **Remove-ADObject** Removes an Active Directory object

A comprehensive treatise on managing Active Directory using the capabilities provided by the Active Directory Module for Windows PowerShell could easily fill this book. The preceding are some extremely basic examples of how, with a little study and a little practice, you can learn to enhance and streamline the processes by which you perform your regular Active Directory management tasks, using the tools provided in Windows Server 2008 R2.

Active Directory Administrative Center: Better Interactive Administration

Of course, there are some administrators who are simply not comfortable working from the command line. Indeed, there are some who scarcely know it exists. However, the capabilities provided by the Active Directory Module for Windows PowerShell need not be lost on those who prefer a graphical interface. Windows Server 2008 R2 also includes a new graphical Active Directory Management tool, called Active Directory Administrative Center (ADAC).

ADAC is a shell application for, and is dependent on, the cmdlets in the Active Directory Module for Windows PowerShell. You must install the Active Directory Module and have all of its prerequisite requirements in place before you can use ADAC. The console works by taking the selections you make and the information you supply in the ADAC graphical interface and translating them into the proper command-line syntax, using the cmdlets in the Active Directory Module. The program then executes the commands, receives the results, and displays the results in a graphical fashion.

As shown in Figure 5-2, the basic structure of the ADAC interface uses a scope pane (on the left) and a details pane (on the right)—the same organizational paradigm as Windows Explorer and most Microsoft Management Console (MMC) snap-ins. The Overview page provides access to the root of your domain, as well as basic functions, such as directory search and password reset. As with most pages in ADAC, you can customize the appearance of the page, in this case by clicking the Add Content link and specifying which tiles should appear in the details pane.

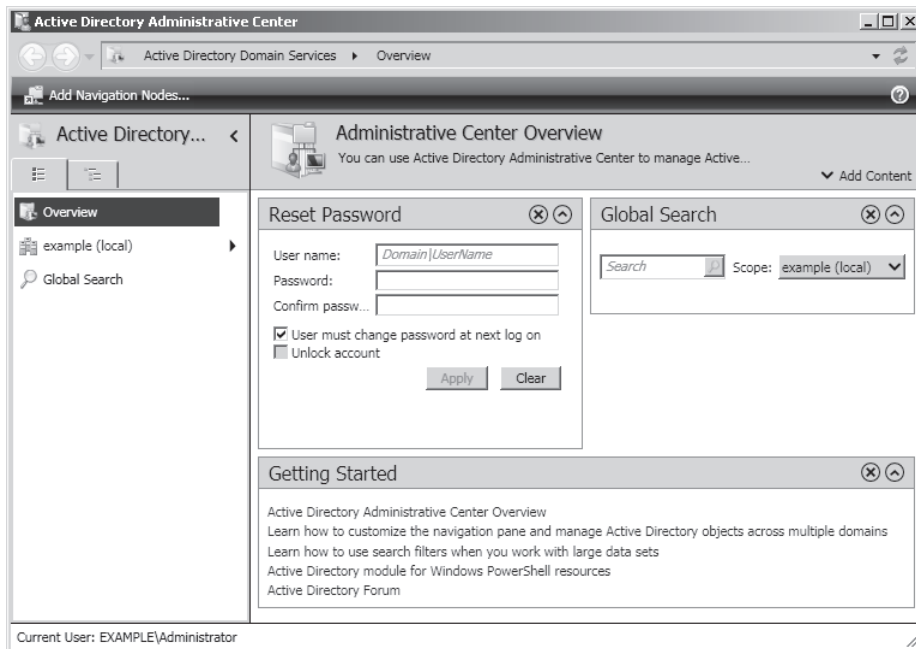


FIGURE 5-2 The Overview page in Active Directory Administrative Center.

Compared to the existing graphical management tool for AD DS—the Active Directory Users and Computers console (which remains unchanged in Windows Server 2008 R2)—Microsoft has designed ADAC with the following general improvements in mind:

- **Streamlined procedures** By completing tasks in one step that previously required two or more, ADAC makes Active Directory management simpler and more intuitive.
- **Increased information density** By displaying more information on a single page, administrators using ADAC can manage Active Directory objects without navigating through multiple tabs and dialog boxes.
- **Greater interface customization** By enabling administrators to select the tools and features they use most often, ADAC can provide a simplified, and yet more comprehensive, interface.

Creating Objects

Generally speaking, ADAC enables you to do more with a single step than Active Directory Users and Computers. For example, when creating a new user object, Active Directory Users and Computers only lets you specify the user's name, supply a password, and configure a few basic options. For anything else, you have to create the user first and then open its Properties sheet to configure it, often switching between many different tabbed pages in the process. With ADAC, the Create User page, shown in Figure 5-3, contains a great many more configuration settings—in fact, more than can fit in this figure. This enables you to supply

organizational information for the user, specify group memberships, and configure user profile settings, all while you are actually creating the user object.

FIGURE 5-3 The Create User page in Active Directory Administrative Center.

NOTE Not coincidentally, the list of configuration settings on the Create User page closely resembles the list of parameters for the New-ADUser cmdlet discussed earlier in this chapter.

In addition to creating new Active Directory objects, ADAC also enables you to move, disable, rename, and delete objects, and configure their properties.

Customizing the Interface

ADAC includes a Tree View that you can use to browse your domain, in the style of Active Directory Users and Computers, but it also has a List View option, to which you can add your own navigation nodes, as shown in Figure 5-4.

Navigation nodes are essentially shortcuts that point to specific containers anywhere in your domain or in other domains. Using the Add Navigation Nodes page, shown in Figure 5-5, you can browse your enterprise and select the containers you need to access on a regular basis. For AD DS installations that span multiple domains, or even multiple forests, administrators can manage objects in containers anywhere in the enterprise, as long as there are trusts in place between the domains or forests.

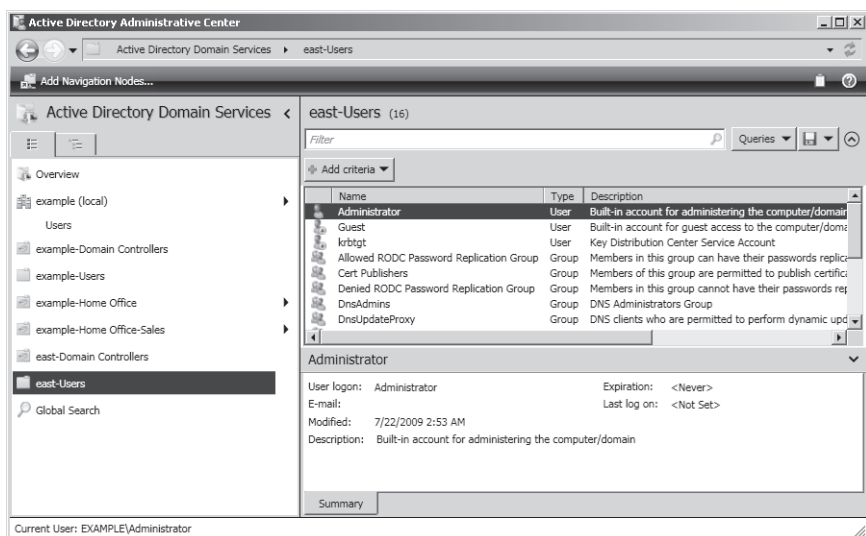


FIGURE 5-4 The Active Directory Administrative Center List View, with additional navigation nodes.

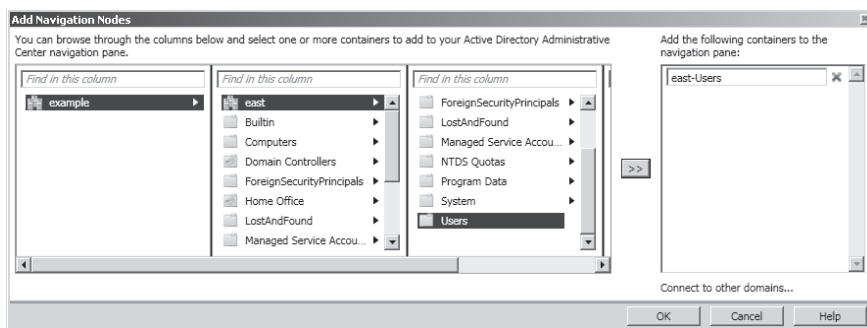


FIGURE 5-5 The Add Navigation Nodes page in Active Directory Administrative Center.

ADAC also provides a powerful Active Directory object search mechanism. You can build complex queries by specifying the exact object criteria you want to search within, limiting the scope of the search to specific navigation nodes, and using the Lightweight Directory Access Protocol (LDAP) query syntax. Suppose, for example, you are managing a large, multidomain Active Directory installation, and you have to locate the user object of the vice president who just called to complain that he is locked out of his account. You can easily create a query that searches only for users with disabled accounts (by selecting the Users With Disabled/Enabled Accounts criterion), within a specific domain (by selecting the domain name in the Scope selector), as shown in Figure 5-6. You can then save the query for later reuse when the vice president locks himself out again.

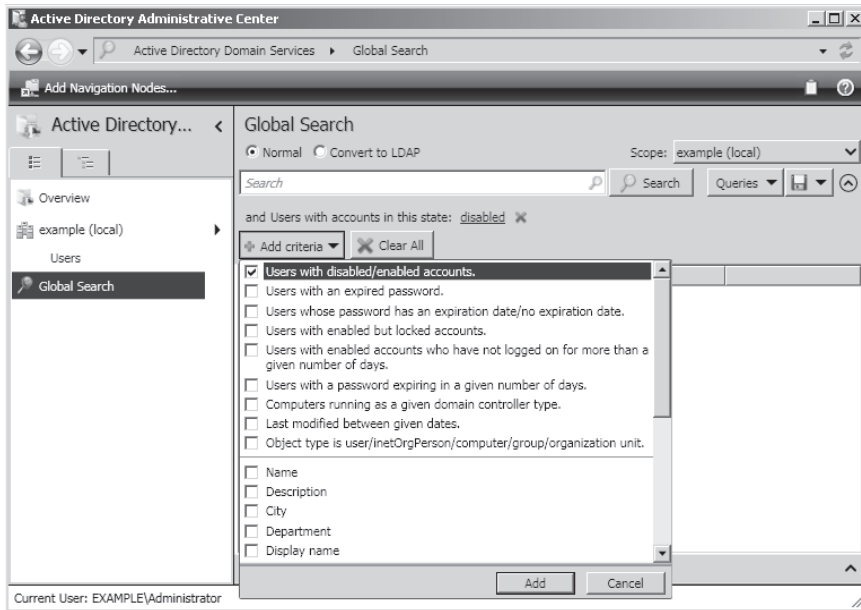


FIGURE 5-6 Building a search query in Active Directory Administrative Center.

Introducing Active Directory Web Services

ADAC might appear to be nothing more than a new management interface for Active Directory, but there is actually quite a bit that is new beneath the surface. As with the traditional management tools implemented as MMC snap-ins, such as Active Directory Users and Computers, you can use ADAC remotely to manage Active Directory resources anywhere on the network. However, unlike the MMC snap-ins, which rely on the Remote Procedure Calls (RPC) protocol for communications between the console and the domain controller, ADAC and the underlying Active Directory Module for Windows PowerShell cmdlets use a new communications infrastructure called Active Directory Web Services (ADWS). New to Windows Server 2008 R2, ADWS represents the first step in a major shift away from the RPC and LDAP models Active Directory has used for network communications up until now.

Windows Server 2008 R2 automatically installs ADWS with the AD DS and AD LDS roles, in the form of an executable called `Microsoft.ActiveDirectory.WebServices.exe`, located in the `%windir%\ADWS` folder. When you promote a server to an AD DS domain controller or create an AD LDS instance, the system configures ADWS to load automatically as a service when the computer starts. ADWS requires Microsoft .NET Framework 3.5.1 to run, which explains the new dependency in the Add Roles Wizard mentioned earlier in this chapter.

NOTE Active Directory Web Services is included in the Windows Server 2008 R2 Standard, Enterprise, and Datacenter editions, but it is not included in Windows Web Server 2008 R2 or Windows Server 2008 R2 for Itanium-Based Systems.

ADWS must be running on at least one directory service computer running Windows Server 2008 R2 for any communication to take place between the Active Directory Module for Windows PowerShell cmdlets (or ADAC) and an AD DS domain controller or an AD LDS instance. This is true not just in remote management scenarios, but for activities confined to the local system as well. If the ADWS service stops or fails to start, or you disable it, you will not be able to use Windows PowerShell or ADAC to manage the directory service, even when working at the domain controller console.

In a remote management scenario, no matter how you install the Active Directory Module for Windows PowerShell, the system will not be able to import the module successfully unless it has access to Active Directory Web Services on a computer running Windows Server 2008 R2. This means that the computer running Active Directory Module for Windows PowerShell must either be an AD DS domain controller or have an AD LDS instance itself, or it must be a member of a domain with at least one domain controller running Windows Server 2008 R2. If the computer is not a member of a domain, or it is a member of a domain without a Windows Server 2008 R2 domain controller, you cannot use the Active Directory Module cmdlets to manage Active Directory.

NOTE Although there has been no official announcement as of yet, it is rumored that Microsoft will eventually release a version of Active Directory Web Services for computers running Windows Server 2008 and possibly earlier versions as well. This would enable administrators to manage domain controllers running those operating systems using the Active Directory Module for Windows PowerShell and ADAC. Unfortunately, this will be no benefit to administrators running Windows Server 2008 Server Core because the pre-R2 version of the operating system lacks the support for .NET Framework 3.5.1 needed to run ADWS.

A single instance of the ADWS host service provides administrative access to all of the Active Directory directory service components on a computer running Windows Server 2008 R2, which can include an AD DS domain controller, a Global Catalog, and multiple AD LDS instances. Internal communication between ADWS and the Active Directory components uses LDAP.

For communication with the Active Directory Module for Windows PowerShell cmdlets, ADWS uses the Windows Communication Foundation (WCF) interface provided by .NET Framework 3.5.1 to exchange messages using standard Web service protocols, such as WS-Transfer and WS-Enumeration. These Web service protocols use SOAP, the native WCF message representation (which at one time stood for Simple Object Access Protocol but, mysteriously, is no longer an acronym), to generate Extensible Markup Language (XML) code, which the system transmits over the network using an application layer or transport layer protocol. The basic ADWS communications architecture is shown in Figure 5-7.

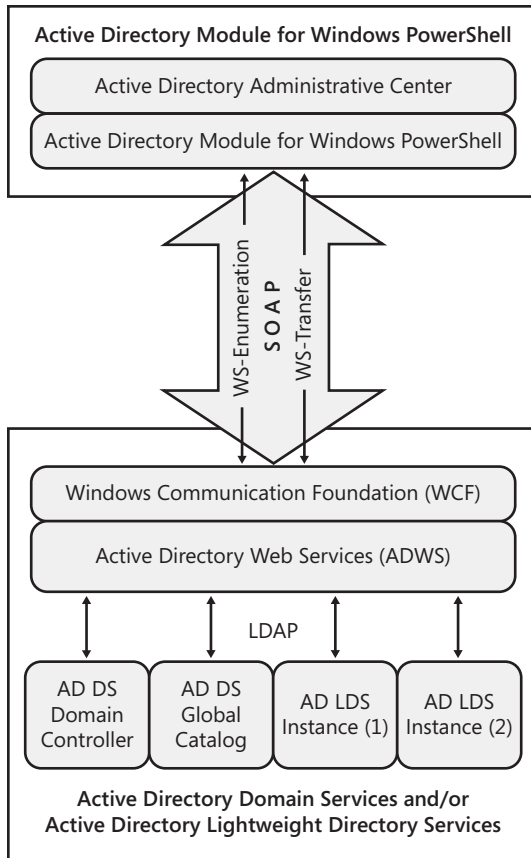


FIGURE 5-7 Active Directory Web Services communications.

For remote administration over the network, ADWS uses Transmission Control Protocol (TCP) port 9389 on the computer running AD DS or AD LDS on Windows Server 2008 R2. Any firewalls between the system running Windows Server 2008 R2 and the computer running Active Directory Module for Windows PowerShell must have this port open.

Remote Active Directory Administration with Windows PowerShell Cmdlets

As mentioned earlier in this chapter, Windows Server 2008 R2 automatically installs the Active Directory Module for Windows PowerShell when you add the AD DS or AD LDS role, and ADAC as well, with AD DS. When you promote the server to an AD DS domain controller or create an AD LDS instance, the system then installs and activates Active Directory Web Services, which is everything you need to manage Active Directory using Windows PowerShell on that computer. However, administrators often want to manage Active Directory

from another computer at a remote location, and you can do so with the Active Directory Module and ADAC, as long as you are running Windows Server 2008 R2 or Windows 7 on the remote computer.

To manage AD DS or AD LDS resources from a computer running Windows Server 2008 R2 that is not an AD DS domain controller and that does not host an AD LDS instance, you must install the Active Directory Module for Windows PowerShell and (optionally) the ADAC module, using the Add Features Wizard, accessible in Server Manager or the Initial Configuration Tasks window. If you prefer, you can also install the features using Windows PowerShell cmdlets or the Servercmd.exe command-line tool, as described in the following sections.

Installing Remote Server Administration Tools with the Add Features Wizard

The Active Directory Module for Windows PowerShell and the ADAC are part of the Remote Server Administration Tools feature, which you can add as a whole or by selecting individual modules, as shown in Figure 5-8. Both modules require you to install the .NET Framework 3.5.1 feature as well, and to install ADAC, you must also install the Active Directory Module for Windows PowerShell and AD DS Snap-Ins and Command-Line Tools features. As mentioned earlier, your server must be a member of an AD DS domain with at least one Windows Server 2008 R2 domain controller.

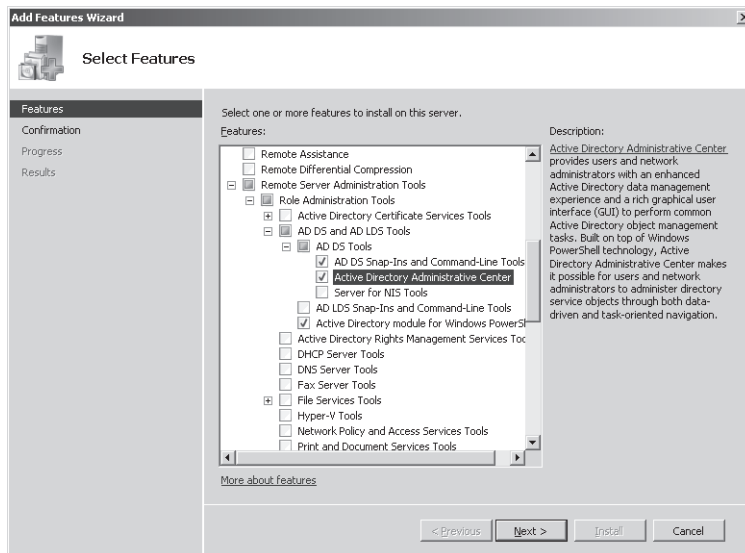


FIGURE 5-8 Installing Remote Server Administration Tools with the Add Features Wizard.

Installing Remote Server Administration Tools with Windows PowerShell

You can also install individual parts of the Remote Server Administration Tools feature from the Windows PowerShell prompt, using the capabilities provided in the ServerManager module. After opening a Windows PowerShell session with elevated privileges (by right-clicking the shortcut and selecting Run As Administrator), use the following command to import the ServerManager module:

```
Import-Module ServerManager
```

Once you have done this, you can install individual features by name using the Add-WindowsFeature cmdlet. To display a list of the Command IDs for all of the roles and features available for installation, use this command:

```
Get-WindowsFeature
```

You can then use the following command to install the Active Directory Module for Windows PowerShell and ADAC features. The cmdlet automatically installs all of the dependent elements the two features require.

```
Add-WindowsFeature RSAT-AD-PowerShell,RSAT-AD-AdminCenter
```

Installing Remote Server Administration Tools with Servercmd.exe

You can also use the same Command IDs to install the features from a standard command prompt, although Microsoft has now declared this method to be deprecated in favor of Windows PowerShell. Here too you must open your command prompt session with elevated privileges, and then execute the following two commands, individually:

```
Servercmd.exe -install RSAT-AD-PowerShell
```

```
Servercmd.exe -install RSAT-AD-AdminCenter
```

Installing Remote Server Administration Tools on Windows 7

You can manage your Active Directory resources from a Windows 7 workstation also, but first you must download and install the Remote Server Administration Tools for Windows 7 package from the Microsoft Download Center at <http://www.microsoft.com/downloads>. After you install the package, you must open the Programs Control Panel, select Turn Windows Features On Or Off, and select the appropriate check boxes under Remote Server Administration Tools, as shown in Figure 5-9.

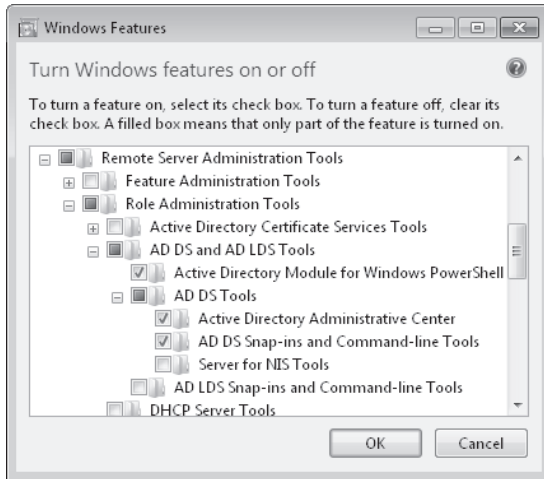


FIGURE 5-9 Turning on Remote Server Administration Tools in Windows 7.

Selecting Functional Levels in Windows Server 2008 R2

In Windows Server 2008 R2, as in all of the previous Windows Server releases since Windows 2000, functional levels are essentially a version control system for domain controllers. Because all of the domain controllers in a domain and in some cases a forest have to communicate with each other, they must all be running the same Active Directory code to implement certain new features. When a Windows Server release adds new functionality to Active Directory, it is often necessary for all participating domain controllers to be running that same release.

Raising a domain or a forest to a specific functional level prevents domain controllers not supporting the same functional level from joining the domain or the forest. This ensures that all of the domain controllers support the same set of features. For example, if you create a new domain and specify that it use the Windows Server 2008 domain functional level, then any additional domain controllers you add to the domain must be running Windows Server 2008 (or a newer version) as well. In the same way, if you set the forest functional level to Windows Server 2008, all of the domains you create in that forest will operate at the Windows Server 2008 domain functional level.

Administrators can set functional levels while promoting a server to a domain controller using the Active Directory Domain Services Installation Wizard (Dcpromo.exe), or after promoting the domain controller by using any of the following tools:

- Active Directory Domains and Trusts, a snap-in for MMC
- The Set-ADForestMode and Set-ADDomainMode cmdlets available in the Active Directory Module for Windows PowerShell

- Active Directory Administrative Center, a graphical interface for the Active Directory Module for Windows PowerShell
- Ldp.exe, Windows Server's graphical LDAP client

IMPORTANT Raising the functional level of a domain or a forest is one of the few irrevocable administrative functions in Windows Server. Once you have raised a domain functional level or forest functional level, you cannot undo that action, except in certain highly specific circumstances.

Raising Functional Levels Using Windows PowerShell

To raise the functional level of a forest or domain using Windows PowerShell, you must have the Active Directory Module for Windows PowerShell installed and imported into a Windows PowerShell session with elevated privileges. Then you can use commands such as the following:

```
Set-ADForestMode -Identity forest_name.com -ForestMode  
Windows2008R2Forest
```

```
Set-ADDomainMode -Identity domain_name.com -DomainMode  
Windows2008R2Domain
```

Using the Windows Server 2008 R2 Forest Functional Level

When you create a new Active Directory forest on a computer running Windows Server 2008 R2, the Active Directory Domain Services Installation Wizard displays a Set Forest Functional Level page, as shown in Figure 5-10, on which you select the functional level you want the forest to use.

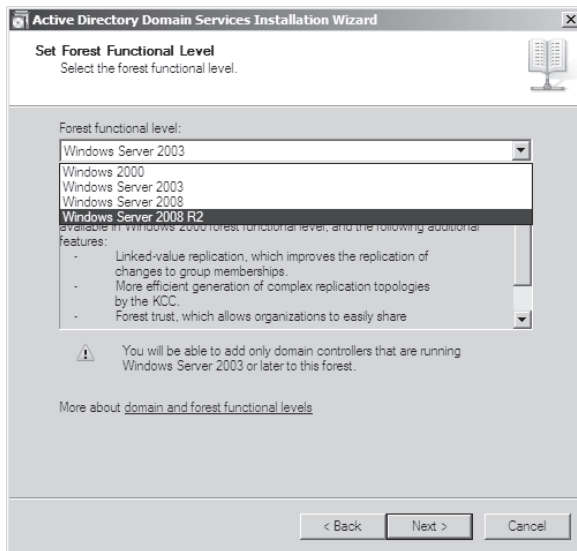


FIGURE 5-10 The Set Forest Functional Level page in the Active Directory Domain Services Installation Wizard.

When you select the Windows Server 2008 R2 forest functional level, the following modifications apply:

- All of the new domains you create in the forest will operate at the Windows Server 2008 R2 domain functional level by default.
- All of the domain controllers in the forest must be running Windows Server 2008 R2 or higher. Active Directory will not permit you to add any domain controller running an operating system prior to Windows Server 2008 R2 to any domain in the forest. Note, however, that this restriction affects only domain controllers, not member servers or workstations.
- The domain controllers in the forest implement all of the features provided by the lower forest functional levels.
- The domain controllers in the forest implement the new Active Directory Recycle Bin feature included with Windows Server 2008 R2. This feature enables administrators to restore deleted Active Directory objects while Active Directory Domain Services is running.

Using the Windows Server 2008 R2 Domain Functional Level

If you select the Windows Server 2008 R2 forest functional level while creating a new forest, you have no choice regarding the domain functional level because all of the domains in a Windows Server 2008 R2 forest must use the Windows Server 2008 R2 domain functional level. However, if you select a forest functional level of Windows Server 2008 or lower, the

Active Directory Domain Services Installation Wizard displays a Set Domain Functional Level page, like that shown in Figure 5-11. This page enables you to select any functional level for the domain equal to or higher than the forest functional level setting.

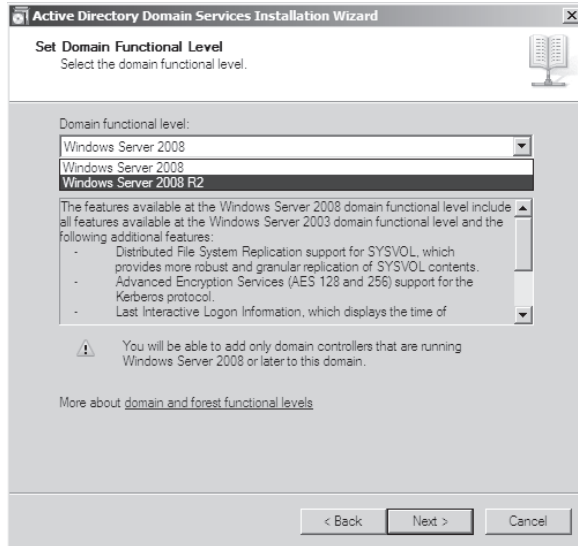


FIGURE 5-11 The Set Domain Functional Level page in the Active Directory Domain Services Installation Wizard.

Although it might seem counterintuitive, it is possible to set the domain functional level higher than the forest functional level, and this is the only scenario in which it is possible to lower a functional level after you have raised it. If your forest is set to the Windows Server 2008 forest functional level, you can raise your domain to the Windows Server 2008 R2 domain functional level, and then lower it back down to the Windows Server 2008 domain functional level, if necessary.

NOTE You can only roll back the domain functional level from Windows Server 2008 R2 to Windows Server 2008, and only when the forest functional level is Windows Server 2008 or below. You cannot roll back the domain functional level to Windows Server 2003, no matter what the value of the forest functional level.

When you elevate the domain functional level to Windows Server 2008 R2, the domain controllers for the domain implement all of the features provided by the lower domain functional levels. The only new feature in the Windows Server 2008 R2 domain functional level is Authentication Mechanism Assurance, a feature that can apply to logons performed within an AD DS forest or to interforest claims generated by AD FS. With Authentication Mechanism Assurance, a domain controller can insert information about a logged-on user's authentication method in the token issued to the user by the Kerberos authentication

protocol. The information takes the form of a global group membership. This enables the system to grant users access to certain protected resources only when they meet specific authentication requirements, such as when they use a smart card or when the smart card they use has a certificate with 2,048-bit encryption.

Active Directory Recycle Bin: Recovering Deleted Objects

Accidental deletions are a common occurrence in computing. Users often delete files or folders they shouldn't, and in the same way, administrators inadvertently delete Active Directory objects. At one time, when a user deleted an important file, it was necessary for an administrator to restore it from a system backup. Microsoft then introduced the Recycle Bin feature to the Windows operating systems, which enables users to reclaim their deleted files themselves. For years, administrators have requested a similar feature for Active Directory.

In Windows Server 2008 and earlier versions, it is possible to restore a deleted Active Directory object from a backup, but the process is daunting. After performing the restoration from the backup medium, you have to mark the object as authoritative, to ensure that it replicates to all of your domain controllers, and you have to do this in Directory Services Restore Mode, which means the domain controller must be offline. With Windows Server 2008 R2, however, we finally have a Recycle Bin for Active Directory that enables administrators to restore deleted objects with all of their attributes and permissions intact.

NOTE Another form of Active Directory object recovery, called *tombstone reanimation*, has also been available since the Windows Server 2003 release, and this recovery process does not require any server downtime. However, objects in their tombstone state lose some of their attribute values, so the recovered objects are lacking some of their properties.

Understanding Windows Server 2008 R2 Object Recovery

On an installation using the Windows Server 2008 forest functional level or lower, when you delete an Active Directory object, it experiences a change of state, becoming a *tombstone object* and losing many of its attributes in the process. With the Windows Server 2008 forest functional level and the Active Directory Recycle Bin enabled, deleting an object causes its state to change to *logically deleted*, with all of its attributes left intact. This is a new state in Windows Server 2008 R2, during which it is possible to restore the object without the loss of any properties or permissions. The system moves objects in this state to a Deleted Objects container and mangles their distinguished names so that they are not accessible by the usual means.

A logically deleted object remains in that state for the duration of its *deleted object lifetime*, which by default is 180 days. At the end of the deleted object lifetime, the object's state changes to *recycled object*. This is also a new state in Windows Server 2008 R2, and although objects in this state lose most of their attributes like tombstone objects, they are not recoverable at this point, using either the Recycle Bin or the authoritative restore process in Directory Services Restore Mode. After the object's *recycled object lifetime* expires, which is another 180 days by default, the garbage collection process physically deletes the object from the Active Directory database.

TIP Administrators can change the lifetime values from their defaults by modifying the `msDS-deletedObjectLifetime` attribute for the deleted object lifetime, and the `tombstone-Lifetime` attribute for the recycled object lifetime. To modify these attributes, you can use the `Set-ADObject` cmdlet in the Active Directory Module for Windows PowerShell or the `Ldp.exe` LDAP client.

Enabling the Active Directory Recycle Bin

The Active Directory Recycle Bin is available in Windows Server 2008 R2, but it is disabled by default. Before you can use the Recycle Bin, you must perform the following procedures for AD DS:

IMPORTANT Enabling the Active Directory Recycle Bin is an irrevocable act. Once you enable it, you cannot disable it again.

- Prepare the Active Directory schema If you are upgrading your forest from Windows Server 2008 or earlier, upgrade the directory schema by using an account with Schema Admins privileges to execute the following commands from a command prompt:
 - `adprep /forestprep` on the server that holds the schema master role
 - `adprep /domainprep /gpprep` on the server that holds the infrastructure operations master role
 - `adprep /rodcprep` if you have any read-only domain controllers on your network

NOTE If you have created your forest on a clean Windows Server 2008 R2 installation, you do not have to upgrade the schema with `Adprep.exe`.

- Upgrade all of your domain controllers to the Windows Server 2008 R2 operating system, if necessary.

- Raise the forest functional level to Windows Server 2008 R2.

If you are running AD LDS, perform the following procedures:

- Upgrade all of your servers running instances of AD LDS to Windows Server 2008 R2, if necessary.
- Update the directory schema by executing the following command, replacing the variables with the appropriate values on each server:

```
ldifde.exe -i -f MS-ADAM-Upgrade-2.ldf -s server_name:port -b username
domain_name password -j . -s adamschema.cat
```

- Raise the functional level of the AD LDS configuration set to Windows Server 2008 R2.

With all of the preparation finished, you are ready to actually enable the Recycle Bin, using one of the following commands from an Active Directory Module for Windows PowerShell prompt with elevated privileges:

- For AD DS, use the following command, replacing the variables with appropriate values for your installation:

```
Enable-ADOptionalFeature 'Recycle Bin Feature' -Scope
ForestOrConfigurationSet -Target 'forest_name.com'
```

- For AD LDS, use the following command, modifying the -Target parameter with the appropriate distinguished name for your installation:

```
Enable-ADOptionalFeature 'recycle bin feature' -Scope
ForestOrConfigurationSet -Server localhost:50000 -Target
'CN=Configuration,CN={372A5A3F-6ABE-4AFD-82DE-4A84D2A10E81}'
```

Using the Active Directory Recycle Bin

Once you have enabled the Active Directory Recycle Bin, you can restore any objects you delete, using the cmdlets in the Active Directory Module for Windows PowerShell.

NOTE Active Directory Recycle Bin makes it possible to restore any objects you delete after it is enabled. You cannot use Recycle Bin to restore objects you deleted before you enabled Recycle Bin. These are already tombstone objects, and most of their attributes are irrevocably lost.

After opening a session with elevated privileges, restoring deleted objects requires two cmdlets: `Get-ADObject`, to locate the desired object in the Deleted Objects folder, and `Restore-ADObject`, to perform the actual restoration. You can run each cmdlet separately, noting the `ObjectGUID` value displayed by `Get-ADObject` so you can include it on the

Restore-ADObject command line, or you can combine the two by piping the Get-ADObject results to the Restore-ADObject cmdlet in the following manner:

```
Get-ADObject -Filter 'string' -IncludeDeletedObjects | Restore-ADObject
```

The *string* variable must contain search criteria that display the object or objects (and only the object or objects) you want to restore. For example, the following command will restore an object with the display name "Mark Lee."

```
Get-ADObject -Filter 'displayName -eq "Mark Lee"' -IncludeDeletedObjects | Restore-ADObject
```

To display the entire contents of the Deleted Objects folder, use the following command, replacing the *forest_name* and *top_level_domain* variables with values appropriate to your installation:

```
Get-ADObject -SearchBase "CN=Deleted Objects,DC=forest_name,DC=top_level_domain"  
-ldapFilter "(objectClass=*)" -includeDeletedObjects
```

TIP If you do not know the exact display name of the object you want to restore, you can use any viable value for the `-Filter` string. You might want to run the `Get-ADObject` cmdlet by itself first (without the pipeline to `Restore-ADObject`), while you experiment with string values. For example, the command `Get-ADObject -Filter 'displayName -like "M*"' -IncludeDeletedObjects` will return a list of all deleted objects that have display names starting with "M." You can then refine your filter until you create a string that returns only the object or objects you want to restore. For more information on the syntax of the `Get-ADObject -Filter` parameter, type the command `get-help about_ActiveDirectory_Filter` in an Active Directory Module for Windows PowerShell session.

When restoring multiple objects, and especially organizational units (OUs) that contain other objects, the order in which you restore the objects can be critical and the filter strings can be more complex. With the Active Directory Recycle Bin, you can only restore objects to a live parent. This means, for example, that if you accidentally delete an OU object, you must restore the OU itself before you can restore any of the objects in that OU. If you delete an OU that contains other OUs, you must start by restoring the parent OU (that is, the highest deleted OU in the hierarchy) before you can restore the subordinate ones.

TIP When restoring a hierarchy of objects, a series of exploratory `Get-ADObject` commands might be necessary to ascertain the correct order for the restorations. In these cases, you might want to use commands that include the `-Properties lastKnownParent` parameter to determine parental relationships between the deleted objects.

Offline Domain Join: Securing and Facilitating Deployment

Another long-term complaint of Active Directory administrators has been the need to have a workstation connected to an AD DS domain controller to join it to a domain. Many IT organizations prefer to install and configure their servers and workstations at a central location, and then deploy them to their final destinations. In many cases, this means that the domain the computer will eventually join is not available at the time of the installation. The result is that IT personnel have to wait to join the computer to the domain until the system is on site, which is often an impractical solution.

The offline domain join capability in Windows Server 2008 R2 enables administrators to gather the information needed to join a computer running Windows Server 2008 R2 or Windows 7 to a domain and save it to the computer without it requiring access to the domain controllers. When the computer starts for the first time in its final location, it automatically joins to the domain using the saved information, with no interaction and no reboot necessary.

Djoin.exe is a command prompt tool that you run on one computer to gather the metadata needed to join another computer to a domain, create its computer account in AD DS, and save the metadata to an encrypted file. Once this is complete, you copy the file to the computer you want to join to the domain and run Djoin.exe there. The first computer, called the provisioning computer, must be running Windows Server 2008 R2 or Windows 7, and it must have access to a domain controller. By default, the domain controller must be running Windows Server 2008 R2. An example of a basic provisioning command appears as follows:

```
djoin /provision /domain example.local /machine Wkstn1 /savefile c:\wkstn1_join.txt
```

In this example, the /domain parameter specifies the name of the domain you want the target computer to join, the /machine parameter the name you want to assign to the target computer, and the /savefile parameter the name of the metadata file you want to create. Optional parameters enable you to specify the name of an OU where you want to create the computer object, and the name of a specific domain controller to use.

NOTE For more information on the Djoin.exe syntax, type **djoin /?** at a Windows Server 2008 R2 or Windows 7 command prompt.

To deploy the metadata on the target computer, which must also be running Windows Server 2008 R2 or Windows 7, you copy the file Djoin.exe created to that system and run the program again, this time with the /requestodj parameter, as in the following example:

```
djoin /requestodj /loadfile c:\wkstn1_join.txt /windowspath %windir% /localos
```

At this point, the target computer can still be located in the setup facility. The system does not have to have access to its eventual domain, or even be connected to a network. Once you have provisioned the computer, you can move it to its final location. The next time you restart the system, it will be joined to the domain you specified and ready to use.

This example provisioned a computer's local Windows installation, but you can also use Djoin.exe to provision offline virtual machines, or even computers on which you haven't yet installed the operating system. To do the latter, you insert a reference to the metadata file that Djoin.exe created into an Unattend.xml file, for use during an automated installation.

Service Accounts

Applications and services require accounts to access network resources, just as users do. Administrators can configure an application to run using the Local Service, Network Service, or Local System account. These accounts are simple to manage, but they do have drawbacks. First, they are local accounts, which means administrators cannot manage them at the domain level. Second, these system accounts are typically shared by multiple applications, which can be a security issue. It is possible to configure an application to use a standard domain account. This enables you to isolate the account security for a particular application, but it also requires you to manage the account passwords manually. If you change the account password on a regular basis, you must reconfigure the application that uses it, so that it supplies the correct password when logging on to the domain.

The managed service account is a new feature in Windows Server 2008 R2 that takes the form of a new Active Directory object class. Because managed service accounts are based on computer objects, they are not subject to Group Policy–based password and account policies as are domain users. Managed service policies also do not allow interactive logons, so they are an inherently more secure solution for applications and services. Most importantly, managed service accounts eliminate the need for manual credential management. When you change the password of a managed service account, the system automatically updates all of the applications and services that use it.

To create a managed service account, you must use the `New-ADServiceAccount` cmdlet in the Active Directory Module for Windows PowerShell. You can also use the `Get-ADServiceAccount` cmdlet to locate existing managed service accounts. To use a managed service account for a particular application or service, you must run the `Install-ADServiceAccount` cmdlet on the computer hosting the application.

Best Practices Analyzer

As they have already done with Microsoft Exchange Server 2007 products, Microsoft has integrated its BPA technology into the Active Directory Domain Services and Active Directory Certificate Services roles in Windows Server 2008 R2. Administrators can initiate BPA scans using the graphical interface in Server Manager, or from a Windows PowerShell prompt.

NOTE The BPA is included in the Windows Server 2008 R2 Standard, Enterprise, and Data-center editions, but it is not included in Windows Web Server 2008 R2 or Windows Server 2008 R2 for Itanium-Based Systems, nor is it included in any Server Core edition.

The BPA has a collection of predefined rules for each role it supports—rules specifying the recommended architectural and configurational parameters for the role. For example, one AD DS rule recommends that each domain have at least two domain controllers. When you run a BPA scan, the system compares the recommendations to the actual role configuration and points out any discrepancies. The scan returns a status indicator for each rule that indicates whether the system is compliant or noncompliant. There is also a warning status for rules that are compliant at the time of the scan, but that configuration settings might render noncompliant under other operational conditions.

To trigger an AD DS BPA scan in Server Manager, you select the Active Directory Domain Services node and, in the Best Practices Analyzer section, click Scan This Role. After a delay as the analyzer performs the scan, the results appear, as shown in Figure 5-12. You can exclude results that you don't need, such as rules that you do not think you need to follow.

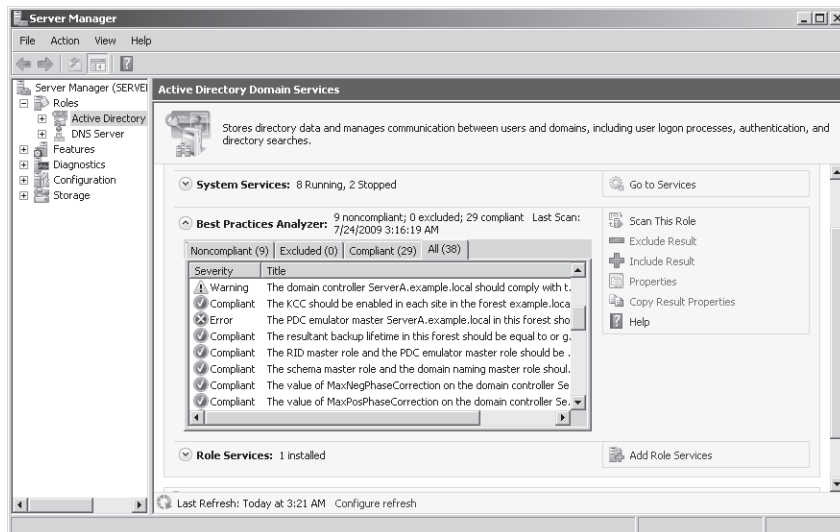


FIGURE 5-12 The Best Practices Analyzer for Active Directory Domain Services in Server Manager.

The BPA works by running a Windows PowerShell script that collects information about the system and stores it in an XML file. The analyzer then compares its preconfigured rules to the information in the XML file and reports the results. You can also run BPA from the Windows PowerShell command prompt directly, but first you must import the ServerManager and BestPractices modules. The cmdlets that perform the BPA operations are as follows:

- **Get-BPAModel** Displays the roles installed on the server that are supported by the BPA
- **Get-BPAResult** Displays the results of the most recently executed BPA scan for a specified role
- **Invoke-BPAModel** Initiates a new BPA scan for a specified role
- **Set-BPAResult** Enables you to include or exclude specific results in a BPA scan report

The File Services Role

- Using the File Classification Infrastructure **91**
- Using BranchCache **101**
- Introducing Distributed File System Improvements **108**

The overall take on file services in Windows Server 2008 R2 is to help administrators in an enterprise environment manage the increasingly large amounts of data that are their responsibility. Although storage space is cheaper and more plentiful than ever before, the increased emphasis on audio and video file types, whether business related or not, has led to a storage consumption rate that in many instances more than equals its growth.

There is only one new role service in the File Services role, but there are innovative new features introduced into some of the existing role services. In an enterprise with multiple sites, increased storage capacity typically leads to increased consumption of bandwidth between sites, and these new features can help administrators manage this bandwidth consumption and improve file access times in the process.

Using the File Classification Infrastructure

An enterprise network can easily have millions of files stored on its servers, and administrators are responsible for all of them. However, different types of files have different management requirements. Enterprise networks typically have a variety of storage technologies to accommodate their different needs. For example, drive arrays using Redundant Array of Independent Disks (RAID) for fault tolerance are excellent solutions for business-critical files, but they are also more expensive to purchase, set up, and maintain. Storing noncritical files on a medium such as this would be a waste.

At the other end of the spectrum, an offline or near-line storage medium, such as magnetic tape or optical disks, can provide inexpensive storage for files that are not needed on a regular basis, or that have been archived or retired. The big problem for the administrator with a variety of storage options is determining which files should go on which medium, and then making sure that they get there.

There are often other storage management factors to consider as well, such as the following:

- **Encryption** Files containing confidential information might require encrypted storage and backup media.
- **Permissions** Business-critical files often need special permission assignments to prevent unauthorized persons from accessing or modifying them.
- **Backups** Important files that change frequently might require additional backups several times per day.

However, determining which files require a certain treatment and seeing that they receive it can be a major administrative problem.

Traditional methods for classifying files include storing them in designated folders, applying special file naming conventions, and, in the case of backups, the long-standing use of the archive bit to indicate files that have changed. None of these methods are particularly efficient for complex scenarios on a large scale, however, because of the manual maintenance they require or their limited flexibility. Who is going to be responsible for making sure that files are named properly, or moved to the appropriate folders? It would not be practical for IT personnel to monitor the file management practices of every user on the network. Also, if you designate one folder for files containing sensitive data and another for files that are modified often, what do you do with a file that is both sensitive and frequently updated?

Introducing the FCI Components

The *File Classification Infrastructure (FCI)* introduced in Windows Server 2008 R2 is a system that enables administrators to define their own file classifications, independent of directory structures and file names, and configure applications to perform specific actions based on those classifications.

FCI consists of four components, as follows:

- **Classification Properties** Attributes created by administrators that identify certain characteristics about files, such as their business value or level of sensitivity
- **Classification Rules** Mechanisms that automatically apply classification properties to certain files based on specific criteria such as file contents
- **File Management Tasks** Scheduled operations that perform specified actions on files with certain classification properties
- **Storage Reports Management** Engine that can generate reports that, among other things, document the distribution of classification properties on file server volume

For example, an administrator might create a classification property that indicates whether a file contains personal or confidential information. To apply that property automatically, the administrator can create a classification rule that searches files for the words “personal” or “confidential.” A backup application can then use the property to differentiate between

sensitive files, which it saves to an encrypted backup medium, and nonsensitive files, which it saves to an unprotected medium that is faster and cheaper.

The Classification Management mechanisms, which you use to create properties and rules, are integrated into the File Server Resource Manager (FSRM) console as new nodes, as shown in Figure 6-1. Also new is the File Management Tools node, which you use to execute specific actions based on the file classifications you have created. The Storage Report Management node now includes the ability to generate reports based on FCI properties, as well as other, traditional criteria.

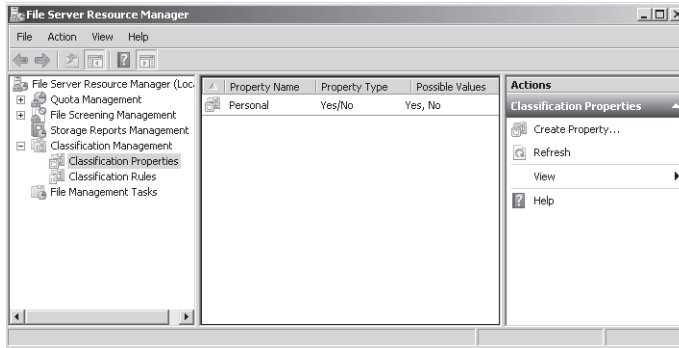


FIGURE 6-1 The File Server Resource Manager console.

FCI is designed to be more of a toolkit for storage administrators than an end-to-end solution. FCI provides various types of classification properties, but it is up to the individual administrator to apply them to the particular needs of an enterprise. File Management Tools provides a basic file expiration function and the ability to execute custom commands against particular file classifications. However, FCI is also designed with an extensible infrastructure so that third-party developers can integrate property-based file selection into their existing products.

Creating FCI Classification Properties

The first step in implementing FCI is to create the classification properties that you will apply to files with certain characteristics. Classification properties are simple attributes, consisting only of a name, a property type, and sometimes a list of values. Property types indicate the nature of the classification you want to apply to your files; they do not have to contain the classification criteria themselves. For example, if you want to create a classification property that indicates whether a file contains sensitive information, you create a Yes/No property, which is a simple Boolean switch that says either “Yes, this file contains sensitive information” or “No, this file does not contain sensitive information.” The classification property itself contains only the Yes or No, not the reason for the selection.

Administrators, therefore, must decide what classifications they need to apply to their files and which property types best express them. FCI supports seven classification property types, as listed in Table 6-1.

TABLE 6-1 FCI Classification Properties

PROPERTY	DESCRIPTION	AGGREGATION
Yes/No	A simple Boolean value, indicating that a file either possesses a certain characteristic or it does not	Yes values take precedence over No values
Date-time	A simple date and time stamp	No aggregation supported
Number	A simple numerical value	No aggregation supported
Multiple Choice List	A list of possible values, any or all of which can be assigned to a file	Different list values from multiple property assignments are combined in a single file
Ordered List	A list of numbered values, only one of which a file can possess	The highest value assigned takes precedence over the lower values
String	A simple character string	No aggregation supported
Multistring	An unordered list of character strings, any or all of which can be assigned to a file	Different list values from multiple property assignments are combined in a single file

The use of different property types enables you to classify files in many different ways. For example, you can use a Yes/No property to indicate whether a file contains sensitive information or not, but for applications requiring more granularity, it might be better to use an Ordered List property to assign a security classification of High, Medium, or Low.

Aggregation refers to the behavior of a classification property type when a rule or other process attempts to assign the same property to a file, but with a different value. Some properties (Yes/No and Ordered List) aggregate multiple assignments by having one value take precedence over the others, whereas other properties (Multiple Choice List and Multistring) can contain multiple values, derived from aggregated assignments.

MORE INFO The Date-time, Number, and String properties do not support aggregation at all. An attempt to assign a second property value to an already-classified file results in an error. You can configure a rule to reevaluate files with these properties, but the rule will simply assign a new value that overwrites the old one, without considering the existing value of the property.

In the current example, the Ordered List property is suitable for a security indicator because it can contain only one value. When there is a value conflict, such as if one rule assigns a file High Security and another rule assigns it Low Security, the High Security value takes precedence, as shown on the left side of Figure 6-2, enabling the property to err on the side of caution and use the greatest possible security measures. However, if you are seeking to categorize files based on subject, the Multiple Choice List property would probably be preferable, because it enables you to assign multiple properties to a single file, as shown on the right side of the figure.

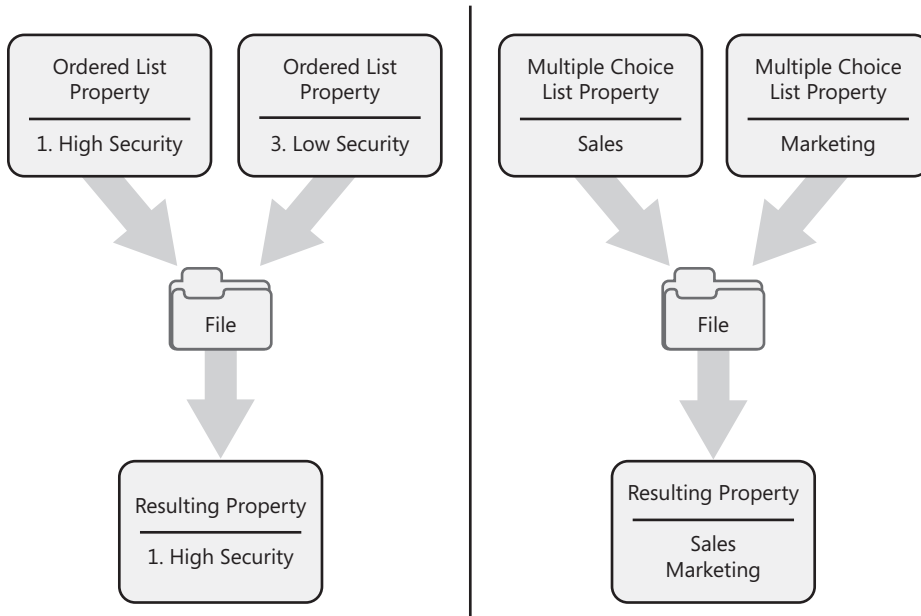


FIGURE 6-2 FCI property value aggregations.

To create properties, you select the Classification Properties node in File Server Resource Manager and click Create Property to open the Create Classification Property Definition dialog box, as shown in Figure 6-3. After specifying a name for the property, and optionally a description, you select a Property Type, and the controls change depending on the type you have chosen. The types that do not support a selection of possible values (Date-time, Number, and String) require no additional configuration. The other types enable you to add the possible values that your classification rules can assign to files, based on criteria you select.

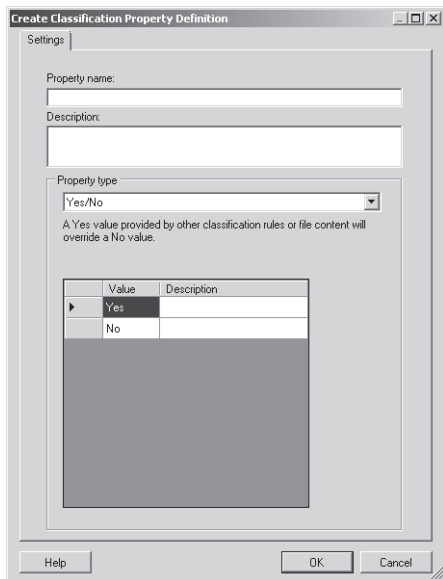


FIGURE 6-3 The Create Classification Property Definition dialog box.

Creating FCI Classification Rules

Once you have created your classification properties, you can assign them to your files by creating classification rules. To create a rule, you select the Classification Rules node in File Server Resource Manager and click Create A New Rule to open the Classification Rule Definitions dialog box. On the Rule Settings tab, shown in Figure 6-4, you supply a name for the rule, and optionally a description, and then click Add to define the scope; that is, specify the volumes or folders containing the files to which you want to apply properties.

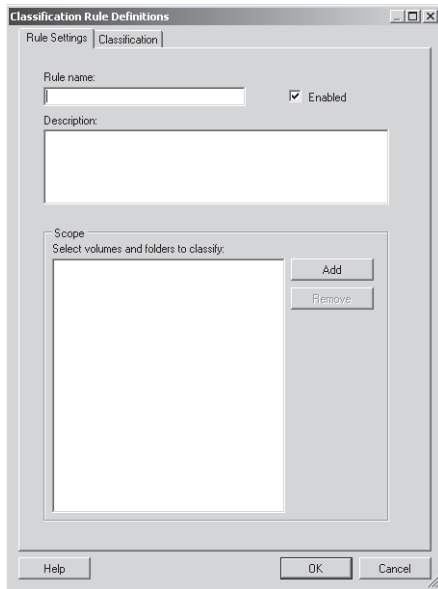


FIGURE 6-4 The Classification Rule Definitions dialog box.

On the Classification tab, shown in Figure 6-5, you select one of the following classification mechanisms:

- **Folder Classifier** Assigns properties to files based on the folders in which they are stored
- **Content Classifier** Assigns properties to files based on a search for specific terms or expressions

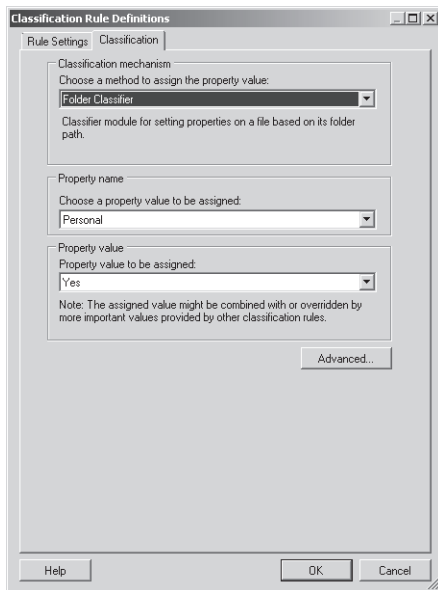


FIGURE 6-5 The Classification tab of the Classification Rule Definitions dialog box.

NOTE These classification mechanisms take the form of plug-in modules, of which Windows Server 2008 R2 includes only two relatively rudimentary examples. Microsoft has designed this part of the FCI to be extensible, so that administrators and third-party developers can use the FCI application programming interface (API) to produce their own classification plug-ins, as well as scripts and applications that set properties on files.

In the Property Name and Property Value fields, you specify which of your classification properties you want to assign to the files the rule selects, and what value the rule should insert into the property.

Clicking Advanced displays the Additional Rule Parameters dialog box, in which you find the following tabs:

- **Evaluation Type** Enables you to specify how the rule should behave when it encounters a file that already has a value defined for the specified property. You can elect to overwrite the existing property value or aggregate the values (for properties that support aggregation).
- **Additional Classification Parameters** Enables you to specify regular expressions or text strings that you want the rule to search for in each file, applying the property only when a file matches the parameters.

NOTE You cannot assign classification properties to files that are encrypted. If you encrypt files after they have classification properties assigned, they retain those properties and applications can read them, but you cannot modify the properties or assign new ones while the files are in their encrypted state.

Once you have created your classification rules, you must execute them to apply properties to your files. You can click Run Classification With All Rules Now to execute your rules immediately, or you can click Configure Classification Schedule to run them at a later time or at regular intervals.

TIP Administrators new to FCI have a tendency to create large numbers of properties and rules, simply because they can. Be aware that processing rules, and especially those that search for complex regular expressions, can take a lot of time and consume a significant amount of server memory. Microsoft recommends only applying classifications that your current applications can utilize.

Performing File Management Tasks

Once you have classified your files, you can use File Server Resource Manager to create file management tasks, which can manipulate the files based on their classification properties. Here again, the capabilities provided with Windows Server 2008 R2 are relatively rudimentary, but as with the classification mechanisms, administrators and third-party developers can integrate property-based file processing into their applications.

Selecting the File Management Tasks node in FSRM and clicking Create File Management Task displays a dialog box of the same name. Here, as in the Classification Rule Definitions dialog box, you supply a name, a description, and a scope for the task.

On the Action tab, you can select one of the following action types:

- **File Expiration** Enables you to move files matching specified property values to another location
- **Custom** Enables you to execute a program, command, or script on files matching specified property values, using the interface shown in Figure 6-6.

On the Condition tab, you specify the property values that files must possess for the file management task to process them, using the Property Condition dialog box, as shown in Figure 6-7. The Schedule tab enables you to configure the task to execute at specified intervals, and the Notification and Report tabs specify the types of information administrators receive about the task processing.

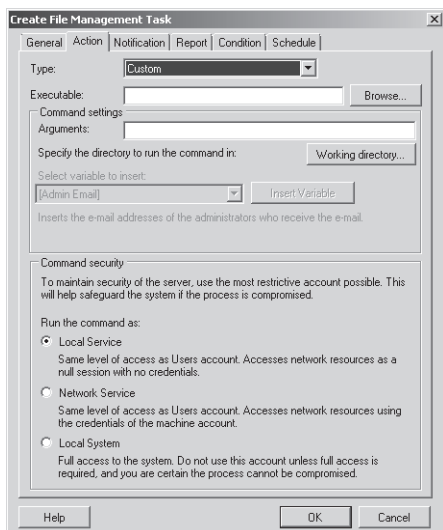


FIGURE 6-6 The Custom action type interface in the Create File Management Task dialog box.

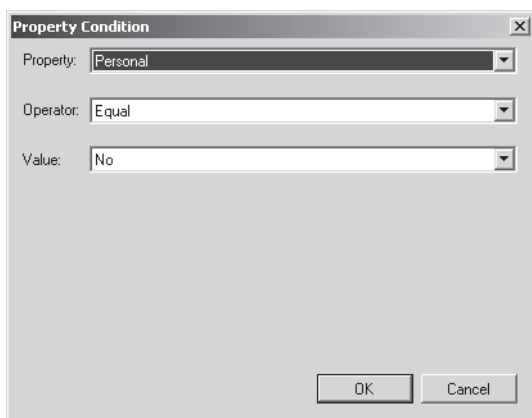


FIGURE 6-7 The Property Condition dialog box.

Although the File Expiration action type enables administrators to migrate files based on property values, it is the Custom action that provides true power for the savvy administrator. Using the Executable and Arguments fields, administrators can run a command, program, or script on the files having the specified properties. Some of the possible scenarios for customized tasks are as follows:

- Modify the permissions for the selected files using Lcacs.exe program
- Compact selected database files at regular intervals using Compact.exe
- Perform customized backups of the specified files using the Windows Server Backup PowerShell cmdlets

- Create a script that moves the selected files to another location and uses Mklink.exe to create symbolic links to them

With the scripting capabilities of Windows PowerShell and the many command prompt applications provided with Windows Server 2008 R2, FCI and its Custom action type form an extremely flexible file management tool.

Using BranchCache

Branch office technologies were a major priority for the Windows Server 2008 R2 and Windows 7 development teams, and BranchCache is one of the results of that concentration. On an enterprise network, a branch office can consist of anything from a handful of workstations with a virtual private network (VPN) connection to a fully equipped network with its own servers and IT staff. In most cases, however, branch offices nearly always require some network communication with the home office, and possibly with other branches as well. The wide area network (WAN) connections between remote sites are by nature slower and more expensive than local area network (LAN) connections, and the primary functions of BranchCache are to reduce the amount of WAN bandwidth consumed by branch office file sharing traffic and improve access times for branch office users accessing files on servers at remote locations.

As the name implies, BranchCache is file caching software. Caching is a technique by which a system copies frequently used data to an alternative storage medium, so that it can satisfy future requests for the same data more quickly or less expensively. BranchCache works by caching files from remote servers on the local drive of a branch office computer so that other computers in the branch office can access those same files locally, instead of having to send repeated requests to the remote server.

BranchCache has two operational modes, as follows:

- **Distributed Cache Mode** Up to 50 branch office computers cache files requested from remote servers on their local drives, and then make those cached files available to other computers on the local network, on a peer-to-peer basis.
- **Hosted Cache Mode** Branch office computers cache files requested from remote servers on a branch office server; the server makes those cached files available to other computers on the branch office network.

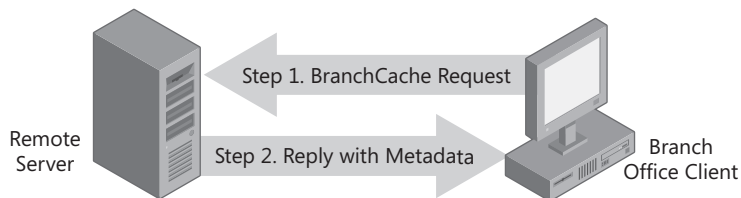
The primary difference between these two modes is that Hosted Cache Mode requires the branch office to have a server running Windows Server 2008 R2, whereas Distributed Cache Mode requires only Windows 7 workstations. The advantage of Hosted Cache Mode is that the server, and therefore the cache, is always available to all of the workstations in the branch office. Workstations in Distributed Cache Mode can only share cached data with computers on the local network, and if a workstation is hibernating or turned off, its cache is obviously unavailable.

BranchCache is a read-only cache, meaning that when client computers read files from a remote server, they store copies in the cache for later use by other computers, but when they save files back to the remote server, BranchCache is not involved at all. This is because caching writes is a much more complicated operation than caching reads, due to possible existence of conflicts between multiple versions of the same file.

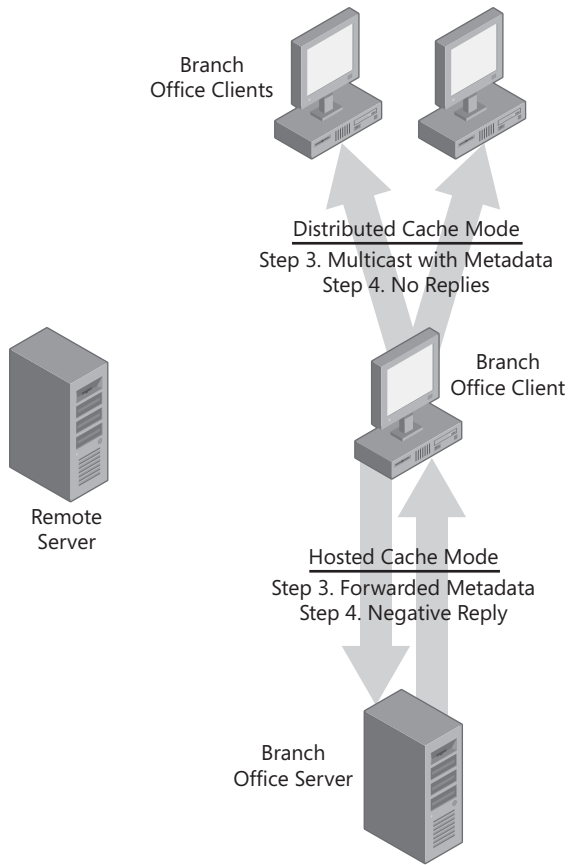
Understanding BranchCache Communications

BranchCache is a client/server application; administrators must configure both the computers at the branch office and the remote servers to use it. The BranchCache communication between the clients and the remote server proceeds as follows:

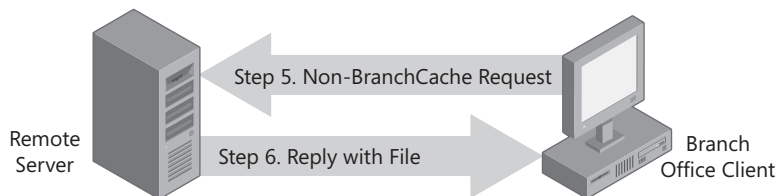
1. A BranchCache-enabled Windows 7 client at a branch office transmits a request for a file to a remote server using the Hypertext Transfer Protocol (HTTP), the Background Intelligent Transfer Service (BITS), or the Server Message Blocks (SMB) protocol. The only difference from a standard request is that the client includes an identifier in the message, indicating that it supports BranchCache.
2. When the BranchCache-enabled remote server receives the request and recognizes that the client also supports BranchCache, it replies, not with the requested file, but with content metadata in the form of a hash describing the requested file, as shown in the following graphic. The metadata is substantially smaller than the requested file itself, so the amount of WAN bandwidth utilized so far is relatively small.



3. On receiving the metadata from the remote server, the client computer forwards it to the caching computer(s) on the local network, to see if the requested file is already present in another computer's cache. On a Distributed Cache Mode installation, the client sends this message as a multicast transmission to the other BranchCache clients on the network, using the BranchCache discovery protocol. On a Hosted Cache Mode installation, the client sends the message to the local server that hosts the cache, using the BranchCache retrieval protocol.
4. In Distributed Cache Mode, the client fails to receive a reply from another client on the network. In Hosted Cache Mode, the client receives a reply from the local server indicating that the requested data is not in the cache, as shown in the following graphic.

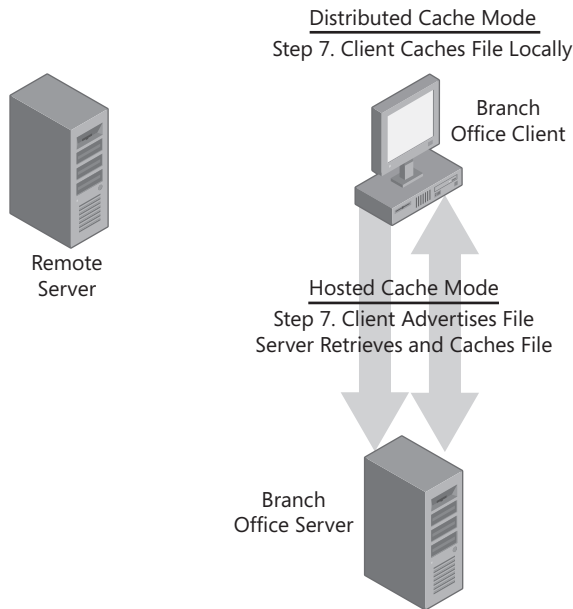


5. The client retransmits its original file request to the remote server. This time, however, the client omits the BranchCache identifier from the request message.
6. The remote server, on receiving a standard (non-BranchCache) request, replies by transmitting the requested file, as shown in the following graphic.



7. The client receives the requested file and, on a Distributed Cache Mode installation, stores the file in its local cache. On a Hosted Cache Mode installation, the client sends a message to its local caching server using the BranchCache hosted cache protocol, advertising the availability of its newly downloaded data. The local server then

connects to the client, downloads the data using the BranchCache retrieval protocol, and stores it in the cache, as shown in the following graphic.



When another client subsequently requests the same data from the remote server, the communication process is exactly the same up until step 4. In this case, the client receives a reply from another computer (either client or server, depending on the mode) indicating that the requested data is present in its cache. The client then uses the BranchCache retrieval protocol to download the data from the caching computer. For this and subsequent requests for that particular file, the only WAN traffic required is the exchange of request messages and content metadata, both of which are much smaller than the actual data file.

Configuring a BranchCache Server

Windows Server 2008 R2 and Windows 7 both support BranchCache as clients, but only R2 can function as a BranchCache server. BranchCache is not installed by default on Windows Server 2008 R2; you must install one or both of the BranchCache modules supplied with the operating system, and then create Group Policy settings to configure them.

As mentioned earlier, BranchCache can transmit HTTP, BITS, or SMB data. HTTP and BITS are protocols that Web servers and application servers typically use, whereas SMB is the default Windows file sharing protocol. To enable BranchCache for all three protocols, you must install both of the following two modules using Server Manager. If you only intend to cache SMB data, you do not have to install the BranchCache feature, and to cache only HTTP and BITS data, you do not need the BranchCache for Network Files role service.

- **BranchCache** A Windows Server 2008 R2 feature that provides caching support for the HTTP and BITS protocols, as well as the BranchCache client and Hosted Cache Mode functionality
- **BranchCache for Network Files** A role service in the File Services role that provides caching support for the SMB protocol, as well as command prompt administration support

Enabling Hash Publication

To enable a computer to function as a BranchCache server, you must configure a Group Policy setting called Hash Publication for BranchCache, which you can find in the Computer Configuration > Policies > Administrative Templates > Network > Lanman Server node of a Group Policy object (GPO) or the Local Computer Policy. This setting enables the file server to transmit content metadata to qualified BranchCache clients instead of the actual files they request. When you enable Hash Publication for BranchCache, as shown in Figure 6-8, you can elect to allow hash publication for all file shares on the computer, or only for the file shares on which you explicitly enable BranchCache support.

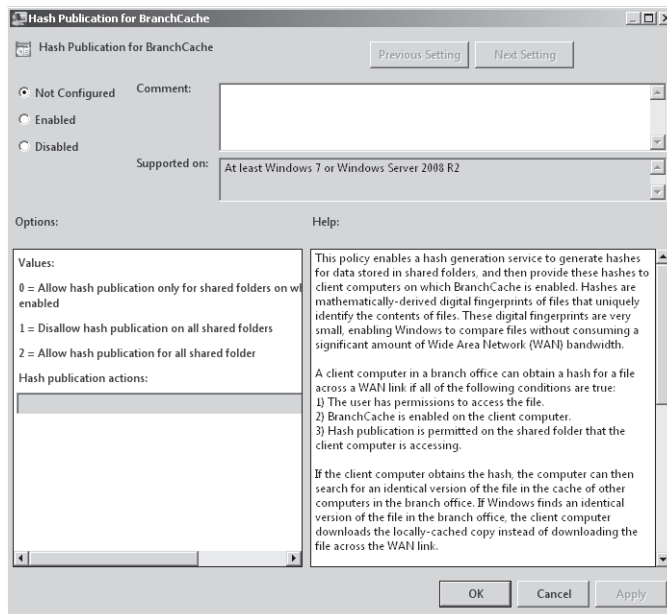


FIGURE 6-8 The Hash Publication for BranchCache dialog box.

Configuring File Shares to Support BranchCache

When you select the Allow Hash Publication Only For Shared Folders On Which BranchCache Is Enabled option in the Hash Publication for BranchCache Group Policy setting, as described in the previous section, you must configure each share that you want to provide content

metadata to BranchCache clients. To do this, you use the Share and Storage Management console to open a share's Properties dialog box, then click Advanced. On the Caching tab, select the Enable BranchCache check box, as shown in Figure 6-9.

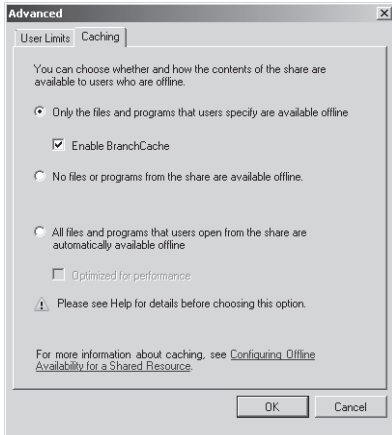


FIGURE 6-9 The Advanced dialog box for a share in the Share and Storage Management console.

Configuring BranchCache Clients

To configure a computer running Windows Server 2008 R2 to function as a BranchCache client, you must install the BranchCache feature. Computers running Windows 7 have the BranchCache client installed by default. However, for both operating systems, you must configure some of the following Group Policy settings found in the Computer Configuration > Policies > Administrative Templates > Network > BranchCache node of a GPO or the Local Computer Policy before the client is operational:

- **Turn On BranchCache** This setting enables BranchCache on the client computer. Enabling this setting along with either Set BranchCache Distributed Cache Mode or Set BranchCache Hosted Cache Mode configures the client to use one of those operational modes. Enabling this setting without either one of the mode settings configures the client to cache server data on its local drive only, without accessing caches on other computers.
- **Set BranchCache Distributed Cache Mode** When enabled along with the Turn On BranchCache setting, this setting configures the client to function in Distributed Cache Mode.
- **Set BranchCache Hosted Cache Mode** When enabled along with the Turn On BranchCache setting, this setting configures the client to function in Hosted Cache Mode. In the Enter The Location Of The Hosted Cache field, you must specify the fully qualified domain name (FQDN) of the computer running Windows Server 2008 R2 that will function as the Hosted Cache server on the branch office network.

- **Configure BranchCache For Network Files** When enabled, this setting controls the round-trip network latency value that BranchCache uses to differentiate local from remote servers. The default setting is 80 ms. When you decrease the value, the client caches more files; increasing the value causes it to cache fewer files.
- **Set Percentage Of Disk Space Used For Client Computer Cache** When enabled, this setting specifies the maximum amount of total disk space that the computer should devote to the BranchCache cache. The default value is 5 percent.

IMPORTANT BranchCache clients operating in Distributed Cache Mode communicate with the other clients on the branch office network using the HTTP and WS-Discovery protocols. To facilitate this communication, administrators must configure any firewalls running on the clients to admit incoming traffic on the ports these two protocols use, which are Transmission Control Protocol (TCP) port 80 and User Datagram Protocol (UDP) port 3702, respectively. Clients operating in Hosted Cache Mode only require the HTTP port (TCP port 80) to be open.

Configuring a Hosted Cache Mode Server

To configure a computer running Windows Server 2008 R2 to function as a Hosted Cache server, you must install the BranchCache feature and enable the Turn On BranchCache and Set BranchCache Hosted Cache Mode Group Policy settings, as described in the previous sections. You must then provide the server with a certificate issued by a certification authority (CA) that the clients on the branch office network trust. This can be an internal CA running on the network or a commercial CA run by a third party.

Once you have obtained the required certificate, you import it on the Hosted Cache server using the Certificates snap-in for Microsoft Management Console (MMC), noting the certificate's Thumbprint value as you do so. Then, to link the certificate to BranchCache, you execute the following command from an elevated command prompt, replacing the *thumbprint* variable with the value you obtained from the certificate:

```
NETSH HTTP ADD SSLCERT IPORT=0.0.0.0:443 CERTHASH=thumbprint APPID={d673f5ee-a714-454d-8de2-492e4c1bd8f8}
```

TIP In addition to linking the certificate, you can also use the Netsh.exe program at the command prompt to manually configure the BranchCache client. Note, however, that client configuration values you set using Group Policy take precedence over those you set with Netsh.exe from the command prompt.

Introducing Distributed File System Improvements

The Distributed File System (DFS) is an important element of the File Services role, which has received some enhancements in the Windows Server 2008 R2 release, including the following:

- **Performance enhancements** The DFS Namespaces implementation in Windows Server 2008 R2 provides reduced startup times and improved performance for namespaces with 5,000 linked folders or more.
- **Access-based enumeration support** *Access-based enumeration* is a function that hides file system elements from users that do not have the permissions needed to access them. In Windows Server 2008 R2, you can now enable access-based enumeration on a DFS namespace, using the DFS Management console or the Dfsutil.exe command prompt utility. However, to do so, the namespace must be hosted on a server running Windows Server 2008 R2 or Windows Server 2008. If you enable access-based enumeration on a DFS namespace and on the target shares that the namespace links to (using the Share and Storage Management console), the shared folders are completely hidden from unauthorized users.
- **Read-only replicated folders** Using the Windows Server 2008 R2 version of the DFS Management console, you can configure a member of a DFS Replication group to be read-only, preventing users from modifying the files in the replicated folder. Prior to the R2 release, you could only do this by manually changing the permissions on the replicated folder. Note, however, that read-only folders impose an additional performance burden on the servers hosting them, because DFS Replication must intercept every Create and Open function call to determine if the requested destination is in a read-only folder.
- **Additional performance counters** Windows Server 2008 R2 includes three new DFS-related performance counters that you can use in the Performance Monitor snap-in to gather information about DFS processes. The DFS Namespace Service API Queue counter displays the number of currently queued DFS Namespace requests. The DFS Namespace Service API Requests counter monitors the frequency of specific DFS namespace request types. The DFS Namespace Service Referrals counter displays information about DFS namespace referral requests.
- **Failover cluster support in DFS Replication** Administrators can now add a failover cluster as a member of a replication group, as long as DFS Replication, the DFS Management console, and the failover cluster are all running on Windows Server 2008 R2.

IIS 7.5: Improving the Web Application Platform

- Installing IIS 7.5 **109**
- Using New IIS Services **113**
- Hosting Applications with IIS 7.5 **115**
- Managing IIS 7.5 **118**
- Accessing IIS Resources on the Internet **128**

In Windows Server 2008, Microsoft introduced Internet Information Services (IIS) 7.0, a major architectural update to its Web and application server platform. Since then, as anticipated, the IIS development team has been working on a variety of enhancements and extensions that build on that new architecture. Now, in Windows Server 2008 R2, Microsoft introduces IIS 7.5. Although based on the same basic structure as IIS 7.0, this new version includes numerous new features and refinements. This chapter lists the new features in IIS 7.5 and explains how they enhance the capabilities of the Web and application server platform.

Installing IIS 7.5

The Web Server (IIS) role in Windows Server 2008 R2 is only slightly different in appearance from that in Windows Server 2008. When you select the role in the Add Roles Wizard, the Add Features Required For Web Server (IIS) dialog box does not appear and prompt you to install the Windows Process Activation Service (WPAS) feature, as it did in Windows Server 2008. That dependency is still there, however. Even when you don't explicitly install WPAS, IIS 7.5 starts the service as needed.

IIS 7.5 also adds three new role services, as follows:

- **WebDAV Publishing** Enables users to publish content to IIS Web sites interactively and securely. For more information, see the section "Using IIS WebDAV," later in this chapter.

- **FTP Server** Enables users to transfer files to and from an IIS server and perform basic file management tasks. For more information, see the section “Using FTP Server,” later in this chapter.
- **IIS Hostable Web Core** Enables developers to integrate IIS request handling functionality into their own applications.

WebDAV Publishing and FTP Server were both add-on products for IIS 7.0 that administrators had to download and install separately. Now, in Windows Server 2008 R2, they are both fully integrated into the Web Server (IIS) role, and you can install them as part of IIS 7.5.

NOTE WebDAV Publishing and FTP Server remain downloadable add-ons for the IIS 7.0 platform on Windows Server 2008, but Microsoft has released updated versions of the downloads that provide the same capabilities as the IIS 7.5 versions.

Using Microsoft Web Platform Installer

Although Windows Server 2008 R2 administrators can still install IIS and create Web sites in the traditional manner, using the Server Manager and Internet Information Services (IIS) Manager consoles, Microsoft now provides another way. The Microsoft Web Platform is an integrated set of servers and tools that enable you to deploy complete Web solutions, including applications and ancillary servers, with a single procedure. The Microsoft Web Platform Installer is a tool that enables you to select, download, install, and configure the features you want to deploy on your Web server.

MORE INFO The Web Deployment Tool is available as a free download from the Microsoft Web site at <http://www.microsoft.com/web>.

The Web Platform Installer file you download is a stub, a tiny file that enables you to select the modules you want to install and then to download them, using the interface shown in Figure 7-1. Unlike the Web Server (IIS) role in Windows Server 2008 R2, the Web Platform Installer enables you to download other servers and applications that are produced by Microsoft and third parties. The installer provides a selection of collaboration, e-commerce, portal, and blog applications, and enforces the dependencies between the various elements. If, for example, you select an application that requires a database, the installer will download and install SQL Server Express 2008, Microsoft’s free SQL database product.

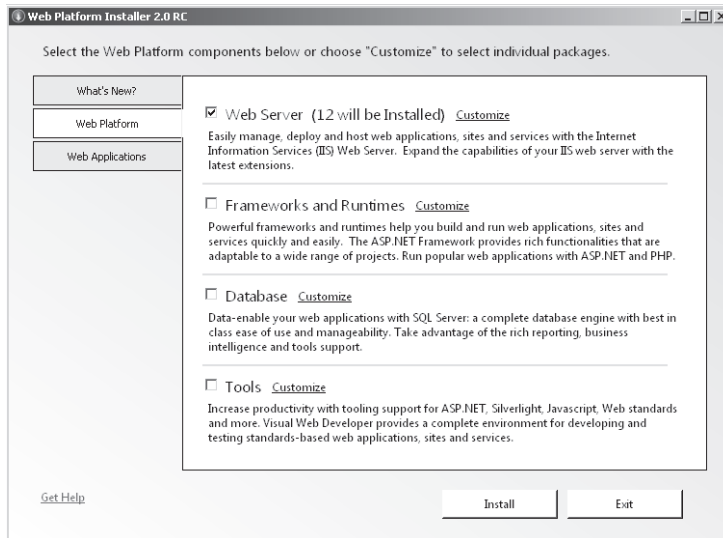


FIGURE 7-1 The Microsoft Web Platform Installer interface.

During the installation process, Web Platform Installer prompts you for information needed by your selected applications, such as what subdirectory to install them into, what passwords to use, and so on. When the process is complete, you have a fully functional Web site, complete with IIS and applications and ready to use.

Using the IIS Web Deployment Tool

The Web Deployment Tool (formerly called MS Deploy) is an IIS extension that enables administrators to package entire Web sites, Web servers, and applications for deployment on other computers, or just for backup purposes. Packages include all of a site's content, including configuration settings, permissions, databases, and certificates.

MORE INFO The Web Deployment Tool is available as a free download from the Microsoft Web site at <http://www.iis.net/extensions/WebDeploymentTool>.

When you run the Web Deployment Tool offline, it adds a Manage Packages section to the Actions pane of the Internet Information Services (IIS) Manager console, as shown in Figure 7-2.

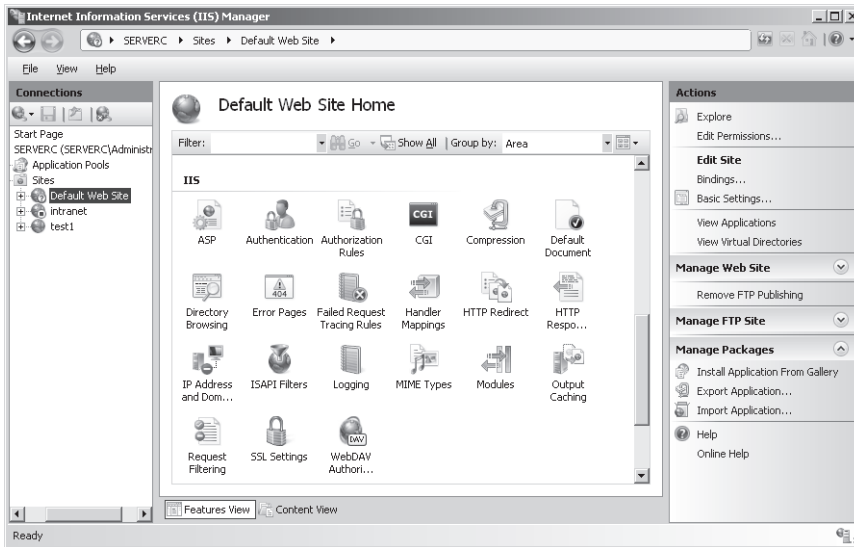


FIGURE 7-2 The Internet Information Services (IIS) Manager console, with the Web Deployment Tool installed.

Selecting a server, site, or application and clicking **Export Application** launches a wizard in which you can select the elements that you want to export, as shown in Figure 7-3. The wizard then creates a package in the form of a Zip file, which contains the original content plus configuration settings in Extensible Markup Language (XML) format.

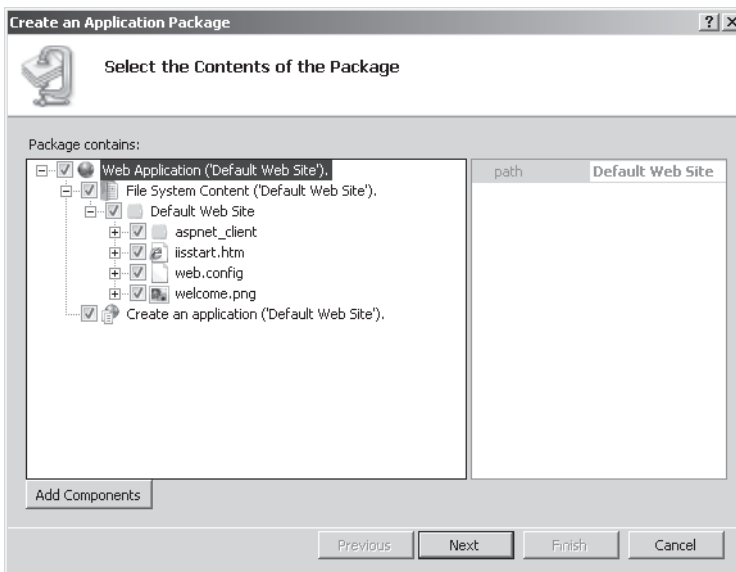


FIGURE 7-3 The Create an Application Package Wizard, provided by the Web Deployment Tool.

The package file now contains a complete copy of the server, site, or application you selected. You can save the package file to function as a backup or an archive of the site's current configuration, or copy it to another IIS server running the Web Deployment Tool and import it. The tool also includes a Remote Agent Service, which administrators can use to synchronize Web servers in real time over a network connection. This enables you to replicate sites and servers on a regular basis so that you can create Web farms for load balancing and fault tolerance purposes.

Using New IIS Services

A number of Web services that were previously available as separate downloads are now integrated into IIS in Windows Server 2008 R2, as described in the following sections.

Using IIS WebDAV

Web-based Distributed Authoring and Versioning (WebDAV) is an IIS extension, now implemented as a role service called WebDAV Publishing, which expands the capabilities of the Hypertext Transfer Protocol (HTTP) by making it possible for administrators and users to publish documents on Web sites simply by copying them to a mapped network drive. After installing the role service, you create an authoring rule that specifies what content you want to be able to publish and which users can publish it, using the interface shown in Figure 7-4. Then, using a feature called the WebDAV redirector on the client computer, you map a drive to your Web site. Copying files to that drive automatically publishes them on the Web site.

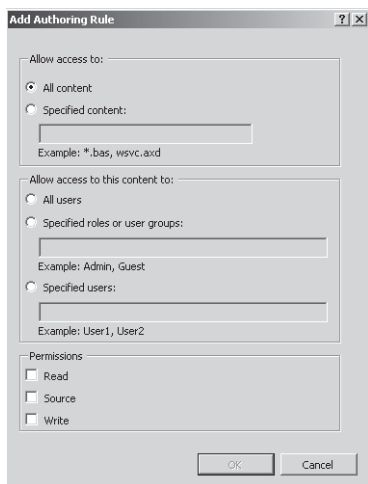


FIGURE 7-4 The Add Authoring Rule dialog box in the Windows Server 2008 R2 Internet Information Services (IIS) Manager console.

The WebDAV implementation in Windows Server 2008 R2 is fully integrated into the IIS 7.5 architecture, and supports the following features:

- **Standard compliance** The WebDAV implementation in IIS 7.5 is fully compliant with the Request for Comment (RFC) 4918 standard published by the Internet Engineering Task Force (IETF).
- **Site-level support** Unlike earlier versions, you can now enable WebDAV publishing at the site level, instead of on the entire server.
- **Support for HTTP over SSL** This enables clients to publish documents securely by encrypting transmissions using the Secure Sockets Layer (SSL) protocol.
- **Supports for locks** The WebDAV in IIS 7.5 supports both shared and exclusive locks to prevent lost updates due to overwrites.
- **Per-URL authoring rules** This enables administrators to specify WebDAV security settings for individual Uniform Resource Locators (URLs). This provides the ability to create different sets of security parameters for standard HTTP requests and WebDAV authoring.

NOTE Windows Server 2008 will always require you to obtain the WebDAV Publishing feature compatible with IIS 7.0 as a download. However, Microsoft is releasing an updated version of the service, to synchronize its feature set with the version included with Windows Server 2008 R2.

Using FTP Server

File Transfer Protocol (FTP) is one of the early protocols in the Transmission Control Protocol/Internet Protocol (TCP/IP) suite. It was created at a time when security was not as great a concern as it is now, and as a result, it has no built-in data protection of any kind. Clients transmit passwords in clear text, and transfer files to and from servers in unencrypted form. Windows Server 2008 R2, however, has an FTP server implementation that is enhanced with better security measures and other new features.

The FTP Publishing Service role service included in the Windows Server 2008 release is a holdover from Windows Server 2003. It requires you to install the old IIS 6.0 version of the management console because it is not compatible with the new IIS 7.0 architecture. Soon afterward, however, Microsoft released, as a free download, a new FTP Publishing Service that was compatible with IIS 7.0. Administrators could create and manage FTP sites using the current Internet Information Services (IIS) Manager console, and the service also included new features, such as the following:

- **FTP over Secure Sockets Layer (SSL)** Enables the FTP server to establish secure connections using password protection and SSL data encryption

- **Combined FTP and Web hosting** Enables a single IIS site to support both HTTP and FTP connections
- **Virtual host naming** Enables a single IIS server to host multiple FTP sites using a single IP address and port number, distinguishing between the sites by using host names, just as it can with Web sites
- **Improved logging and error handling** IIS log files include additional fields for FTP connections, and IIS can generate detailed error messages for clients on the local network

Now, in Windows Server 2008 R2, Microsoft has fully incorporated that FTP Publishing Service into IIS 7.5, as shown in Figure 7-5, so there is no need for a special download and no need to install an outdated management console. They have also included an additional role service, FTP Extensibility, which enables developers to use their own managed code to create customized authentication, authorization, logging, and home directory providers.

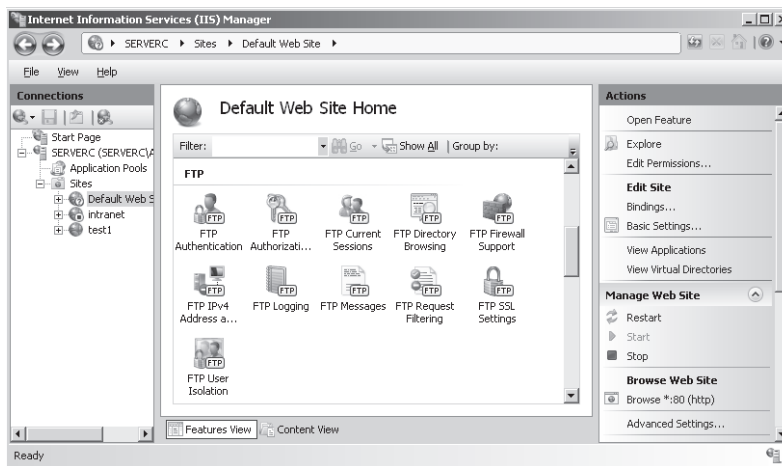


FIGURE 7-5 Managing FTP Server using the Windows Server 2008 R2 Internet Information Services (IIS) Manager.

NOTE Windows Server 2008 will always require you to obtain the FTP Publishing Service compatible with IIS 7.0 as a download. However, Microsoft is releasing an updated version of the service to synchronize its feature set with the version included with Windows Server 2008 R2.

Hosting Applications with IIS 7.5

The IIS 7.5 implementation in Windows Server 2008 R2 includes some major enhancements in its application hosting capabilities, as described in the following sections.

Running ASP.NET Applications

One of the most significant improvements in IIS 7.5 is that it now supports ASP.NET applications on computers running the Server Core installation of Windows Server 2008 R2. Server Core is a stripped-down version of the Windows Server operating system that eliminates many roles and features and most of the graphical interface. One of the features not available in Windows Server 2008 Server Core is Microsoft .NET Framework, and IIS requires this feature to support ASP.NET. Because ASP.NET is one of the most commonly used development environments for Web applications today, this was a major shortcoming. However, Windows Server 2008 R2 provides support for .NET Framework 2.0, 3.0, 3.5.1, and 4.0 in Server Core; IIS 7.5 can therefore host ASP.NET applications.

MORE INFO The .NET Framework support in Server Core also provides support for remote IIS server administration using Windows PowerShell. For more information on using Windows PowerShell with IIS, see the section “Automating IIS Administration with Windows PowerShell,” later in this chapter.

The ASP.NET implementation in IIS 7.5 also now supports different Common Language Runtime (CLR) versions, enabling administrators to switch versions without modifying the underlying IIS infrastructure. Microsoft has also incorporated this capability into Windows Server 2008 Service Pack 2.

You can specify different CLR settings for individual application pools by creating custom ASPNET.config files. To use these files, you add code specifying their locations to the pool's applicationHost.config file, as in the following example:

```
<applicationPools>
  <add name="MyApplicationPool" CLRConfigFile="c:\InetPub\CLRConfigFile.txt" />
</applicationPools>
```

IIS 7.5 also includes a new application *auto-start* feature in its ASP.NET 4.0 implementation. This feature enables an administrator to configure an application pool to start up automatically, while temporarily not processing HTTP requests. This allows applications requiring extensive initialization to finish loading the data they need or to complete other processes before they begin accepting HTTP requests. To use this feature, you must add code like the following to the pool's applicationHost.config file:

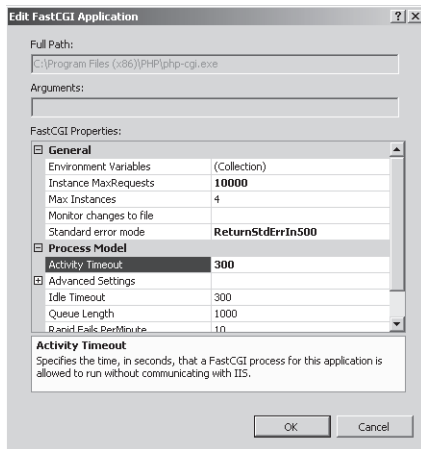
```
<applicationPools>
  <add name="MyApplicationPool" startMode="AlwaysRunning" />
</applicationPools>
```

FastCGI Support in IIS 7.5

FastCGI is a language-independent extension to the Common Gateway Interface (CGI) that enables Web servers to execute applications more quickly and efficiently. Unlike CGI, which creates a separate process for each incoming request, FastCGI uses a single process to handle multiple requests. IIS uses FastCGI to support the popular PHP scripting language, which makes it one of the more important features of the product.

IIS 7.5 includes a number of enhancements to its FastCGI support, including the following:

- **FastCGI configuration in IIS Manager** The graphical FastCGI administration interface, previously available only in Administration Pack for IIS 7.0, is now fully integrated into the Internet Information Services (IIS) Manager console, as shown in the following graphic.



- **Real-time tuning** In the Edit FastCGI Application dialog box, the Max Instances property specifies the maximum number of FastCGI processes that IIS can launch for each application pool. This equates to the maximum number of FastCGI requests that IIS can process simultaneously for that application. The default value is 4, but in IIS 7.5, if you change the value to 0, the FastCGI module automatically shifts the number of requests up and down, based on the current system load and the number of queued requests.
- **Configuration file monitoring** In the Edit FastCGI Application dialog box, the Monitor Changes To File property enables you to specify the path to a configuration file, such as Php.ini, for each application. When IIS 7.5 detects a change to the specified file, it recycles the FastCGI processes for that application.
- **New error-handling options** IIS 7.5 now provides a choice of four FastCGI error handling options, which you configure in the Edit FastCGI Application dialog box using the Standard Error Mode property. These options enable you to specify what error

information IIS logs and how much of it gets returned to users. You can also configure the property to terminate the FastCGI process when an error occurs.

- **Failed request tracing** In IIS 7.5, the FastCGI module can send the information in a process' STDERR stream to the Failed Request Tracing (FREB) logs maintained by IIS for debugging purposes (as long as Failed Request Tracing is enabled).

Using Managed Service Accounts

IIS 7.5 can use the managed service accounts—now supported by Active Directory Domain Services in Windows Server 2008 R2—as service identities, thus eliminating problems caused by expired application pool passwords.

MORE INFO For more information on managed service accounts, see “Service Accounts” in Chapter 5, “Active Directory: Improving and Automating Identity and Access.”

The Application Pool Identity is a concept first introduced in IIS 7.0 which IIS uses to set permissions for an application pool's configuration file. You can also use it for anonymous authentication in place of the IUSR account. In IIS 7.5, the Application Pool Identity is a managed service account, and IIS now uses it to run the W3wp.exe worker process in place of the Network Service account introduced in Windows Server 2003.

Managing IIS 7.5

The biggest improvement in IIS 7.5 is in the area of management. Windows Server 2008 R2 includes a number of IIS configuration tools that were previously available only as separate downloads, and Microsoft has enhanced many of the existing tools.

Automating IIS Administration with Windows PowerShell

As in many other areas of the Windows Server 2008 R2 operating system, Microsoft is emphasizing Windows PowerShell as an important tool for managing IIS 7.5. The IIS PowerShell snap-in provides dozens of new cmdlets and enables administrators to manage IIS properties in several different ways.

Selecting Windows PowerShell Modules from the Administrative Tools program group loads the system modules included with Windows Server 2008 R2, including the WebAdministration module that provides the IIS functionality. You can also import the module manually from any Windows PowerShell prompt by using the following command:

```
Import-Module WebAdministration
```

TIP To manage IIS, you should open the Windows PowerShell window using elevated privileges by selecting Run As Administrator from the Windows PowerShell Modules shortcut menu. You might also have to modify the system's execution policy with the Set-ExecutionPolicy RemoteSigned command before you can import the WebAdministration module.

Once you have access to the IIS Windows PowerShell snap-in, you can display all of the cmdlets it contains by using the following command:

```
Get-Command -pssnapin WebAdministration
```

The snap-in uses three different types of cmdlets, as follows:

- PowerShell provider cmdlets
- Low-level configuration cmdlets
- Task-oriented cmdlets

These cmdlet types correspond to three different methods of managing IIS from the Windows PowerShell prompt, as described in the following sections.

Using the IIS PowerShell Provider

The IIS PowerShell provider creates a hierarchical IIS namespace that administrators can navigate just like a file system directory structure. When you type **iis:** and press Enter at a Windows PowerShell prompt (with the WebAdministration module imported), the prompt changes to PS IIS:> and typing the **dir** command displays not the file system, but the top level of the IIS namespace, as follows:

```
Name
----
AppPools
Sites
SslBindings
```

After changing to the Sites directory with the `cd Sites` command, the `dir` command displays a list of the IIS sites on the server, as follows:

Name	ID	State	Physical Path	Bindings
----	--	-----	-----	-----
Default Web Site	1	Started	%SystemDrive%\inetpub\wwwroot	http *:80: ftp *:21:

The `Get-Item` cmdlet enables you to display selected sites in the same format. By piping the results of the `Get-Item` cmdlet to the `Select-Object` cmdlet, you can display all of the properties of a selected site, as shown in Figure 7-6.

```

Administrator: Windows PowerShell Modules
PS C:\Users\Administrator> IIS:
PS IIS:\> dir

Name
----
AppPools
Sites
SslBindings

PS IIS:\> cd sites
PS IIS:\sites> Get-Item '.\Default Web Site' | Select-Object *

PSPath                : WebAdministration:\SERVERC\Sites\Default Web Site
PSParentPath           : WebAdministration:\SERVERC\Sites
PSChildName             : Default Web Site
PSDrive                : IIS
PSProvider              : WebAdministration
PSIsContainer           : True
name                   : Default Web Site
id                     : 1
serverAutoStart        : True
bindings               : Microsoft.IIS.PowerShell.Framework.ConfigurationElement
limits                 : Microsoft.IIS.PowerShell.Framework.ConfigurationElement
logFile                : Microsoft.IIS.PowerShell.Framework.ConfigurationElement
traceFailedRequestsLogging : Microsoft.IIS.PowerShell.Framework.ConfigurationElement
applicationDefaults     : Microsoft.IIS.PowerShell.Framework.ConfigurationElement
virtualDirectoryDefaults : Microsoft.IIS.PowerShell.Framework.ConfigurationElement
ftpServer              : Microsoft.IIS.PowerShell.Framework.ConfigurationElement
collection             : <Microsoft.IIS.PowerShell.Framework.ConfigurationElement>
applicationPool        : DefaultAppPool
enabledProtocols        : http
physicalPath            : %SystemDrive%\inetpub\wwwroot
userName               : 
password               : 
itemXPath              : /system.applicationHost/sites/site[Name='Default Web Site' and Id='1']
state                  : Started
attributes             : {Name, id, serverAutoStart, state}
childElements          : {Bindings, limits, logFile, traceFailedRequestsLogging...}
elementTagName         : site
methods                : {Start, Stop}
schema                 : Microsoft.IIS.PowerShell.Framework.ConfigurationElementSchema

PS IIS:\sites>

```

FIGURE 7-6 Using the IIS PowerShell Provider namespace to display the properties of a site.

Generic cmdlets such as `Get-Item` and `Select-Object` are part of the standard Windows PowerShell interface. Any module that includes a provider hierarchy must support them. Once within the IIS hierarchy, you can use low-level configuration cmdlets to manage specific IIS elements without having to type extended path names.

Using Low-Level Configuration Cmdlets

IIS 7.0, first released as part of Windows Server 2008, represents a complete revision of the IIS architecture, and extensibility was a major priority of that revision. This new architecture, carried over into the IIS 7.5 release in Windows Server 2008 R2, is schema driven and uses XML-based configuration files, which are two major contributing factors to its complete extensibility. This extensibility complicates the process of developing a Windows PowerShell management strategy, however. Cmdlets might have static parameters that enable them to manage specific properties of an element, but if a third-party developer creates an IIS extension that adds new properties to that element, the existing cmdlets cannot manage them.

Therefore, the IIS Windows PowerShell snap-in includes low-level configuration cmdlets that you can use to view and manage all of the hundreds of IIS configuration settings, including custom settings added by IIS extensions. One set of these low-level cmdlets, concerned with IIS configuration elements, is as follows:

- **Add-WebConfiguration** Adds a collection element to an existing IIS configuration collection

- **Backup-WebConfiguration** Creates a backup of an IIS configuration
- **Clear-WebConfiguration** Removes configuration settings from the specified location
- **Get-WebConfiguration** Gets an IIS configuration element at a specified location
- **Restore-WebConfiguration** Restores IIS configuration elements from a previously executed backup
- **Select-WebConfiguration** Returns Web configuration objects
- **Set-WebConfiguration** Sets an IIS configuration element to a specified value

Using Task-Oriented Cmdlets

In addition to the low-level configuration cmdlets, the IIS Windows PowerShell snap-in includes a large collection of cmdlets designed to simplify common IIS maintenance tasks, such as creating, removing, starting, and stopping specific IIS elements. One set of task-oriented cmdlets, concerned with managing IIS sites, is as follows:

- Get-Website
- New-Website
- Remove-Website
- Start-Website
- Stop-Website

Unlike the low-level cmdlets, the task-oriented cmdlets do not rely on the IIS namespace (although they can utilize it), and they use static parameters to configure specific properties. For example, to create a new Web site, you might use a command like the following:

```
New-Website -Name Intranet -Port 80 -HostHeader intra.example.local -PhysicalPath
"$env:systemdrive\inetpub\intranet" -Ssl
```

This command creates a new site with the name *Intranet*, using the default port number value 80, and using the host header value *intra.example.local* to differentiate this site from other sites that use the same address and port number. The Web site will use content files located in the *\inetpub\intranet* folder on the computer's system drive, and it will allow users to connect with SSL encryption by using the HTTPS: prefix in their URLs.

Once you have created the site, you can even use the Windows PowerShell interface to create new content. After switching to the site directory in the IIS hierarchy with the command `cd\sites\Intranet`, you can use the following command to open a Notepad window containing a new *Index.html* file:

```
notepad "$(Get-WebFilePath .)\index.html"
```

Once you have typed some Hypertext Markup Language (HTML) code into the *Index.html* file and saved it, you will have created a home page for your new site.

Using IIS Administration Pack Extensions

The IIS Administration Pack is a downloadable collection of extensions for IIS 7.0. In Windows Server 2008 R2, most of the contents of the Administration Pack are included in the IIS 7.5 implementation. For example, the ASP.NET and FastCGI configuration capabilities described earlier in this chapter were originally part of the Administration Pack, and are now incorporated into the default user interface of the Internet Information Services (IIS) Manager console in Windows Server 2008 R2. Also accessible through the console are the features described in the following sections.

NOTE The IIS Reports feature from the Administration Pack is not included in Windows Server 2008 R2, nor is the Database Manager extension, which was unbundled from the Administration Pack and repackaged as a separate download long before the R2 release. You can download both of these features from the IIS Download Center at <http://www.iis.net/downloads> and install them on a computer running Windows Server 2008 R2, if desired.

Using Configuration Editor

Configuration Editor is a graphical tool that enables administrators to view and manage any setting in any of the IIS configuration files. Because the tool is based on the IIS configuration schema, it can even manage custom settings without any interface modifications. In addition, once you have performed your modifications, the Configuration Editor can generate a script that duplicates those modifications for execution on other servers.

For example, you can use Configuration Editor to create a new IIS site, setting the same parameters as the New-Website Windows PowerShell cmdlet if desired. To do this, you open the Configuration Editor in the Internet Information Services (IIS) Manager console at the server level and, in the Section drop-down list, select `system.applicationHost/sites`, as shown in Figure 7-7.

When you open the Collection Editor window, you see the server's existing Web sites and an interface for creating a new one, as shown in Figure 7-8. You can configure a multitude of settings for the new site, after which it appears as part of the collection.

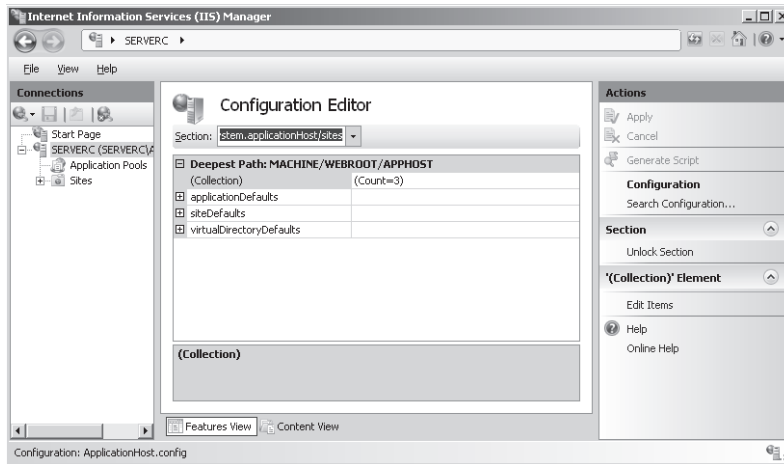


FIGURE 7-7 The IIS Configuration Editor interface.

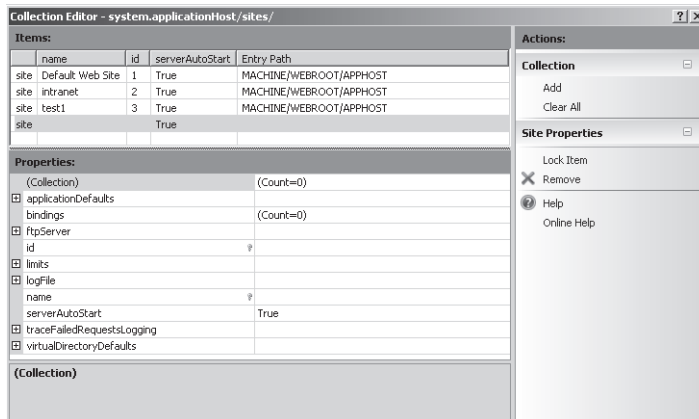


FIGURE 7-8 The sites collection in the IIS Configuration Editor.

Finally, back on the Configuration Editor page, clicking Generate Script in the Actions pane displays script code that will create a new site identical to the one you just added, using managed code (C#), JavaScript, or the Appcmd.exe program at the command prompt, as shown in Figure 7-9. From this window, you can copy the code to a text file to save for later use.

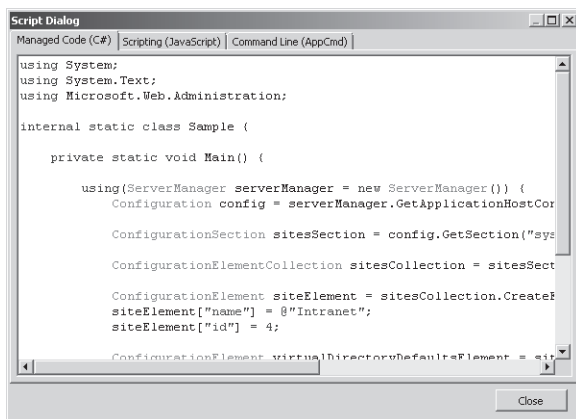


FIGURE 7-9 The Script Dialog window in the IIS Configuration Editor.

Using Request Filtering

The Request Filtering module integrates the capabilities of a separate product called Microsoft Urlscan Filter 3.1 into the default Internet Information Services (IIS) Manager console in Windows Server 2008 R2. Request Filtering is essentially a graphical interface that inserts code into Web.config files that limits the type of HTTP requests a particular IIS server or site will process. Requests that the filtering mechanism rejects are logged with error codes that indicate the reason for the rejection.

The Request Filtering page, shown in Figure 7-10, contains seven tabs that enable you to create the following types of filters:

- **File Name Extensions** Filters incoming HTTP requests based on the extension of the file requested. For example, to prevent IIS from serving any Active Server Pages files, you would add a Deny File Name Extension entry, using the extension .asp.
- **Rules** Filters incoming HTTP requests based on rules that specify text strings that cannot appear in the URL, a query string, or the HTTP header of a request for a particular file extension.
- **Hidden Segments** Filters incoming HTTP requests based on specific segments of a URL. For example, this enables you to filter out requests for files in the *bin* folder without rejecting requests for files in the *binary* folder.
- **URL** Filters incoming HTTP requests based on specified character strings in the requested URL.
- **HTTP Verbs** Filters incoming HTTP requests based on the verb specified in the HTTP message.

- **Headers** Filters incoming HTTP requests based on size limits for particular HTTP header values.
- **Query Strings** Filters incoming HTTP requests based on specific query strings. This capability is particularly useful in preventing SQL injection attacks, in which query strings contain escape characters or other damaging code.

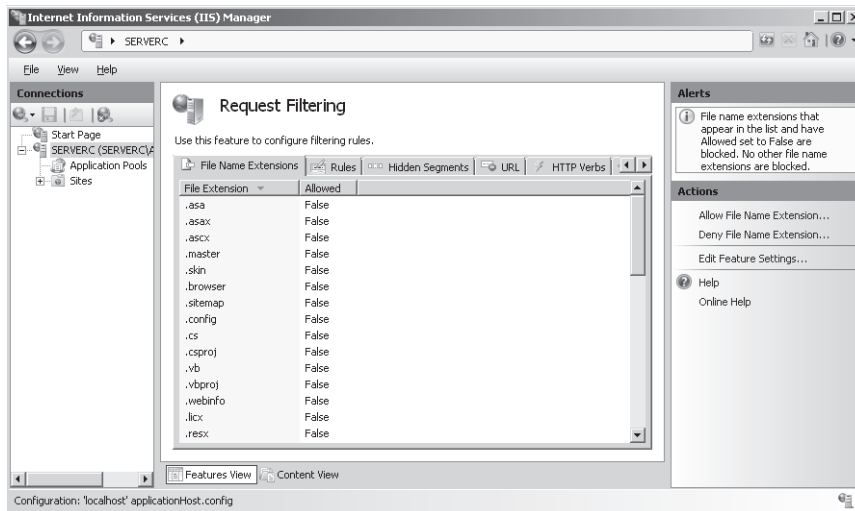


FIGURE 7-10 The Request Filtering page in the Internet Information Services (IIS) Manager console.

Creating IP Address Restrictions

The IP and Domain Restrictions role service enables you to create rules that specify which computer should be permitted (or not permitted) to access your IIS Web sites. In IIS 7.5, this role service now supports Internet Protocol version 6 (IPv6) addresses, as evidenced by the changes in the Add Allow Restrictions Rule and Add Deny Restrictions Rule dialog boxes, as shown in Figure 7-11.

In these dialog boxes, the Specific IP Address and IP Address Range fields replace those calling specifically for Internet Protocol version 4 (IPv4) addresses in IIS 7.0. In addition, the Mask or Prefix field now accepts an IPv4 mask or an IPv6 prefix, as opposed to just a mask.

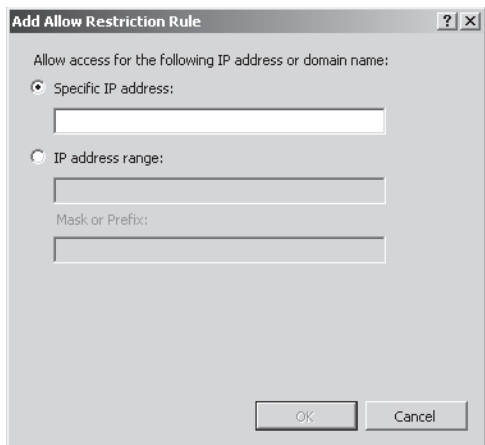


FIGURE 7-11 The Add Allow Restrictions Rule dialog box in the Internet Information Services (IIS) Manager console.

Using Configuration Tracing

Starting in version 7.5, IIS is capable of tracing and logging all modifications made anywhere in the IIS configuration system. Because all of the different IIS configuration mechanisms are essentially tools that modify the same set of configuration files, it doesn't matter if you use the Internet Information Services (IIS) Manager console, Windows PowerShell cmdlets, Appcmd.exe, or any other tool to manage IIS; the system traps any changes made to the configuration files, generates events, and adds the changes to the appropriate log.

In Windows Server 2008 R2, configuration tracing is disabled by default. To enable it, you must open the Event Viewer console, browse in the Applications and Services Logs node to the Microsoft > Windows > IIS-Configuration folder, and enable the Operational log, as shown in Figure 7-12.

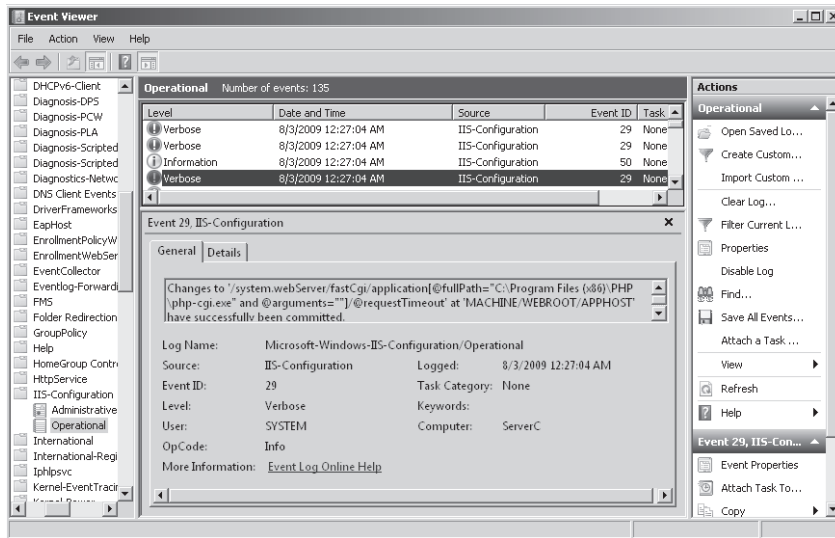


FIGURE 7-12 The IIS-Configuration log in the Event Viewer console.

Using Best Practices Analyzer

Microsoft has integrated its Best Practices Analyzer (BPA) technology into several roles in Windows Server 2008 R2, including the Web Server (IIS) role. In the Server Manager console, the Web Server (IIS) node contains a Best Practices Analyzer section, as shown in Figure 7-13. Clicking Scan This Role initiates the process by which the analyzer gathers information about IIS and compares it with a set of predefined rules. IIS conditions that differ substantially from the rules are listed in the analyzer as noncompliant results.

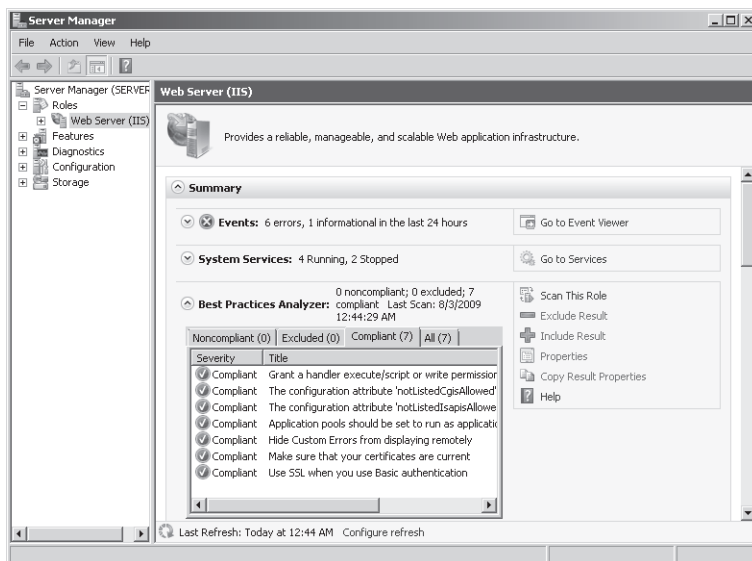


FIGURE 7-13 The Best Practices Analyzer for the Web Server (IIS) role in Server Manager.

Using New Performance Counters

The Performance Monitor console in Windows Server 2008 R2 includes two new performance objects that enable you to monitor IIS activities. The APP_POOL_WAS performance object includes counters that measure various aspects of application pool and worker process performance for each individual pool on the server. The Microsoft FTP Service performance object contains counters that track the amount of data sent and received by each FTP site on the server, and monitor the number and type of FTP connections.

Accessing IIS Resources on the Internet

IIS is one of the most complex roles in Windows Server 2008 R2, and also one of the most versatile. As a result, there is a great deal to learn about it, and there are a great many extensions and add-ons available. In addition to its regular Web site at <http://microsoft.com>, Microsoft maintains two other IIS-oriented sites: the Internet Information Services site at <http://www.iis.net> and the Microsoft Web site at <http://www.microsoft.com/web>. Both of these sites provide the latest IIS news, learning tools, community participation, and software downloads.

DirectAccess and Network Policy Server

- Introducing DirectAccess **129**
- Deploying DirectAccess **133**
- Using VPN Reconnect **140**
- New Features in Network Policy Server **142**

The percentage of the corporate workforce that relies on remote connectivity to enterprise network resources is increasing steadily. In late 2008, sales of mobile computers exceeded those of desktop computers for the first time. Many of these mobile users require access to the internal resources of their corporate networks to perform their required tasks, and Microsoft provides a number of mechanisms that enable them to do so.

Virtual private networking can provide remote clients with complete access to the company intranet, and Network Policy Server helps administrators keep remote connections safe and secure. In Windows Server 2008 R2, Microsoft has enhanced these services with new features, and also has introduced a new remote connectivity service for R2 servers and Windows 7 clients called DirectAccess.

Introducing DirectAccess

A virtual private network (VPN) connection is a secure pipeline between a remote client computer and a network server, using the Internet as a conduit. When the client establishes the VPN connection with the server, it uses a process called *tunneling* to encapsulate the intranet traffic within standard Internet packets. DirectAccess is a new feature in Windows Server 2008 R2 and Windows 7 that is similar to a VPN connection, but improves on the VPN model in several important ways.

With VPNs, the user on the client computer must explicitly launch the connection to the server, using a process similar to establishing a dial-up networking connection. The server then authenticates the user and authorizes access to the internal network

resources. Depending on the server policies, this can take several minutes. If the client loses its Internet connection for any reason, such as wandering out of a wireless hot spot, the user must manually reestablish the VPN connection.

DirectAccess, by contrast, uses connections that the client computer establishes automatically and that are always on. Users can access intranet resources without any deliberate interaction, just as though they were connected directly to the corporate network. As soon as the client computer connects to the Internet, it begins the DirectAccess connection process, which is completely invisible to the user. By the time the user is logged on and ready to work, the client can have downloaded e-mail and mapped drives to file server shares on the intranet.

DirectAccess not only simplifies the connection process for the user, it also benefits the network administrator. DirectAccess connections are bidirectional, and Windows 7 clients establish their computer connections before the user even logs on to the system. This enables administrators to gain access to the client computer at any time so they can apply Group Policy settings, deploy patches, or perform other upgrade and maintenance tasks.

Some of the other benefits of DirectAccess are as follows:

- **Intranet detection** The DirectAccess client determines whether the computer is connecting directly to the corporate network or accessing the network remotely and behaves accordingly.
- **Dual authentication** The DirectAccess client performs a computer authentication during system startup, and a user authentication during the user logon process. Users can authenticate with smart cards or biometric devices.
- **Data encryption** All of the intranet traffic exchanged by DirectAccess clients and servers is encrypted using the IPsec protocols.
- **Selective authorization** Administrators can configure DirectAccess to grant clients full access to the intranet, or limit their access to specific resources.
- **Health verification** Using Network Access Protection (NAP) and Network Policy Server (NPS), administrators can require DirectAccess clients to meet certain update and configuration requirements before they can access intranet resources.
- **Protocol flexibility** DirectAccess supports a variety of protocols that enable the computers to transmit their native Internet Protocol version 6 (IPv6) traffic over Internet Protocol version 4 (IPv4)-only networks, such as the Internet.
- **Traffic separation** In a VPN connection, all traffic generated by the client goes through the tunnel to the intranet, including traffic destined for the Internet. In DirectAccess, clients send intranet traffic through the tunnel, while the Internet traffic bypasses the tunnel and goes directly to the Internet. This is called *split-tunnel routing*.

IPv6 and IPsec

IPv6 expands the protocol's address space from 32 bits (in IPv4) to 128 bits, and it also provides globally routable addresses. The latter feature is why DirectAccess relies so heavily on IPv6 for its connectivity. Client computers can use the same IPv6 addresses wherever they happen to be in the world. Unfortunately, many networks still use IPv4, including the Internet. Therefore, DirectAccess includes support for a number of IPv6 transition technologies, which are essentially protocols that enable computers to transmit IPv6 packets over an IPv4 network. These transition technologies are as follows:

- **6to4** Provides IPv6 connectivity over IPv4 networks for hosts or sites that have public IP addresses
- **Teredo** Provides IPv6 connectivity over IPv4 networks for hosts or sites that have private IP addresses and are located behind a Network Address Translation (NAT) router
- **IP-HTTPS** Enables systems that cannot use 6to4 or Teredo to transmit IPv6 packets using a Secure Sockets Layer (SSL) tunnel
- **Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)** Provides IPv6 connectivity for DirectAccess servers and application servers on an IPv4-only intranet
- **Network Address Translation–Protocol Translation (NAT-PT)** Hardware device that enables DirectAccess clients to access applications that do not support IPv6

Internet Protocol Security (IPsec) is a set of extensions to IP that enables computers to secure data using authentication, data integrity, and encryption services before they transmit it. DirectAccess uses IPsec to authenticate client computers and users, and to ensure that the private intranet data that clients and servers transmit over the Internet remains private. IPsec provides end-to-end security, meaning that only the source and final destination systems can read the contents of the encrypted data packets. This also means that intermediate systems—the routers that forward packets through the Internet to their destinations—do not have to support IPsec.

When a client connects to a DirectAccess server, it creates two separate IPsec tunnels. The first connection uses a computer certificate and enables the client to access the Domain Name System (DNS) server and the Active Directory Domain Services (AD DS) domain controller on the intranet. With this access, the client can download Group Policy objects and initiate the user authentication process. The client then uses the second connection to authenticate the user account and access the intranet resources and application servers.

IPsec supports two protocols, Authenticated Header (AH) and Encapsulating Security Payload (ESP), and two operational modes, transport mode and tunnel mode. In transport mode, IPsec provides protection for the application data that IP datagrams carry as their payload. In tunnel mode, IPsec protects the entire IP datagram, including the header and the payload. DirectAccess uses the ESP protocol for its authentication and encryption capabilities. The

operational mode that DirectAccess uses depends on the access model you choose for your deployment.

The degree to which your intranet and the computers on it support IPv6 and IPsec is a critical factor in how you will deploy DirectAccess on your enterprise network. DirectAccess clients and servers, which must run Windows 7 or Windows Server 2008 R2, all have full support for IPsec connections using IPv6, but your application servers might not. Even if this is the case, however, it is still possible to use DirectAccess, as described in the section “Deploying DirectAccess,” later in this chapter.

Understanding the DirectAccess Connection Process

The process by which a DirectAccess client establishes a connection to a DirectAccess server, and thereby to the company intranet, is a complicated one. However, the process is completely invisible to the user on the client computer. The DirectAccess server processes the client’s connection request, authenticates the client computer and the user, and authorizes the user to access applications and other resources on the intranet. The individual steps of the connection process are as follows:

1. The client attempts to connect to a designated Web server on the intranet. The availability of the Web server indicates that the client is directly connected to the intranet. The inability to access the Web server indicates that the client is at a remote location. The client then proceeds to initiate a DirectAccess connection to access the intranet.
2. The client establishes its first connection to the DirectAccess server on the intranet. By default, the client attempts to connect using IPv6 and IPsec natively, but if an IPv6 connection is not available (such as when the client is connected to the IPv4 Internet), it uses 6to4 or Teredo, depending on whether the computers have public or private IPv4 addresses. If the client cannot connect using 6to4 or Teredo due to an intervening firewall or proxy server, it uses IP-HTTPS as a last resort, to connect to the server using the SSL port.
3. Once the client is connected to the DirectAccess server, the two computers authenticate each other using their respective computer certificates. Once the computer authentication is complete, the client has access to the domain controller and the DNS server on the intranet. The process up to this point can occur before the user logs on to the client computer.
4. The client establishes its second connection to the DirectAccess server and, using the domain controller access it obtained from the first connection, performs a standard AD DS user authentication, using NTLMv2 credentials and the Kerberos V5 authentication protocol.
5. The DirectAccess server authorizes the client to access intranet resources by checking the AD DS group memberships for the computer and the user.

6. If the server is configured to require health validation, the client submits a health certificate to an NPS, which verifies that the client complies with the appropriate policies.
7. The DirectAccess server begins functioning as a gateway between the client computer and the application servers and other resources the client is authorized to use.

Deploying DirectAccess

Deploying DirectAccess on a network is a relatively complicated process, requiring careful planning, a detailed understanding of the network's capabilities, and the installation of various supporting infrastructure resources. The following sections provide a high-level overview of the deployment process.

Choosing an Access Model

The access model you choose for your DirectAccess deployment specifies where on your intranet the IPsec encryption will terminate and how the traffic to and from the client will proceed once it passes through the DirectAccess server. The basic architecture of a DirectAccess deployment is shown in Figure 8-1. The client is at a remote location, typically connected to the Internet. The corporate intranet, protected behind a firewall, has a DirectAccess server on a perimeter network, which makes it directly accessible from the Internet using a public IP address. Clients connect to the DirectAccess server, and the server forwards their traffic to the other resources on the intranet.

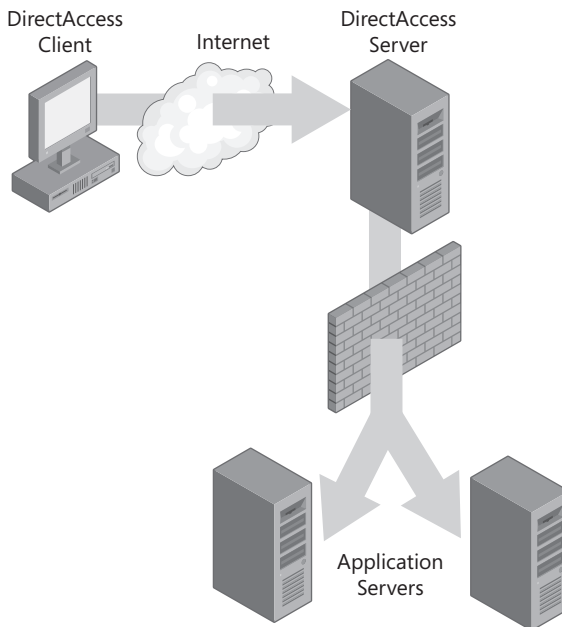
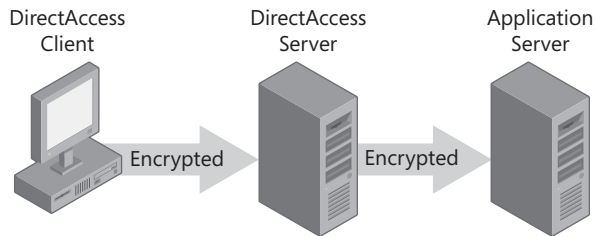


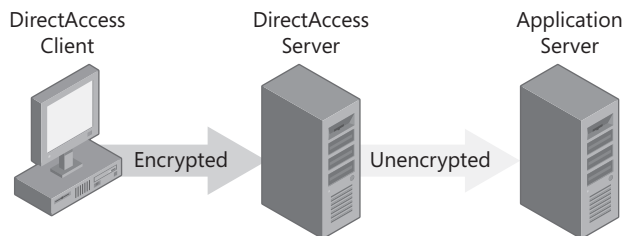
FIGURE 8-1 The DirectAccess connection architecture.

The access model you choose for your deployment specifies how the DirectAccess server will forward the client traffic to the resources on the intranet. There are three access models supported by DirectAccess, as follows:

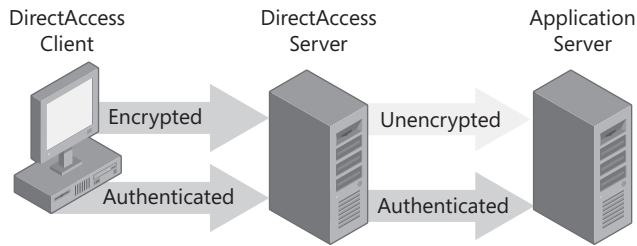
- **End-to-end** In this model, DirectAccess clients establish transport mode ESP connections that go through the DirectAccess server and all the way to the individual application servers on the intranet, as shown in the following graphic. This is the ideal solution from a security standpoint, but it requires all of the application servers to support IPsec connections using IPv6. This means that the application servers must all be running Windows Server 2008 or Windows Server 2008 R2 and be configured to use both IPv6 and IPsec.



- **End-to-edge** In this model, DirectAccess clients establish tunnel mode connections to an IPsec gateway server, which is typically (but doesn't have to be) the computer functioning as the DirectAccess server. The IPsec gateway server then forwards the client traffic, now protected by IPsec, to the application servers on the intranet, as shown in the following graphic. This model keeps IPsec traffic off of the intranet and enables you to use application servers that run Windows Server 2003, or any other operating system that supports IPv6.



- **Modified end-to-edge** This model is identical to the end-to-edge model, except that it uses an additional IPsec tunnel that authenticates clients at the application server. Client traffic is therefore encrypted only as far as the IPsec gateway server, but it is authenticated all the way to the application server, as shown in the following graphic. The need for this additional authentication also makes it easier for administrators to limit client access to specific application servers. To use this model, application servers must be running Windows Server 2008 R2.



All of these access model descriptions assume that the intranet applications and resources are all capable of supporting IPv6 connections to the DirectAccess server. However, if this is not the case, the intranet needs some kind of IPv4-to-IPv6 transition mechanism, such as ISATAP or a NAT-PT device.

MORE INFO The Windows Server 2003 operating system itself supports IPv6, but many of its built-in applications and services do not. Windows Server 2003 also supports IPsec, but it does not support the use of IPsec over IPv6 connections. If you have IPv6-capable applications or services running on Windows Server 2003 servers, DirectAccess clients can reach them only if you use the end-to-edge or modified end-to-edge access model. If you have applications or services that only support IPv4 on your Windows Server 2003 servers, DirectAccess clients can only reach them if you use the end-to-edge or modified end-to-edge access model and have a NAT-PT device installed on your intranet.

DirectAccess Server Requirements

The computer that functions as the DirectAccess server must be running Windows Server 2008 R2, and it must meet the following additional requirements:

- **Member server** The computer must be a member of an AD DS domain.
- **Two network adapters** The computer must have two network interface adapters, with one connected to the public Internet and one to the company intranet.
- **Two IPv4 addresses** To support Teredo, the computer must have two consecutive IPv4 addresses that are static, public, and resolvable using the Internet DNS.
- **Direct Internet access** The computer must not use NAT to access the Internet.

DirectAccess Client Requirements

The computers that function as the DirectAccess clients must be running Windows 7 Enterprise or Ultimate Edition or Windows Server 2008 R2. The clients must also be joined to the same domain as the DirectAccess server. This means that before clients can use DirectAccess to connect to the intranet from remote locations, you must deploy their computers on the intranet so that they can first receive certificates and Group Policy settings.

DirectAccess Infrastructure Requirements

In addition to the DirectAccess server and clients, the company intranet must include the following services, features, and policies in its network infrastructure to support DirectAccess:

- **Active Directory Domain Services** The intranet must have an AD DS domain, with at least one DNS server and one domain controller running on Windows Server 2008 R2.
- **Group Policy** The AD DS computer objects for DirectAccess client computers must be members of a security group that will enable them to receive DirectAccess settings using Group Policy.
- **Public Key Infrastructure (PKI)** The intranet must have a certification authority that can provide DirectAccess clients and servers with the certificates they will use for authentication.
- **Network detection server** DirectAccess requires a Web site that is accessible only from the intranet, which clients can use to determine whether they are currently connected to the intranet.
- **Certificate revocation list (CRL)** DirectAccess requires that the CRL for the intranet detection site's SSL certificate must be published on a distribution point that is accessible from the intranet. In the same way, CRLs for IP-HTTPS certificates must be accessible from a distribution point on the Internet.
- **ICMPv6 policies** For DirectAccess clients to access the intranet from the Internet using Teredo, the DirectAccess server must have firewall policies that permit inbound Internet Control Message Protocol version 6 (ICMPv6) Echo Request messages.
- **IPv6 and transition technologies** DirectAccess clients must be able to communicate with the DirectAccess server, an intranet domain controller, and the application servers on the intranet using IPv6, or one of the transition technologies that enables IPv6 communications using IPv4, such as 6to4, Teredo, IP-HTTPS, ISATAP, and NAT-PT.
- **Firewall exceptions** For DirectAccess clients to communicate with the DirectAccess server, the firewalls on the Internet interface of each computer must admit traffic through the appropriate ports, based on the protocols the clients are using. The same is true for any firewall between the DirectAccess server and the application servers on the intranet.

Configuring DirectAccess

To create a DirectAccess server, you must first install the DirectAccess Management Console feature on Windows Server 2008 R2 using the Add Features Wizard in Server Manager. You can then open the DAMgmt console and run the DirectAccess Setup wizards to configure the server. When you select the Setup node in the DAMgmt console, the console displays any of the DirectAccess prerequisites that the server does not meet, as shown in Figure 8-2.

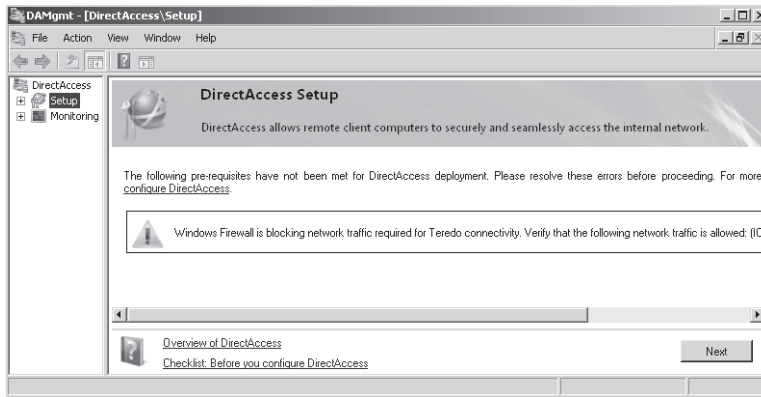


FIGURE 8-2 The prerequisite check in the DAMgmt console.

If the console detects that the server meets all of the prerequisites, a diagram appears that outlines the configuration steps you will perform, as shown in Figure 8-3.

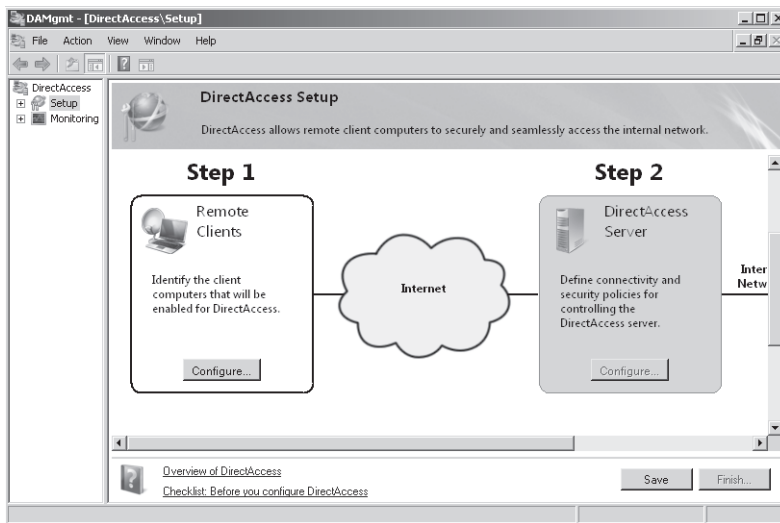
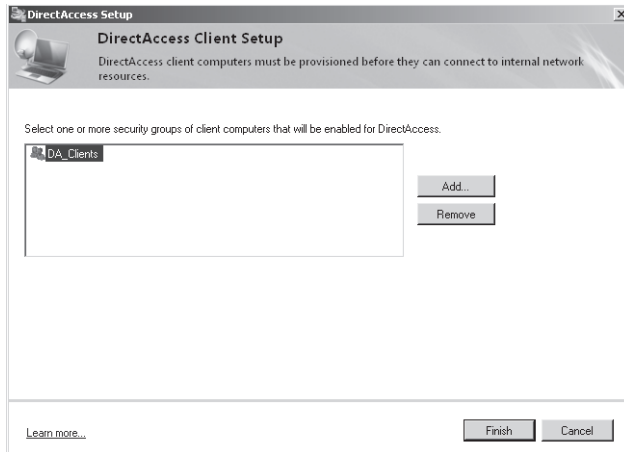


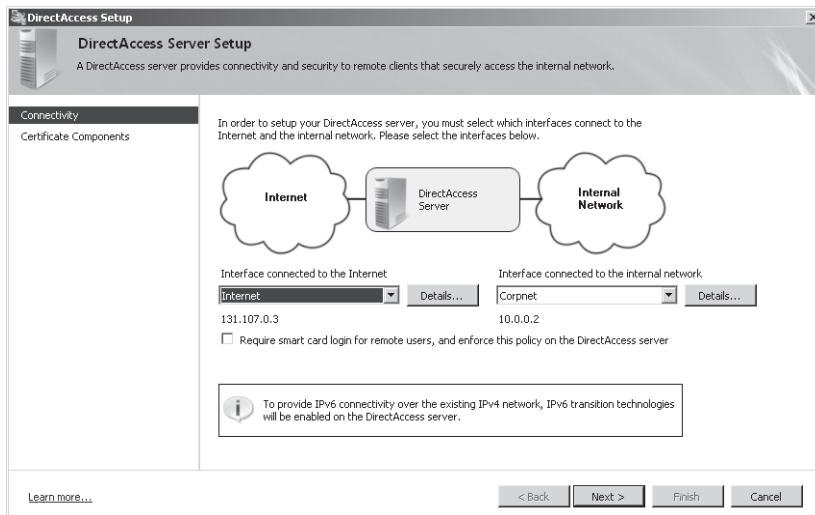
FIGURE 8-3 The DirectAccess Setup page.

The four DirectAccess server configuration steps proceed as follows:

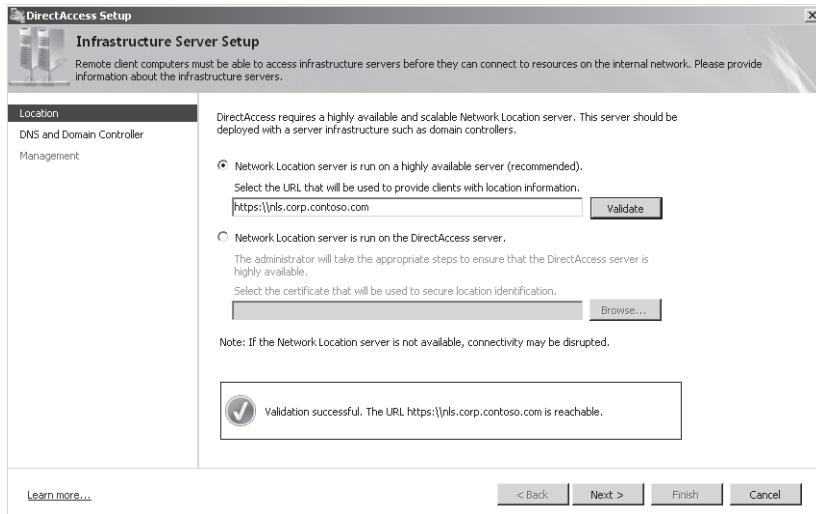
- In step 1, in the DirectAccess Client Setup Wizard, shown in the following graphic, you specify the name of the security group that contains your client computers as members.



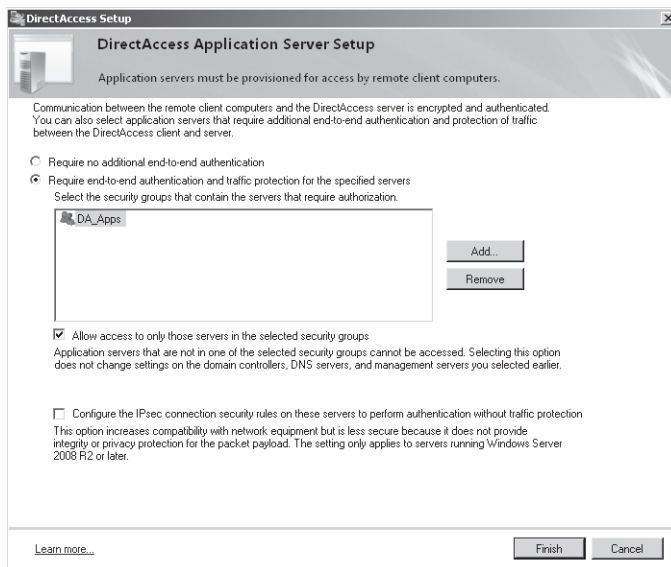
- In step 2, in the DirectAccess Server Setup Wizard, shown in the following graphic, you specify which of the server's two network interfaces provides access to the Internet and which provides access to the intranet. Then, you select the certificates that the server will use.



- In step 3, in the Infrastructure Server Setup Wizard, shown in the following graphic, you specify and validate the Uniform Resource Locator (URL) of the network location server you created.



- In step 4, in the DirectAccess Application Server Setup Wizard, shown in the following graphic, you can specify which application servers on the intranet require additional authentication and specify security groups that limit client access to specific applications.



Once you have completed the DirectAccess server setup process, you must force a Group Policy update on your clients who are still connected to the intranet to make sure they receive the Group Policy settings created by the server setup process. Once this is done, you can disconnect the client computers from the intranet and deploy them to their remote locations.

The clients will then initiate DirectAccess connections to the DirectAccess server whenever they are connected to the Internet.

Using VPN Reconnect

If you and your network are not yet ready for the admittedly large commitment that DirectAccess requires, VPNs are still a viable solution that the Windows Server 2008 R2 development teams have certainly not abandoned. Windows Server 2008 R2 and Windows 7 still have their VPN server and client capabilities, and Microsoft's new Forefront Intelligent Application Gateway (IAG) product is based largely on SSL VPN technology.

As mentioned earlier in this chapter, one of the major problems with VPNs is the need to manually reestablish the VPN connection whenever the underlying Internet connection is interrupted for any reason. This is particularly true for wireless clients. A Wi-Fi or mobile broadband connection reestablishes itself automatically after an outage, but the VPN connection does not. This is an annoyance for the remote user, who might have to spend several minutes reconnecting, and it imposes an additional burden on the VPN servers that have to reauthenticate and reauthorize the user. For organizations that use VPN tunnels to connect branch offices to a home office network, service outages can inconvenience whole offices and require IT staff to constantly monitor their VPN connections.

VPN Reconnect, formerly known as agile VPN, is a new feature in Windows Server 2008 R2 and Windows 7 that addresses this problem by enabling a VPN connection to persist, even when its underlying connection is lost. When the computer loses its interface, it can switch the VPN to another interface without having to reestablish the connection. This capability can benefit users in a number of scenarios, including the following:

- A computer with an underlying interface that provides intermittent connectivity
- A mobile computer that moves from one wireless access point to another
- A computer connected using mobile broadband that arrives at the office and connects to the corporate network
- A computer connected with one interface that suffers a service outage and has another interface available
- A computer that moves between an IPv4 network and an IPv6 network

In each of these cases, when a client's interface is lost or interrupted, a connection using VPN Reconnect can persist until the system switches to another interface or until the original interface is resumed. The VPN tunnel itself remains intact; only its endpoints change.

VPN Reconnect is based on an IPsec tunnel-mode connection using the Internet Key Exchange version 2 (IKEv2) protocol and its mobility and multihoming extension, called MOBIKE. IKEv2 enables the VPN client and server to authenticate each other and create a security association that includes the shared secret information and cryptographic algorithms needed to build IPsec connections using the ESP or AH protocol. IKEv2 builds these

connections between two specific IP addresses, which are the endpoints of the tunnel. The role of the MOBIKE protocol is to enable the IKEv2 connection to exchange one endpoint (or IP address) or another without breaking down the tunnel between them.

Support for IKEv2 VPN connections is built into the network connection client in Windows 7 and Windows Server 2008 R2. After you create a VPN connection, open its Properties dialog box and, on the Security tab, in the Type Of VPN drop-down list, select IKEv2, as shown in Figure 8-4.

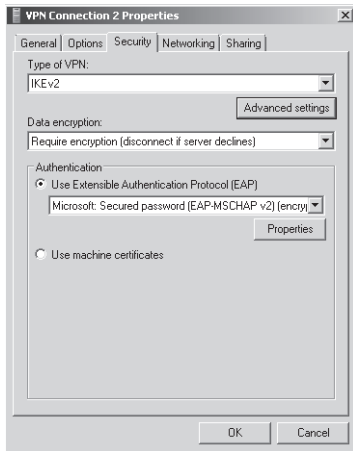


FIGURE 8-4 An IKEv2 VPN connection's Properties dialog box.

Clicking Advanced Settings displays the Mobility control, as shown in Figure 8-5, which is enabled by default. You can also specify the amount of time that your VPN connection will persist after a network outage occurs. The default value is 30 minutes.

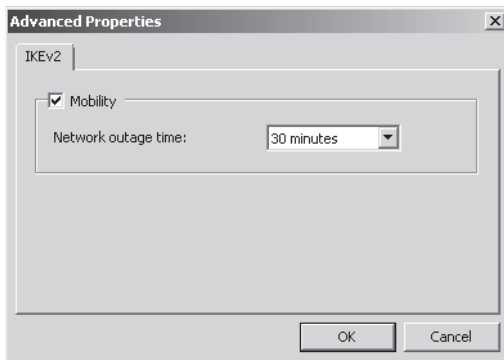


FIGURE 8-5 The Advanced Properties dialog box for an IKEv4 VPN connection.

On the server side of the connection, you must use Routing and Remote Access Services (RRAS) in Windows Server 2008 R2. Opening the server's Properties dialog box in the Routing

and Remote Access console and selecting the IKEv2 tab, as shown in Figure 8-6, displays controls that enable you to calibrate the persistence of the server's IKEv2 connections and security associations.

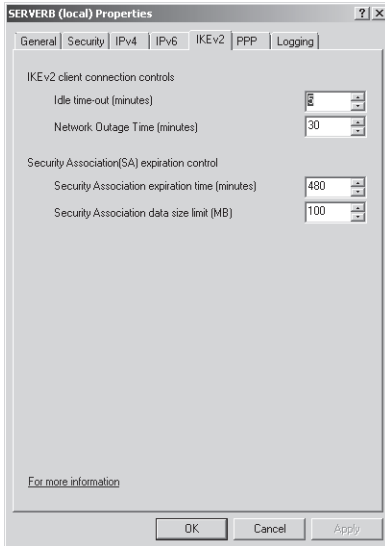


FIGURE 8-6 The IKEv2 controls in an RRAS server's Properties dialog box.

New Features in Network Policy Server

NPS is the replacement for Microsoft Internet Authentication Service (IAS). First appearing in Windows Server 2008, NPS enables a Windows server to perform the following functions:

- **Remote Authentication Dial-In User Service (RADIUS) server** Provides authentication, authorization, and accounting services for network access devices such as remote access servers and wireless access points
- **RADIUS proxy** Forwards RADIUS message traffic generated by network access devices to RADIUS servers on other networks
- **Network Access Protection (NAP) health policy server** Enables administrators to create and enforce health policies that stipulate software, update and configuration requirements for IPsec, 802.1X, VPN, Dynamic Host Configuration Protocol (DHCP), and Remote Desktop Gateway clients. Clients not meeting the health policy requirements are denied access to the network resources.

In Windows Server 2008 R2, Microsoft has added a number of new administrative tools to NPS, as described in the following sections.

Configuring NPS Logging

NPS has always been able to save its accounting log data to a SQL database, either on the local server or a remote one. The version of NPS in Windows Server 2008 R2 enhances this capability, however, in two ways.

First, NPS now enables you to mix SQL and text file logging in several combinations, using the interface shown in Figure 8-7. You can maintain SQL and text file logs individually; you can also combine the two by logging to both simultaneously or by logging to the SQL database and using text files as a failover option should the database be unavailable.

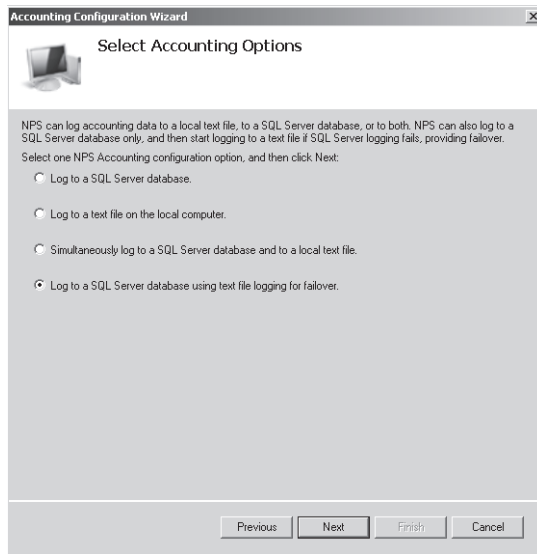


FIGURE 8-7 Network Policy Server logging options.

Second, NPS now simplifies the process of configuring SQL database logging. When you configure the SQL server logging options, using the Accounting Configuration Wizard interface shown in Figure 8-8, you can either specify the name of an existing instance on your SQL Server computer or have the wizard create a new instance for you simply by specifying the name you want to use.

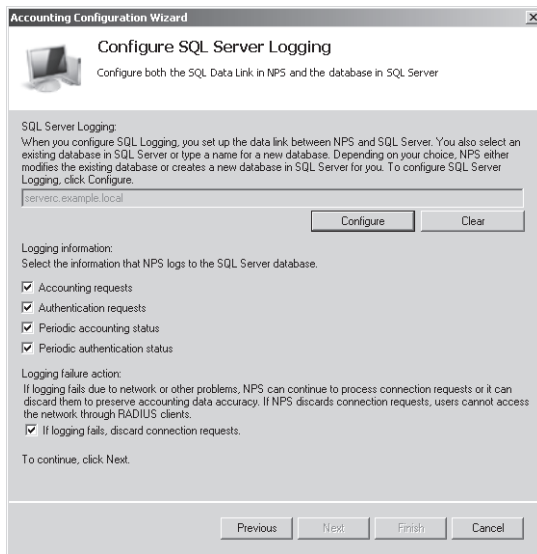


FIGURE 8-8 The Configure SQL Server Logging page in the Accounting Configuration Wizard.

Using NPS Templates

The most exciting new feature in the Windows Server 2008 R2 NPS implementation is the introduction of NPS templates. In NPS, templates are collections of configuration settings that exist as elements separate from the standard NPS configuration settings. When you create a template, you specify values for certain settings and save them for later use. When you configure an NPS feature, you can, in many cases, specify the template you want to use instead of configuring individual settings. The feature then inherits the settings you specified in the template. At a later time, you can modify the settings in your templates, and all of the features that use the templates are automatically updated as well.

For example, when you create a new RADIUS client in the Network Policy Server console, you have the option of specifying a shared secret manually or letting the program generate one for you. NPS in Windows Server 2008 R2 now offers another option: you can select a Shared Secret template instead. When you create a Shared Secret template, using the New RADIUS Shared Secret Template dialog box shown in Figure 8-9, you see basically the same Shared Secret controls as in the New RADIUS Clients dialog box.

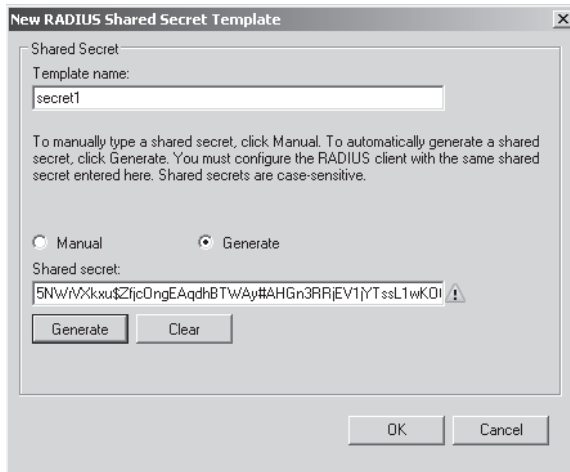


FIGURE 8-9 The New RADIUS Shared Secret Template dialog box.

Selecting a template when you create a new client, as shown in Figure 8-10, causes the shared secret you entered or generated in the template to be plugged into the client. This enables you to easily use the same shared secret value for multiple clients. Then, you can change the secret in all of your clients at one time simply by modifying the secret value in the template. To make things even easier, you can also simplify the process of creating new RADIUS clients by using a RADIUS Client template.

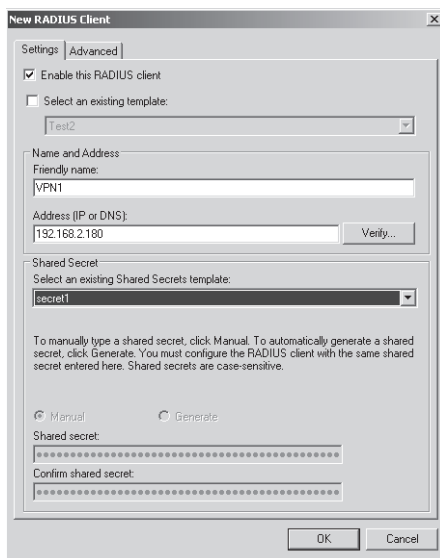


FIGURE 8-10 The New RADIUS Client dialog box.

NPS supports six types of templates, which you can access in the new Templates Management node of the Network Policy Server console. These six templates are as follows:

- Shared Secrets
- RADIUS Clients
- Remote RADIUS Servers
- IP Filters
- Health Policies
- Remediation Server Groups

Migrating IAS Configuration Settings

IAS, the previous version of the Microsoft RADIUS server product, stores its configuration settings in a Microsoft Access database file with the extension .mdb. NPS stores its configuration settings as Extensible Markup Language (XML) files. When you upgrade a computer running Windows Server 2003 with IAS installed to Windows Server 2008, the setup program migrates the IAS settings to the NPS format. However, upgrading the operating system is the only way to do this. NPS has an Import Configuration function, but it cannot read IAS database files. There is no way to export the settings from IAS and import them into NPS on Windows Server 2008 without performing an operating system upgrade.

Windows Server 2008 R2 resolves this problem by including a command prompt utility called `lasmigreader.exe` that saves the configuration settings on an IAS server in a text file format that you can import into NPS. To use the utility, copy the 32- or 64-bit version of the `lasmigreader.exe` file from a computer running Windows Server 2008 R2 to your IAS server and run it from the command prompt. The program creates a file called `las.txt`, which contains all of the IAS configuration settings. You can then copy this file to the server running R2 and import it by using the `Netsh.exe` utility at the command prompt, as in the following example:

```
Netsh nps import e:\ias.txt
```

IMPORTANT The `las.txt` file created by the `lasmigreader.exe` program contains shared secret data from the IAS configuration. Be sure to store the file in a safe place to avoid compromising this sensitive information.

Other Features and Enhancements

- Using Windows Server Backup **147**
- BitLocker ToGo **158**

The previous chapters covered most of the new features and capabilities in Windows Server 2008 R2, but there are still a few topics that don't fit neatly into the areas already covered. The following sections discuss some of these features.

Using Windows Server Backup

The Windows Server Backup utility provided with Windows Server 2008 was completely different from the backup program included with earlier Windows Server versions. Unlike previous versions and most commercial backup products, the new program is designed primarily to back up entire volumes to an external hard disk drive. The program also uses a different format for its backup files; it uses the Microsoft Virtual Hard Disk (VHD) format, which makes the files accessible to Hyper-V, Virtual PC, and the Complete PC backup utility.

The Windows Server 2008 backup utility also had some distinct shortcomings, however. It could only back up and restore entire volumes, not individual files and folders, and it required you to designate an entire disk as a backup disk, preventing you from using that disk for anything else. The Windows Server Backup program in Windows Server 2008 R2 addresses these shortcomings, and includes a number of additional improvements, as described in the following sections.

Backing Up Selected Files and Folders

The Windows Server 2008 version of Windows Server Backup enables you to back up your entire server or selected volumes on that server; however, you cannot select individual files or folders for backup. The Shadow Copies for Shared Folders feature eliminates the need for individual file and folder backups and restores to some degree, but many administrators have requested this feature. Therefore, when you choose the

Custom configuration option in Windows Server 2008 R2, both the Backup Once Wizard and the Backup Schedule Wizard enable you to select individual items for backup, using the interface shown in Figure 9-1. Unlike Windows Server 2008, you can also perform a scheduled backup that excludes the system drive.

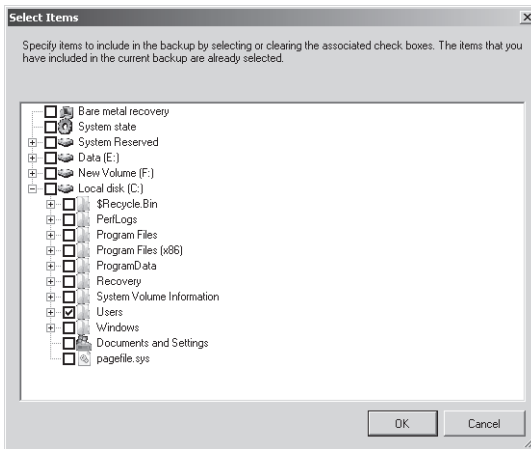


FIGURE 9-1 The Select Items dialog box from the Backup Once Wizard and the Backup Schedule Wizard in Windows Server Backup.

In addition to individual file and folder selection, the program also enables you to create exclusions. An *exclusion* is a filter that prevents a job from backing up specified files or file types in the selected targets. For example, if you want to back up all of a server's Data volume except for the video files, you can either browse through the entire volume in the Select Items dialog box and select everything but the video files, or you can select the entire volume and create an exclusion for the video files.

To create exclusions, go to the Select Items For Backup page of the Backup Once Wizard or Backup Schedule Wizard and click Advanced Settings. Click Add Exclusion and select a file or folder to exclude in the Select Items To Exclude dialog box, shown in Figure 9-2.

To exclude an entire file type instead of a specific file or folder, you can modify an entry in the Excluded File Types list by adding standard wildcard characters, as shown in Figure 9-3.

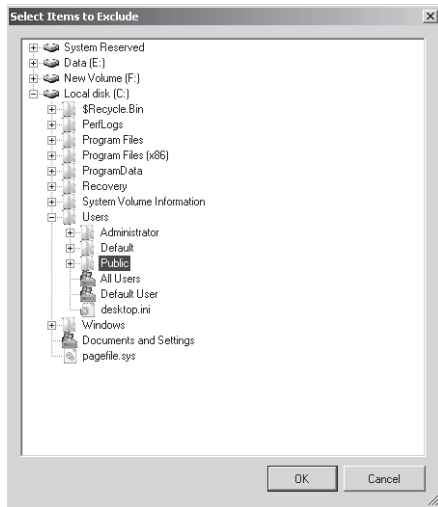


FIGURE 9-2 The Select Items To Exclude dialog box from the Backup Once Wizard and the Backup Schedule Wizard in Windows Server Backup.

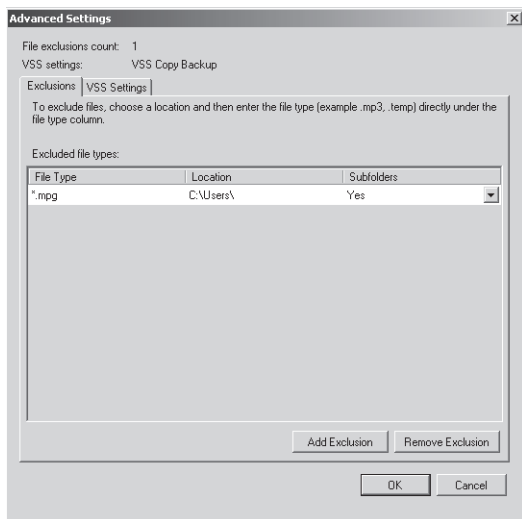


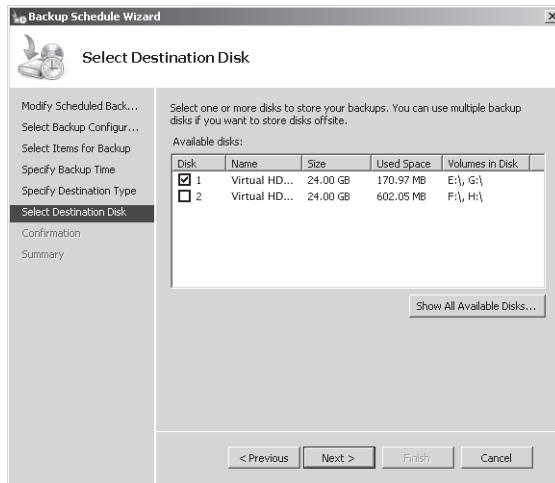
FIGURE 9-3 The Exclusions tab of the Advanced Settings dialog box from the Backup Once Wizard and the Backup Schedule Wizard in Windows Server Backup.

Selecting a Backup Destination

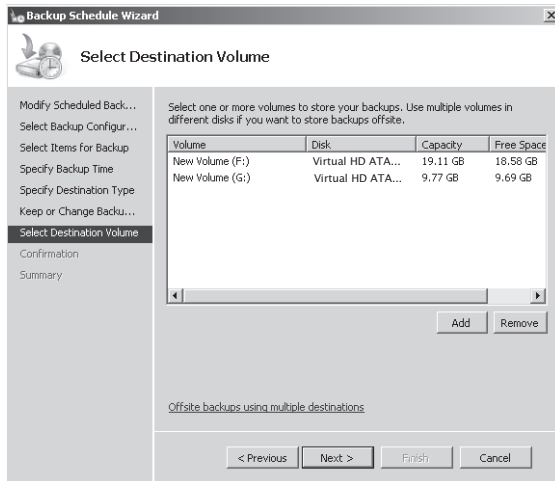
In the Windows Server 2008 version of Windows Server Backup, when you create a scheduled backup job, you have to select a local disk (not a volume) to function as the backup drive. The Windows Server 2008 R2 version provides additional options.

In the Backup Schedule Wizard, after you select the items you want to back up and create a schedule, the Specify Destination Type page appears, providing the following three options:

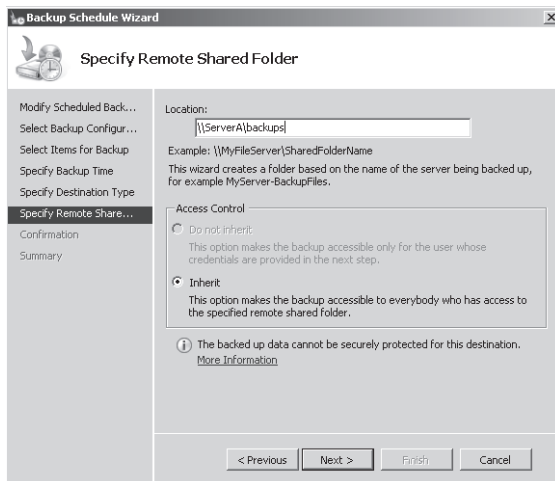
- **Back Up To A Hard Disk That Is Dedicated For Backups (Recommended)** This option requires that you allocate an entire disk as the backup drive, using the interface shown in the following graphic. The wizard reformats the disk and dedicates it to that purpose exclusively. You cannot use the disk for anything else, nor can you access it using standard file management tools such as Windows Explorer. This is the default option in Windows Server 2008 R2 and the only option in Windows Server 2008.



- **Back Up To A Volume** This option enables you to select a specific volume for backups instead of an entire disk, using the interface shown in the following graphic. The wizard creates a folder on the volume called `WindowsImageBackup`, beneath which there are subfolders containing the backup files and the catalog of backed up files, but the rest of the folder remains available for use in the normal manner. The drawback of this option is that the backup jobs are slowed down by as much as 200 percent.



- **Back Up To A Shared Network Folder** This option enables you to specify a shared folder on another computer as the destination for your backups, using a Universal Naming Convention (UNC) designation in the format `\\server\share`, as shown in the following graphic. After you specify the destination and press Enter, the wizard prompts you for credentials that it should use to access the share. Backing up to a remote share prevents Windows Server Backup from performing incremental jobs. Each time the backup job runs, it overwrites the existing backup files on the specified share.



TIP If you select more than one disk or volume as the backup destination, the program creates a separate copy of the backup on each of the destinations you select. This enables you to use external media for offsite storage, as well as one of the server's internal disks.

Creating Incremental Backups

An *incremental backup* is a backup job that only saves the files that have changed since the last backup job. Traditional tape backup software products use incremental jobs to save tape and reduce backup times. To perform restores—or recoveries in Windows Server Backup parlance—you have to restore the last full backup job and each of the subsequent incremental jobs, so that you have the most recent version of each file. Windows Server Backup supports incremental jobs, but because the product is designed to back up to hard disks and not tape, it approaches the jobs in a different manner.

Unlike traditional backup software products, you cannot elect to perform incremental backups on a job-by-job basis in Windows Server Backup. In the Windows Server 2008 version, the program performs full backups by default until the destination disk is filled (or contains 512 jobs) and then begins deleting the oldest backups. If you select the Always Perform Incremental Backup option in the Optimize Backup Performance dialog box, the program performs a full backup first and then performs incremental backups for the next 14 days (or 14 jobs) after that.

In Windows Server 2008 R2, Windows Server Backup always performs incremental jobs by default, but it can do so in two different ways depending on the options you choose in the Optimize Backup Performance dialog box, as shown in Figure 9-4.

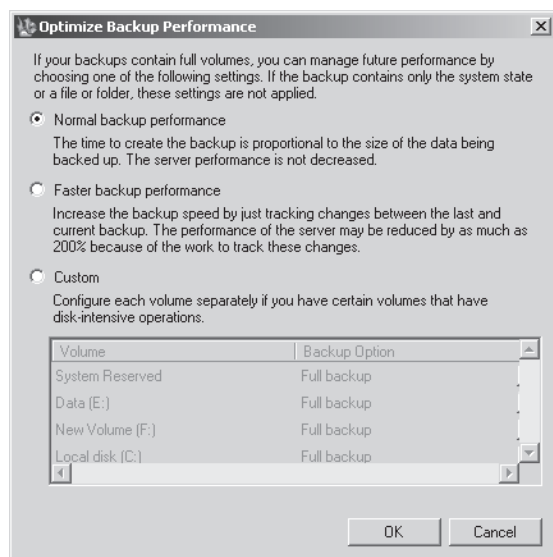


FIGURE 9-4 The Optimize Backup Performance dialog box in Windows Server Backup.

The options in the Optimize Backup Performance dialog box are as follows:

- **Normal Backup Performance** The system transfers all of the selected source files to the destination medium, overwriting the files that are the same. Only the files that have changed consume additional storage space.
- **Faster Backup Performance** During the initial full backup, the system creates a shadow copy on the source drive(s) to track the changes made to the files. During the next backup, the program uses the shadow copy to select the files that have changed and transfers only those files to the destination medium. This speeds up the backup process substantially, but maintaining the shadow copy can degrade the write performance of the source disk.
- **Custom** This option enables you to configure Windows Server Backup to perform full or incremental backups for each individual volume on the server.

The primary advantage of the incremental backup support in Windows Server Backup is that the recovery process does not require any version management from the administrator. When you perform a recovery, the program automatically integrates the appropriate version of each file into the recovered folders.

Backing Up the System State

In Windows Server 2008 R2, the Windows Server Backup program also provides additional options for backing up the system state elements. In Windows Server Backup, the System State is a collective term for a group of operating system elements that are not normally accessible by the file system when the computer is running. The System State includes the Windows Registry, the Active Directory database (if the system is a domain controller), and a number of files that are locked open by the operating system.

Unlike the Windows Server 2008 version, the Select Items dialog box in Windows Server 2008 R2 enables you to individually select the System State element and a Bare Metal Recovery element. Selecting System State backs up the elements listed earlier, independent of the drive on which they are stored. In Windows Server 2008, you can only back up the System State elements along with the system drive.

When you select the Bare Metal Recovery element, the wizard also selects the System State item; the System Reserved partition, which contains the boot files; the system drive; and any other drives in the computer; in short, everything you need to restore the entire server to a new computer or a new hard disk. The best practice is to perform a Bare Metal Recovery backup to an external hard drive, so you can easily access it from a new computer.

To recover an entire computer, you connect your external drive containing the backup to the new computer and boot from the Windows Server 2008 R2 installation disk. Select Repair Your Computer in the Windows Setup Wizard, and in the System Recovery Options dialog box that appears, as shown in Figure 9-5, select Restore Your Computer Using A System Image That You Created Earlier.



FIGURE 9-5 The System Recovery Options dialog box.

The system scans the external drive and enables you to select an image on it, using the interface shown in Figure 9-6. The recovery process formats the drive(s) in the new computer and recovers the data from the backup, rebuilding the system to the exact state it was in when you performed the backup.

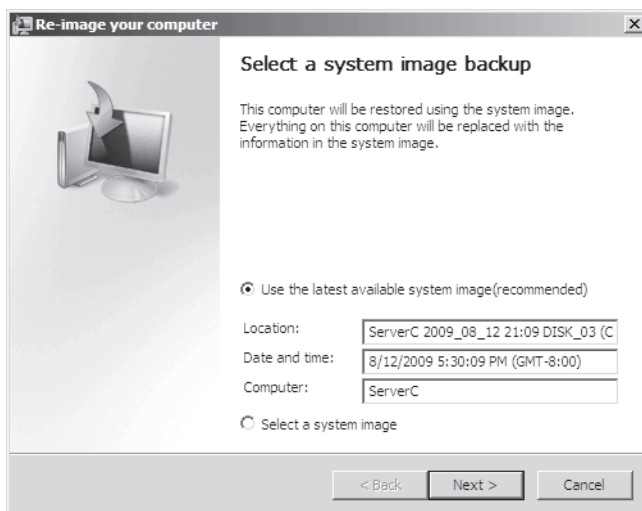


FIGURE 9-6 The Re-Image Your Computer Wizard.

Backing Up Hyper-V

Hyper-V complicates the problem of backing up a server running Windows Server 2008 R2. The big question is whether to back up the host server running Hyper-V or back up the virtual machines (VMs) individually, using internal software. Both alternatives have advantages and disadvantages.

Running Hyper-V Host Server Backups

If you back up the host server, it is possible in most cases to include the VMs in those backups, thus protecting the entire system with one process. The Hyper-V Volume Shadow Copy Service (VSS) Writer is the component that makes it possible to back up the VMs from the host system. To use Windows Server Backup to protect an entire Hyper-V server and its VMs, you must register the VSS Writer with the backup software by creating the following registry key:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\WindowsServerBackup\
Application Support\{66841CD4-6DED-4F4B-8F17-FD23F8DDC3DE}
```

Then, in that registry key, you must create a String Value with the following settings:

- Name: Application Identifier
- Type: REG_SZ
- Value: Hyper-V

With this registry setting in place, a full backup of the host server will include the VM configuration settings, the VM snapshots, and the virtual hard drives in the VMs. However, this type of backup will not include virtual networks, nor will it include certain types of storage. The VSS Writer can access virtual hard disks, as stated, but it cannot back up pass-through disks; that is, it cannot back up physical disks directly attached to a VM, nor can it back up iSCSI storage when the initiator is running inside a virtual VM.

Running Internal Virtual Machine Backups

Running backup software such as Windows Server Backup inside VMs has the obvious disadvantage that, in the event of a catastrophic server failure, you must rebuild the Hyper-V host server and recreate the VM settings before you can recover the VMs themselves. However, internal VM backups can address physical disks and iSCSI storage running inside the VM. Also, the new Hyper-V hot storage addition and removal capabilities in Windows Server 2008 R2 make it possible for administrators to create dedicated backup VMs that are responsible for protecting the virtual hard drives in other VMs.

Backing Up from the Command Line

In addition to the graphical interface provided by the Windows Server Backup snap-in for Microsoft Management Console (MMC), it is possible to manage backups from the command line in two ways: by using the Wbadmin.exe program from the command prompt, and by using Windows PowerShell cmdlets. In Windows Server 2008 R2, Microsoft has updated both of these methods to reflect the new capabilities in the graphical backup management tool.

NOTE To use either of these methods, you must select Command-Line Tools when you are installing Windows Server Backup Features using the Add Features Wizard in Server Manager.

Backing Up with Wbadmin.exe

The Wbadmin.exe program in Windows Server 2008 R2 still has the same 12 commands as the previous version, but the commands have new parameters that enable you to duplicate the new capabilities in Windows Server Backup from the command prompt. For example, the Wbadmin Enable Backup command now supports 13 parameters instead of six, and the functionality of some of the existing parameters is expanded as well, as in the following examples:

- To specify a backup destination for a scheduled job, the `-addtarget` parameter now enables you to specify a disk, a volume, or a UNC path.
- The `-include` parameter, which previously could only accept volume identifiers, now accepts paths to files and folders as well, and supports the standard `*` and `?` wildcard characters.
- A new `-exclude` parameter enables you to define exclusions.

TIP For a complete list of the Wbadmin.exe commands, type `Wbadmin -?` from the command prompt. For a complete list of the parameters supported by each command, type `Wbadmin command -?` from the command prompt. When using Wbadmin.exe to manage your backups, you should always work from an elevated command prompt.

Backing Up with Windows PowerShell

As with many other areas of the operating system, Windows Server 2008 R2 includes expanded Windows PowerShell support for the Windows Server Backup program. There are more than a dozen new cmdlets for managing backups, but this functionality is integrated into a Windows PowerShell snap-in that you must load before you can use them.

To load the snap-in containing the Windows Server Backup cmdlets, run the following command from an elevated Windows PowerShell prompt:

```
add-psnapin windows.serverbackup
```

TIP To open a Windows PowerShell window with all of the available modules loaded, you can right-click the Windows PowerShell icon on the Taskbar and select **Import System Modules** from the shortcut menu.

Once you have loaded the snap-in, you can use the following command to display all of the Windows Server backup cmdlets:

```
get-command *wb* -commandtype cmdlet
```

There are now 30 backup cmdlets, as opposed to 15 in Windows Server 2008. The new backup cmdlets in Windows Server 2008 R2 are as follows:

- **Add-WBBareMetalRecovery** Adds the System State, the system drive, and other items needed to perform a Bare Metal Recovery of the server
- **Add-WBFileSpec** Specifies the files, folders, or volumes to include in or exclude from a backup
- **Add-WBSystemState** Adds the System State item to a backup
- **Get-WBBackupSet** Displays a list of the backup jobs created for the server
- **Get-WBBareMetalRecovery** Specifies whether the Bare Metal Recovery items have been added to a backup job
- **Get-WBFileSpec** Lists the items included in and excluded from a backup job
- **Get-WBJob** Displays the backup job that is currently running
- **Get-WBSystemState** Specifies whether the System State item has been added to a backup job
- **Get-WBVssBackupOptions** Specifies whether a backup job is a VSS copy job or a VSS full job
- **New-WBFileSpec** Creates a new list of files, folders, or volumes to be included in or excluded from a backup
- **Remove-WBBareMetalRecovery** Removes the Bare Metal Recovery item from a backup job
- **Remove-WBFileSpec** Removes a list of files, folders, or volumes to be included in or excluded from a backup
- **Remove-WBSystemState** Removes the System State item from a backup
- **Set-WBVssBackupOptions** Specifies whether a backup job should be a VSS copy job or a VSS full job
- **Start-WBBackup** Starts a one-time backup job

In addition, some of the cmdlets from Windows Server 2008 now support additional parameters. For example, the `New-WBBackupTarget` cmdlet now allows you to specify a disk, a volume, or a shared folder.

As with `Wbadmin.exe`, the new cmdlets are designed to implement the new capabilities in the Windows Server 2008 R2 version of Windows Server Backup. Using a combination of cmdlets or a script containing a series of commands, you can configure and execute a backup job entirely from the Windows PowerShell prompt.

For example, the following script contains commands that create a basic job that backs up the E: volume and the C:\Users folder to a dedicated disk and schedules it to execute.

NOTE The `Windows.ServerBackup` snap-in for Windows PowerShell uses the term “policy” to refer to a backup job.

```

$pol = New-WBPolicy # Creates a new backup policy
$tgt = New-WBBackupTarget -volumepath g: # Creates a backup target out of the G: volume
Add-WBBackupTarget -policy $pol -target $tgt # Adds the backup target to the policy
$file = New-WBFilespec -filespec c:\users\ # Creates a file path to be backed up
Add-WBFilespec -policy $pol -filespec $file # Adds the file path to the policy
$vol = Get-WBVolume E: # Gets a listing of the server's E: volume
Add-WBVolume -policy $pol -volume $vol # Adds the volume to be backed up to the policy
Add-WBSystemState -policy $pol # Adds the System State item to the policy
$sched = [datetime]"08/13/2009 21:00:00" # Specifies a date and time
Set-WBSchedule -policy $pol -schedule $sched # Adds a schedule to the policy
Set-WBPolicy -policy $pol -force # Activates the policy

```

BitLocker ToGo

BitLocker ToGo is a new feature of Windows 7 and Windows Server 2008 R2 that provides encryption for removable drives. This is an especially important feature for server backups because it ensures that your backups are protected in case they get into the hands of someone they shouldn't. We've read enough stories over the last several years of hard drives or tapes that contained sensitive data being lost or stolen that we've all become more aware of the issues and the reasons why encryption is very important for sensitive data.

Before you can use BitLocker ToGo, you need to add the BitLocker feature to Windows Server 2008 R2. From Server Manager, highlight the server and select Add Features from the Action menu to open the Add Features Wizard. Select BitLocker Drive Encryption and you'll get both the regular BitLocker that is designed for nonremovable drives and that uses a Trusted Platform Module (TPM) for encryption, and the new BitLocker ToGo that is used with removable drives.

To add the BitLocker Drive Encryption feature from the Windows PowerShell command line, use the following from an elevated Windows PowerShell command line:

```

Import-Module ServerManager

Add-WindowsFeature BitLocker

```

You manage BitLocker ToGo by double-clicking the BitLocker Drive Encryption icon in the Control Panel, shown in Figure 9-7.

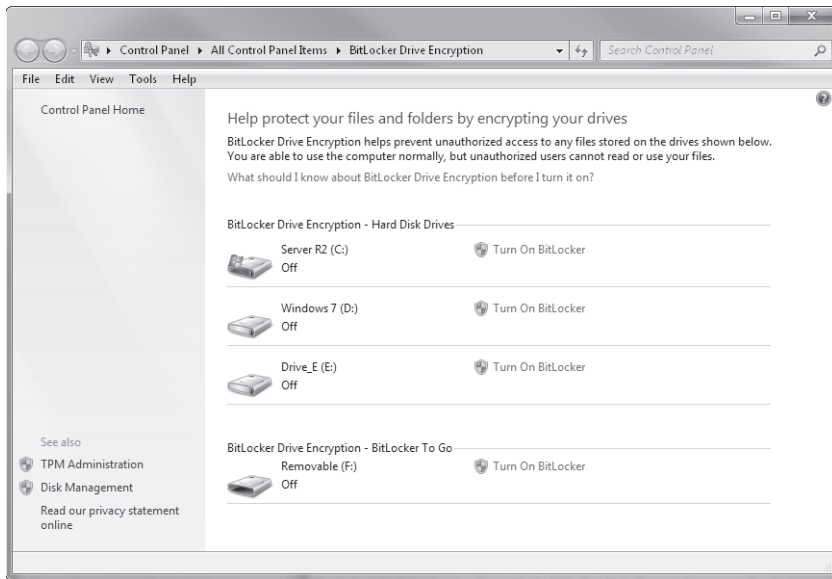


FIGURE 9-7 The BitLocker Drive Encryption Control Panel icon.

To enable BitLocker ToGo on the removable drive, click on Turn On BitLocker. If this is the first time you’ve run BitLocker or BitLocker ToGo on the server, you’ll see a message warning you that this can impact performance, as shown in Figure 9-8.



FIGURE 9-8 Warning about disk performance during encryption.

Once you click Yes, the BitLocker Drive Encryption Wizard starts, as shown in Figure 9-9. Choose how you’ll unlock the drive—using either a password or a smart card.

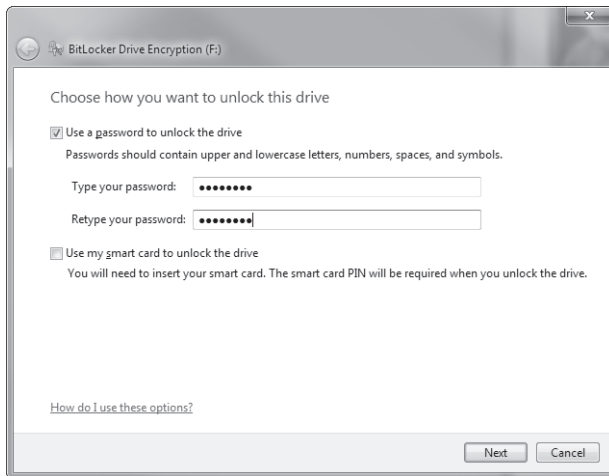


FIGURE 9-9 You can use a password or smart card to unlock the BitLocker ToGo drive.

Next you'll be offered a choice of ways to save the recovery key, as shown in Figure 9-10. We prefer using every possible method. Save to a file and then put the file somewhere safe, *but accessible!* Print the recovery key out as well, and store it in a safe location.

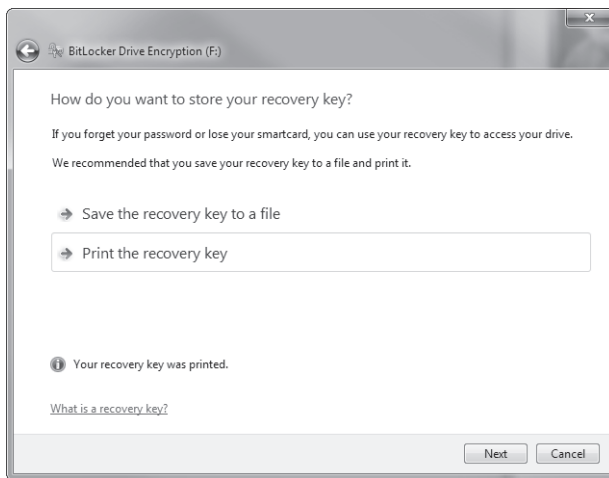


FIGURE 9-10 Store your recovery key where you can find it when you need it.

Finally, you'll have one last chance to change your mind. If you don't, click Start Encrypting to begin the process. Once encryption begins, you should not remove the drive until the process is complete. If you have to shut down the server or remove the drive, pause the encryption first. The encryption of a large drive can take a substantial amount of time, so it's best to plan this for a time when it will impact as few users as possible. Once the drive is fully encrypted, the performance penalty is quite minor, and not noticeable in normal use.

When encryption is complete, you will see a padlock icon for the drive and different options on the BitLocker Drive Encryption Control Panel applet, as shown in Figure 9-11.

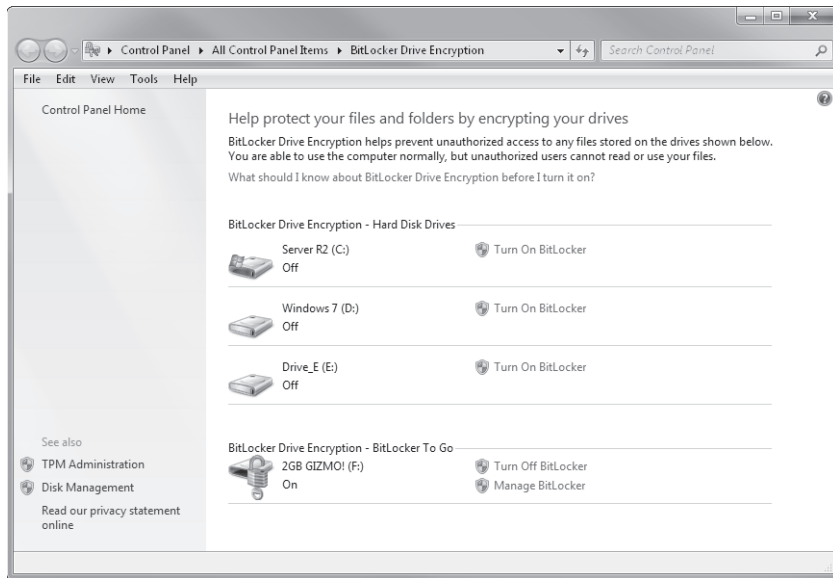


FIGURE 9-11 The F: drive has been encrypted with BitLocker ToGo.

If you click Manage BitLocker, you'll see the choices shown in Figure 9-12. You can change or remove the password, add a smart card as an unlocking mechanism, save the recovery keys, or configure the drive to automatically unlock on the current computer. This last option will mean that anyone with access to the server would not need to know the key to unlock the data on it. If they have access to your server, however, you have other problems.

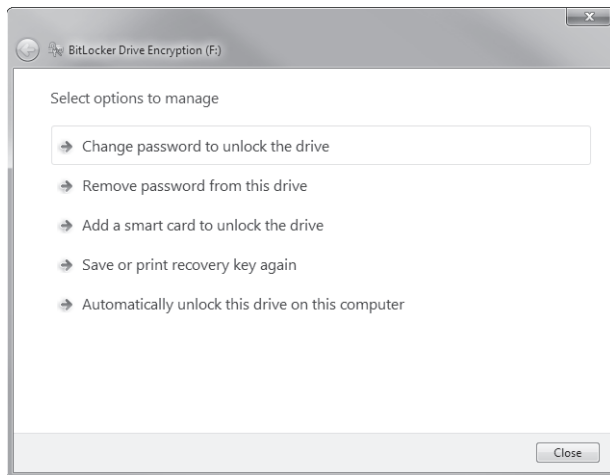


FIGURE 9-12 The Select Options To Manage dialog box.

Finally, when you plug the drive into any computer, you'll be prompted for the unlocking key, either a password or a smart card, as shown in Figure 9-13.

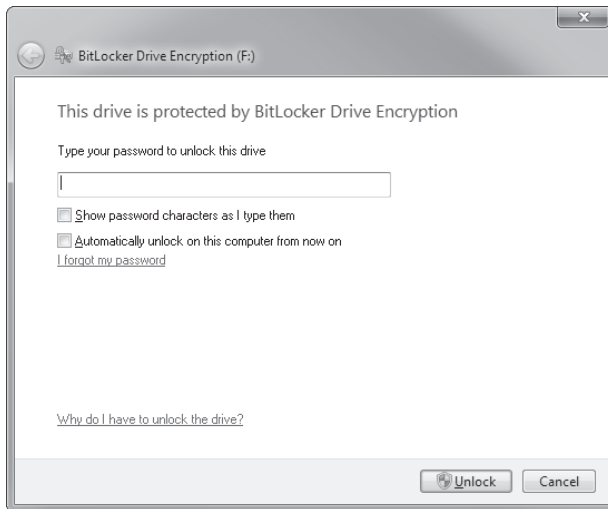


FIGURE 9-13 You can't use a BitLocker ToGo drive until it is unlocked.

Once you've unlocked the drive on a particular computer, you can configure BitLocker ToGo to always unlock on that computer without a password. This certainly simplifies managing backup drives, for example, by allowing backup drives to be managed by employees who don't have the encryption key.

BitLocker ToGo was originally designed for client systems, but we think it has a place for servers, too, especially for removable backup drives. BitLocker ToGo can be used with any drive that is recognized by Windows Server 2008 R2 as a removable drive, including USB, eSATA, and FireWire drives.

Index

Symbols and Numbers

6to4 system, 131

A

access models, 133–135

Accounting Configuration Wizard, 143

ACPI (Advanced Configuration and Power Interface), 13

Active Directory

- administration, 66, 75–77

- BPA support, 88–89

- feature improvements, 3, 65–66

- offline domain join, 86–87

- recovering deleted objects, 82

- service accounts, 87

- Windows PowerShell support, 66–67

Active Directory Administrative Center. *See* ADAC (Active Directory Administrative Center)

Active Directory Certificate Services (AD CS), 65

Active Directory Domain Services. *See* AD DS (Active Directory Domain Services)

Active Directory Domain Services Installation Wizard, 78–80

Active Directory Domains and Trusts, 78

Active Directory Federation Services (AD FS), 81

Active Directory Lightweight Directory Services. *See* AD LDS (Active Directory Lightweight Directory Services)

Active Directory Recycle Bin. *See* AD Recycle Bin

Active Directory Rights Management Services (AD RMS), 65

Active Directory Users and Computers console, 68, 70

Active Directory Web Services. *See* ADWS (Active Directory Web Services)

AD CS (Active Directory Certificate Services), 65

AD DS (Active Directory Domain Services)

- ActiveDirectory module support, 67

- ADAC support, 71

- ADWS support, 73

- Authentication Mechanism Assurance feature, 81

- BPA support, 88

- cmdlet support, 66–67

- DirectAccess support, 131, 136

AD FS (Active Directory Federation Services), 81

AD LDS (Active Directory Lightweight Directory Services)

- ActiveDirectory module support, 67

- ADWS support, 73

- cmdlet support, 66–67

- raising functional level, 84

AD Recycle Bin

- enabling, 83–84

- feature improvements, 3

- functionality, 84–85

- object recovery considerations, 82–83

- recovering deleted objects, 82, 84

AD RMS (Active Directory Rights Management Services), 65

ADAC (Active Directory Administrative Center)

- basic structure, 69

- cmdlet support, 69

- creating objects, 70–71

- customizing interface, 71–72

- feature improvements, 70

- functionality, 66, 69

- List View option, 71

- search mechanism, 72

- selecting functional levels, 79

- Tree View, 71

Add Allow Restrictions Rule dialog box, 125

Add Features Wizard

- ADAC, 76
- BitLocker to Go, 158
- DirectAccess, 136
 - opening, 20
- Add Role Services Wizard, 20
- Add Roles Wizard
 - Hyper-V roles, 28
 - Microsoft .NET Framework, 65, 73
 - opening, 20
 - Web Server (IIS) role, 109
- Add-WBBareMetalRecovery cmdlet, 157
- Add-WBFileSpec cmdlet, 157
- Add-WBSystemState cmdlet, 157
- Add-WebConfiguration cmdlet, 120
- Add-Windowsfeature cmdlet, 77
- administration
 - Active Directory, 66, 75–77
 - changing lifetime values, 83
 - RemoteApp and Desktop Connection, 49–53
- Administrator account, 16
- adprep command, 83
- Advanced Configuration and Power Interface (ACPI), 13
- ADWS (Active Directory Web Services)
 - cmdlet considerations, 74
 - communication considerations, 74–75
 - functionality, 65, 73
 - installing, 73
 - remote management considerations, 74
- aggregation, defined, 94
- agile VPN, 140
- AH (Authenticated Header) protocol, 131, 140
- anonymous authentication, 118
- APIs (application programming interfaces), 33, 54
- Appcmd.exe program, 123, 126
- Application Pool Identity, 118
- application programming interfaces (APIs), 33, 54
- App-V for RDS, 63
- ASP.NET applications, 116, 122
- audio input/recording, 54
- audio/video synchronization, 54
- Authenticated Header (AH) protocol, 131, 140
- authentication
 - anonymous, 118
 - DirectAccess support, 130–131
- Authentication Mechanism Assurance feature, 81
- authorization, 130

B

- Background Intelligent Transfer Service (BITS), 33, 102, 104
- Backup Once Wizard, 147–148
- Backup Schedule Wizard, 147–148, 150–151
- backups. *See also* Windows Server Backup
 - BitLocker ToGo, 158
 - FCI considerations, 92
 - from the command line, 155–157
 - Hyper-V, 154–155
 - incremental, 152–153
 - selected files/folders, 147–148
 - selecting destination, 150–151
 - system state, 153–154
 - VM considerations, 154–155
 - with Windows PowerShell, 156–157
- Backup-WebConfiguration cmdlet, 121
- Best Practices Analyzer. *See* BPA (Best Practices Analyzer)
- BIOS, 56
- BitLocker ToGo, 158
- BITS (Background Intelligent Transfer Service), 33, 102, 104
- BPA (Best Practices Analyzer)
 - cmdlet support, 89
 - functionality, 7, 88–89
 - IIS support, 4, 127
 - Server Manager support, 20
- BranchCache
 - communication considerations, 102–104
 - configuring clients, 106–107
 - configuring file shares, 105
 - configuring servers, 104–105
 - Distributed Cache Mode, 101–102, 106
 - feature improvements, 3
 - functionality, 101
 - Hash Publication for BranchCache setting, 105
 - Hosted Cache Mode, 101–102, 106–107
 - operational modes, 101–102
- BranchCache discovery protocol, 102
- BranchCache hosted cache protocol, 103
- BranchCache retrieval protocol, 102–103

C

CA (certification authority), 107
 caching, defined, 101
 CALs (Client Access Licenses)
 feature improvements, 2
 licensing considerations, 60
 name equivalent, 48
 cd sites command, 119
 certificate revocation list (CRL), 136
 certificates, exporting, 56
 certification authority (CA), 107
 classification properties, 93–95
 Classification Rule Definitions dialog box, 96
 classification rules, 96–99
 Clear-WebConfiguration cmdlet, 121
 Client Access Licenses. *See* CALs (Client Access Licenses)
 CLR (Common Language Runtime), 116
 Cluster Shared Volume. *See* CSV (Cluster Shared Volume)
 clustering
 system requirements, 13
 VM considerations, 28
 cmdlets
 Active Directory support, 66–69
 ADAC support, 69
 ADWS considerations, 74
 backup support, 156–157
 BPA support, 89
 failover clusters, 31–32
 IIS support, 118–121
 listing, 67
 low-level configuration, 120–121
 parameter considerations, 67
 RemoteDesktopServices module, 51
 task-oriented, 121
 Codeplex project, 31–32
 command line, backing up from, 155–157
 comma-separated value (CSV), 68
 Common Language Runtime (CLR), 116
 Complete PC backup utility, 147
 configuration
 BranchCache clients, 106–107
 BranchCache servers, 104–105
 DirectAccess, 136–139
 file shares, 105
 low-level cmdlets, 120–121

NPS logging, 143
 process overview, 16–17
 RD Virtualization Host, 55
 remote applications, 58
 RemoteApp and Desktop Connection, 55
 role-based, 19
 Server Core, 21–23
 virtual machine settings, 30–31
 VM for live migration, 38–45
 VM settings, 30–31
 Configuration Editor, 122–123
 configuration tracing, 126
 ConnectionBroker container, 51
 core parking, 13
 Create an Application Package Wizard, 112
 CRL (certificate revocation list), 136
 CSV (Cluster Shared Volume)
 adding cluster storage, 40–43
 enabling, 40
 failover clustering, 35
 functionality, 13
 CSV (comma-separated value), 68

D

Data Execution Protection, 56
 Date-Time property, 94
 Dcpromo.exe program, 78
 default password policy, 17
 deleted object lifetime, 83
 deleted objects
 garbage collection and, 83
 logically deleted, 83
 recovering, 82
 restoring, 84
 Deleted Objects container, 82
 deploying
 DirectAccess, 133–139
 virtual machines, 27–28
 DFS (Distributed File System), 108
 DHCP (Dynamic Host Configuration Protocol), 142
 digitally signed drivers, 16
 dir command, 51, 119
 DirectAccess
 benefits, 130
 choosing access models, 133–135

- client requirements, 135
- configuring, 136–139
- connection process, 132–133
- deploying, 133–139
- functionality, 5, 129–130
- infrastructure requirements, 136
- IPsec support, 131–132
- IPv6 support, 131
- server requirements, 135
- VPN comparison, 130
- DirectAccess Application Server Setup Wizard, 139
- DirectAccess Client Setup Wizard, 137
- DirectAccess Infrastructure Server Setup Wizard, 138
- DirectAccess Server Setup Wizard, 138
- Directory Services Restore mode, 83
- DirectX redirection, 54
- Dism.exe program. *See* Server Core
- Distributed File System (DFS), 108
- Djoin.exe program, 86–87
- DNS (Domain Name System), 131
- domain controllers
 - Authentication Mechanism Assurance feature, 81
 - functional levels and, 78, 80
- domain functional levels, 80–81
- Domain Name System (DNS), 131
- domains, preparing, 16
- drivers, digitally signed, 16
- Dsadd.exe program, 66
- Dsget.exe program, 66
- Dsmmod.exe program, 66
- Dynamic Host Configuration Protocol (DHCP), 142

E

- Enable-ADOptionalFeature cmdlet, 84
- Encapsulating Security Payload (ESP) protocol, 131, 140
- encryption
 - BitLocker To Go, 158
 - DirectAccess support, 130
 - FCI considerations, 92
- Enhanced Page Tables, 12
- Enterprise Remote Access. *See* RemoteApp and Desktop Connection
- eSATA drives, manipulating, 35
- ESP (Encapsulating Security Payload) protocol, 131, 140
- Event Viewer console, 126

- exclusions, defined, 148
- exporting
 - SSL certificates, 56
 - virtual machines, 28
- Extensible Markup Language (XML), 74, 112

F

- Failover Cluster Manager
 - creating VMs, 43–45
 - Hyper-V considerations, 27
 - Validate a Configuration Wizard, 39
- failover clusters
 - cmdlet support, 31–32
 - creating, 39
 - CSV support, 35
 - DFS support, 108
 - VM considerations, 38
 - Windows PowerShell support, 31–32, 43–45
- FastCGI, 117–118, 122
- FCI (File Classification Infrastructure)
 - components supported, 92–93
 - creating classification properties, 93–95
 - creating classification rules, 96–99
 - file management tasks, 99–101
 - functionality, 3, 91–92
- features
 - adding, 20, 136
 - role-based configuration, 19
- File Classification Infrastructure. *See* FCI (File Classification Infrastructure)
- File Server Resource Manager. *See* FSRM (File Server Resource Manager)
- File Services role, 91
- file shares, configuring, 105
- File Transfer Protocol (FTP), 4, 114
- files
 - backing up, 147–148
 - managing, 99–101
- FileSystem provider, 51
- firewalls, 136
- folders, backing up, 147–148
- Forefront IAG, 140
- forest functional levels, 79–80, 84
- forests, preparing, 15–16
- FQDN (fully qualified domain name), 106

FSRM (File Server Resource Manager)

- Classification Rules node, 96
- depicted, 93
- File Management Tasks node, 99

FTP (File Transfer Protocol), 4, 114

FTP Server role service

- downloading, 110
- functionality, 110, 114–115

fully qualified domain name (FQDN), 106

functional levels

- defined, 78
- domain, 80–81
- forest, 79–80
- raising, 79
- selecting, 78–79
- setting, 78–79

G

garbage collection, 83

GatewayServer container, 51

Get-ADObject cmdlet, 69, 84–85

Get-BPAModel cmdlet, 89

Get-BPAResult cmdlet, 89

Get-ChildItem cmdlet, 51

Get-Command cmdlet, 67, 119

Get-Help cmdlet, 52, 67

Get-Item cmdlet, 119–120

Get-WBBackupSet cmdlet, 157

Get-WBBareMetalRecovery cmdlet, 157

Get-WBFileSpec cmdlet, 157

Get-WBJob cmdlet, 157

Get-WBSystemState cmdlet, 157

Get-WBVssBackupOptions cmdlet, 157

Get-WebConfiguration cmdlet, 121

Get-Website cmdlet, 121

GPU (graphical processing unit), 54

graphical processing unit (GPU), 54

Group Policy

- BranchCache support, 104–106
- configuring remote applications, 58
- DirectAccess support, 136, 139
- P-state management, 13

H

hardware virtualization, 56

health verification, 130, 142

High Availability Wizard, 45

host-compatibility checks, 33

HTML (Hypertext Markup Language), 121

HTTP (Hypertext Transfer Protocol), 102, 104, 113

Hypertext Markup Language (HTML), 121

Hypertext Transfer Protocol (HTTP), 102, 104, 113

Hyper-V

backing up, 154–155

CSV support, 35

enabling role, 56

feature improvements, 3–4, 25

licensing considerations, 26

live migration, 37

managing VM storage, 35

optimizing performance, 45–46

Processor Compatibility settings, 33

Quick Migration support, 37

SLAT support, 12, 45

Windows Server Backup support, 147

Hyper-V Manager console

configuring VM settings, 30–31

creating virtual machines, 28–30

functionality, 27–28

I

IAG (Intelligent Application Gateway), 140

IAS, migrating settings, 146

Iasmigreader.exe program, 146

ICMPv6, 136

IETF (Internet Engineering Task Force), 114

IIS (Internet Information Services)

accessing resources on Internet, 128

ASP.NET applications, 116

auto-start feature, 116

BPA support, 4, 127

cmdlet support, 118–121

Configuration Editor, 122–123

configuration tracing, 126

creating IP address restrictions, 125

FastCGI support, 117–118

feature improvements, 4, 109, 113–115

- hosting applications, 116–118
- installing, 109–113
- managing, 118–128
- performance counters, 128
- Request Filtering module, 124–125
- Windows PowerShell, 4, 6, 116, 118–121
- IIS Administration Pack, 122–125
- IIS Hostable Web Core role service, 110
- IIS Manager, 117, 122, 124
- IIS PowerShell provider, 119–120
- IKv2 (Internet Key Exchange version 2), 140
- Import-CSV cmdlet, 68
- Import-Module cmdlet, 67, 77
- incremental backups, 152–153
- Initial Configuration Tasks Wizard, 17, 20
- installation
 - configuration step, 16–20
 - IIS, 109–113
 - license keys and, 16
 - process overview, 16
 - Remote Server Administration Tools, 77
 - schema updates, 15–16
 - supported upgrade scenarios, 14–15
- Intelligent Application Gateway (IAG), 140
- Internet Engineering Task Force (IETF), 114
- Internet Information Services. *See* IIS (Internet Information Services)
- Internet Key Exchange version 2 (IKv2), 140
- Internet Protocol version 6. *See* IPv6 (Internet Protocol version 6)
- Intra-Site Automatic Tunnel Addressing Protocol (ISATAP), 131
- Invoke-BPAModel cmdlet, 89
- IP addresses
 - configuring Server Core, 21
 - creating restrictions, 125
 - virtual, 52
- IP-HTTPS, 131
- IPsec, 131–132
- IPv6 (Internet Protocol version 6)
 - DirectAccess support, 131, 136
 - IIS support, 4, 125
 - transition technologies, 131, 136
- ISATAP (Intra-Site Automatic Tunnel Addressing Protocol), 131
- iSCSI
 - manipulating drives, 35

- NIC considerations, 28
- VM backups, 155
- IUSR account, 118

J

- Jumbo Frames, 46

L

- language settings, 54
- LDAP (Lightweight Directory Access Protocol)
 - functionality, 72–73
 - Ldp.exe client, 79, 83
 - Ldp.exe program, 79, 83
 - license keys, 16
 - LicenseServer container, 51
- licensing
 - Hyper-V considerations, 26
 - RDS considerations, 60–63
 - support changes, 2
 - Volume License, 62
- lifetime values, changing, 83
- Lightweight Directory Access Protocol. *See* LDAP (Lightweight Directory Access Protocol)
- live migrations
 - benefits, 37
 - configuring VM, 38–45
 - host-compatibility checks, 33
 - Hyper-V considerations, 37
 - optimizing VM performance, 45–46
 - queuing, 33
 - Quick Migration comparison, 37
- logging, NPS, 143
- logically deleted objects, 83

M

- managed service accounts, 87, 118
- management
 - ADWS considerations, 74
 - feature improvements, 3
 - file, 99–101
 - IIS, 118–128

- RemoteApp and Desktop Connection, 49–53
- Server Core, 23–24
- virtual machines, 27–28
- MDOP (Microsoft Desktop Optimization Pack), 63
- memory
 - SLAT considerations, 12
 - system requirements, 12
- metadata, 102
- Microsoft .NET Framework
 - ADWS requirements, 73
 - Server Core support, 4, 116
- Microsoft Desktop Optimization Pack (MDOP), 63
- Microsoft Management Console (MMC), 24, 107
- Microsoft Urlscan Filter, 124
- Microsoft Web Platform Installer, 110–111
- migrations. *See* live migrations
- MMC (Microsoft Management Console), 24, 107
- MOBIKE, 140
- modules, defined, 67
- Move-ADObject cmdlet, 69
- MS Deploy tool, 111
- multimedia redirection, 54
- multimonitor support, 54
- Multiple Choice List property, 94–95
- Multistring property, 94

N

- NAP (Network Access Protection)
 - DirectAccess support, 5, 130
 - NPS support, 142
- NAT-PT, 131
- navigation nodes, 71
- Netdom command, 22
- Network Access Protection (NAP)
 - DirectAccess support, 5, 130
 - NPS support, 142
- network adapters, 31
- network interface card (NIC), 28
- Network Load Balancing (NLB), 4, 14
- Network Policy Server. *See* NPS (Network Policy Server)
- Network Service account, 118
- networking
 - feature improvements, 5
 - VM improvements, 46
- New RADIUS Clients dialog box, 144

- New RADIUS Shared Secret Template dialog box, 144
- New Virtual Machine Wizard, 29–30, 45
- New-ADComputer cmdlet, 67
- New-ADGroup cmdlet, 67
- New-ADObject cmdlet, 67–68
- New-ADOrganizationalUnit cmdlet, 67
- New-ADUser cmdlet, 67–68
- New-WBFileSpec cmdlet, 157
- New-Website cmdlet, 121–122
- NIC (network interface card), 28
- NLB (Network Load Balancing), 4, 14
- NPS (Network Policy Server)
 - configuring logging, 143
 - DirectAccess support, 130
 - feature improvements, 142
 - migrating IAS configuration settings, 146
 - template support, 144–146
- Number property, 94

O

- objects, creating, 70
- Ocsetup.exe program. *See* Server Core
- offline domain join, 86–87
- Oobe (Out of Box Experience), 17
- Optimize Backup Performance dialog box, 152–153
- Ordered List property, 94–95
- Out of Box Experience (OOBE), 17

P

- passwords
 - default policy, 17
 - FTP considerations, 114
 - setting for Administrator account, 16
- performance counters, 128
- Performance Monitor console, 128
- permissions
 - Application Pool Identity, 118
 - FCI considerations, 92
- PHP scripting language, 117
- PKI (Public Key Infrastructure), 136
- power consumption, 13
- Processor Compatibility Mode, 38

- processors
 - core parking, 13
 - system requirements, 12
- Properties dialog box, 141
- providers, defined, 51
- PSHyperV, 32
- P-state management, 13
- Public Key Infrastructure (PKI), 136

Q

- queries, building, 72
- queuing live migrations, 33

R

- RAD. *See* RemoteApp and Desktop Connection
- RAD Control Panel, 54–55
- RADIUS (Remote Authentication Dial-In User Service), 142, 144
- RADIUS proxy, 142
- RAID (Redundant Array of Independent Disks), 91
- Rapid Virtualization Indexing (RVI), 12
- RD CALs, 48, 60–61
- RD Connection Broker
 - connecting to, 55
 - enabling, 56
 - licensing considerations, 62
 - name equivalent, 48
 - setting up, 55
- RD Gateway, 59
- RD Licensing Manager, 61
- RD Session Host
 - CAL considerations, 61
 - enabling, 56
 - licensing considerations, 61–62
 - name equivalent, 48
 - setting up, 55
- RD Virtualization Host
 - configuring, 55
 - functionality, 47
 - licensing considerations, 62
- RD Web Access
 - enabling, 56
 - functionality, 59–60
 - name equivalent, 48
- RemoteApp and Desktop Connection, 49–50
 - setting up, 55
 - SSL certificates, 56
- RDP (Remote Desktop Protocol), 54
- RDS (Remote Desktop Services)
 - App-V for RDS, 63
 - as TS replacement, 2, 47–48
 - DirectX redirection, 54
 - feature improvements, 2, 47, 54
 - functionality, 48–49
 - licensing considerations, 60–63
 - multimedia redirection, 54
 - multimonitor support, 54
 - Windows 7 considerations, 5, 54–55
 - Windows PowerShell support, 48, 51–53
- RDSConfiguration container, 51
- RDSFarms container, 51
- recovering deleted objects, 82
- Recycle Bin. *See* AD Recycle Bin
- recycled object, 83
- recycled object lifetime, 83
- Redundant Array of Independent Disks (RAID), 91
- Re-Image Your Computer Wizard, 154
- remote access
 - feature improvement, 5
 - managing Server Core, 23
- Remote Agent Service, 113
- Remote Authentication Dial-In User Service (RADIUS), 142, 144
- Remote Desktop Connection Manager console, 55
- Remote Desktop Easy Print, 48
- Remote Desktop Gateway, 48
- Remote Desktop Protocol (RDP), 54
- Remote Desktop Services. *See* RDS (Remote Desktop Services)
- Remote Procedure Calls (RPC), 73
- Remote Server Administration Tools
 - installing, 77
 - installing on Windows 7, 77
 - installing with Servercmd.exe program, 77
- RemoteApp, 48, 58–59
- RemoteApp and Desktop Connection
 - administration, 49–53
 - configuring, 55
 - Control Panel link, 54–55
 - depicted, 49–50

- functionality, 48
- management, 49–53
- RemoteApp container, 51
- RemoteApp Wizard, 58
- Remove-ADObject cmdlet, 69
- Remove-WBBareMetalRecovery cmdlet, 157
- Remove-WBFileSpec cmdlet, 157
- Remove-WBSystemState cmdlet, 157
- Remove-Website cmdlet, 121
- Rename-ADObject cmdlet, 69
- Repair Your Computer in the Windows Setup Wizard, 153
- request filtering, 124–125
- Restore-ADObject cmdlet, 69, 84
- Restore-WebConfiguration cmdlet, 121
- RFC 4918, 114
- role services
 - adding, 20
 - IIS support, 109
 - licensing considerations, 60
 - role-based configuration, 19
- role-based configuration, 19
- roles
 - adding, 20
 - licensing considerations, 60
 - role-based configuration, 19
- Routing and Remote Access Services (RRAS), 141
- routing, split-tunnel, 130
- RPC (Remote Procedure Calls), 73
- RRAS (Routing and Remote Access Services), 141
- RS provider, 51–52
- RVI (Rapid Virtualization Indexing), 12

S

- SANs (storage area networks), 3, 13
- scalability
 - feature improvements, 4, 13
 - VM improvements, 45
- SCCM (System Center Configuration Manager), 63
- schema updates
 - preparing domains, 16
 - preparing forests, 15–16
- SCOM (System Center Operations Manager), 62–63
- Sconfig.exe program. *See* Server Configuration utility
- SCVMM (System Center Virtual Machine Manager)
 - cmdlet support, 32
 - creating virtual machines, 30
 - depicted, 27
 - feature improvements, 33
 - host-compatibility checks, 33
 - licensing considerations, 63
 - queuing live migrations, 33
 - rapid provisioning, 33
 - storage migration, 33
 - third-party storage support, 34
 - VDI support, 62
 - VMware vMotion, 33
 - Windows PowerShell support, 34
- Second Level Address Translation. *See* SLAT (Second Level Address Translation)
- Secure Sockets Layer. *See* SSL (Secure Sockets Layer)
- Select Items dialog box, 148
- Select Items To Exclude dialog box,, 148
- Select-Object cmdlet, 119–120
- Select-WebConfiguration cmdlet, 121
- Server Configuration utility, 23
- server consolidation, 13
- Server Core
 - configuring, 21–23
 - functionality, 9–10
 - installation considerations, 21
 - managing, 23–24
 - Microsoft .NET Framework support, 4, 116
 - Windows PowerShell support, 3
- Server Manager
 - adding features, 20, 136
 - adding role services, 20
 - adding roles, 20
 - BranchCache support, 104
 - functionality, 19
 - Hyper-V support, 27
 - managing Server Core, 24
 - triggering BPA scan, 88
 - Web Server (IIS) node, 127
 - Windows PowerShell support, 9, 20, 77
- Server Message Blocks (SMB), 102, 104
- Servercmd.exe program, 77
- service accounts, 87, 118
- Set-ADDomainMode cmdlet, 78
- Set-ADForestMode cmdlet, 78

Set-ADObject cmdlet

- Set-ADObject cmdlet, 69, 83
- Set-BPAResult cmdlet, 89
- Set-ExecutionPolicy cmdlet, 119
- Set-WBVssBackupOptions cmdlet, 157
- Set-WebConfiguration cmdlet, 121
- shutdown -r command, 22
- SLAT (Second Level Address Translation)
 - functionality, 12
 - Hyper-V support, 4, 45
- SMB (Server Message Blocks), 102, 104
- snapshots of virtual machines, 28
- SOAP, 74
- split-tunnel routing, 130
- SQL databases, 143
- SSL (Secure Sockets Layer)
 - exporting certificates, 56
 - FTP support, 114
 - HTTP support, 114
 - IIS support, 4
 - IP-HTTPS support, 131
 - VPN support, 140
- Start-WBBackup cmdlet, 157
- Start-Website cmdlet, 121
- Stop-Website cmdlet, 121
- storage area networks (SANs), 3, 13
- storage management
 - factors to consider, 92
 - feature improvements, 3
 - migrating for VM, 33
 - third-party support, 34
 - virtual machine support, 35
- String property, 94
- synchronization, audio/video, 54
- System Center Configuration Manager (SCCM), 63
- System Center Operations Manager (SCOM), 62–63
- System Center Virtual Machine Manager. *See* SCVMM (System Center Virtual Machine Manager)
- System Recovery Options dialog box, 153
- system requirements
 - clustering, 13
 - digitally signed drivers, 16
 - minimum, 11
 - power consumption, 13
 - processors and memory, 12
 - scalability, 13
- system state, backing up, 153–154

T

- Task Scheduler, 54
- TCP (Transmission Control Protocol), 107
- TCP Offload, 46
- templates, NPS, 144–146
- Teredo, 131
- Terminal Server, 48
- Terminal Services. *See* TS (Terminal Services)
- text files, 143
- third-party storage, 34
- tombstone objects, 82, 84
- tombstone reanimation, 82
- TPM (Trusted Platform Module), 158
- transition technologies, 131, 136
- Transmission Control Protocol (TCP), 107
- Trusted Platform Module (TPM), 158
- TS (Terminal Services)
 - license server assignment, 61
 - managing Server Core, 23
 - name equivalent, 48
 - RDS as replacement, 2, 47
- TS CALs, 48, 61
- TS Easy Print, 48
- TS Gateway, 48
- TS RemoteApp
 - functionality, 47–48
 - name equivalent, 48
- TS Session Broker, 48
- TS Web Access, 48

U

- UDP (User Datagram Protocol), 107
- UNC (Universal Naming Convention), 151
- Unicode characters, 4
- Uniform Resource Locators (URLs), 114
- Universal Naming Convention (UNC), 151
- upgrades, top reasons, 5–6
- URLs (Uniform Resource Locators), 114
- USB drives, manipulating, 35
- User Datagram Protocol (UDP), 107

V

Validate a Configuration Wizard, 39

VDI (Virtual Desktop Infrastructure)

- audio input/recording, 54
- enabling, 55–58
- feature improvements, 2
- licensing considerations, 62–63
- RAD Control panel applet, 54–55
- Windows Aero support, 54

VDI Premium Suite, 63

VDI Standard Suite, 63

VECD (Virtual Enterprise Centralized Desktop), 62

VECD for SA, 62

Veritas Volume Manager, 34

VHDs (virtual hard disks)

- BITS considerations, 33
- recommendations, 30
- Windows Server Backup support, 147

Virtual Desktop Infrastructure. *See* VDI (Virtual Desktop Infrastructure)

virtual desktops

- RD Virtualization Host, 47
- RD Web Access, 49, 60
- VDI support, 55

Virtual Enterprise Centralized Desktop (VECD), 62

virtual hard disks (VHDs)

- BITS considerations, 33
- recommendations, 30
- Windows Server Backup support, 147

Virtual Machine Manager console, 34

Virtual Machine Queue (VMQ), 46

virtual machines. *See* VMs (virtual machines)

Virtual PC, 147

virtual private network. *See* VPN (virtual private network)

virtualization. *See also* Hyper-V

- feature improvements, 3
- hardware, 56
- strategic role, 25–26

VM Chimney, 46

VMQ (Virtual Machine Queue), 46

VMs (virtual machines)

- adding, 28
- backup considerations, 154
- changing settings, 28

- configuring for live migration, 38–45
- configuring settings, 30–31
- creating, 28–30, 43–45, 56
- deleting, 28
- deploying and managing, 27–28
- exporting, 28
- managing storage, 35
- migration considerations, 37
- networking improvements, 46
- optimizing performance, 45–46
- Processor Compatibility Mode, 38
- running internal backups, 155
- scalability improvements, 45
- SCVMM support, 33–34
- taking snapshots, 28

VMware vMotion, 33

Voice over Internet Protocol (VoIP), 54

VoIP (Voice over Internet Protocol), 54

Volume License, 62

Volume Shadow Copy Service (VSS), 155

VPN (virtual private network)

- defined, 129
- DirectAccess comparison, 130
- reconnecting, 140–141
- SSL support, 140

VPN Reconnect, 140–141

VSS (Volume Shadow Copy Service), 155

VSS Writer, 155

W

W3wp.exe worker process, 118

Wbadmin.exe program, 156

WCF (Windows Communication Foundation), 74

Web Deployment Tool, 111–113

web feature improvements, 4

WebDAV Publishing role service

- downloading, 110
- functionality, 109, 113–114

Windows 7

- installing Remote Server Administration Tools, 77
- R2 improvements and, 5
- RDS support, 50, 54–55

Windows Aero, 54

Windows Communication Foundation (WCF), 74

Windows Installer, 58

Windows PowerShell

- Active Directory support, 66–67

- adding role features, 20

- adding role services, 20

- adding roles, 20

- backing up with, 156–157

- cmdlet support, 31–32

- FailoverClusters module, 31–32, 43–45

- feature improvements, 3, 8

- FileSystem provider, 51

- IIS support, 4, 6, 116, 118–121

- importing ActiveDirectory module, 67

- launching, 67

- managing Server Core, 24

- providers, 51

- raising functional levels, 79

- rapid provisioning, 33

- RemoteDesktopServices module, 48, 51–53

- RS provider, 51–52

- SCVMM support, 34

- ServerManager module, 9, 20, 77

- WebAdministration module, 118

Windows PowerShell Management Library, 31

Windows Registry, 51

Windows Remote Shell, 24

Windows Server 2008

- licensing and packaging changes, 2

- R2 improvements, 2–5

- reasons to upgrade, 5–6

- release cadence, 1

Windows Server 2008 R2 Foundation, 2

Windows Server Backup

- backing up from command line, 155–157

- backing up Hyper-V, 154–155

- backing up system state, 153–154

- Backup Once Wizard, 147–148

- Backup Schedule Wizard, 147–148, 150–151

- functionality, 147

- incremental backups, 152–153

- Shadow Copies for Shared Folders feature, 147–148

- shortcomings, 147

Windows Server Core. *See* Server Core

WindowsImageBackup volume, 150

WS-Discovery protocol, 107

WS-Enumeration protocol, 74

WS-Transfer protocol, 74

X

XML (Extensible Markup Language), 74, 112

Y

Yes/No property, 94

About the Authors

CHARLIE RUSSEL is a chemist by education, an electrician by trade, a UNIX sysadmin and Oracle DBA because he raised his hand when he should have known better, an IT director and consultant by default, and a writer by choice. Charlie is a Microsoft MVP for Windows Server and is the author of more than two dozen computer books on operating systems and enterprise environments, including *Microsoft Windows Small Business Server 2008 Administrator's Companion*, *Windows Server 2008 Administrator's Companion*, and (with Robert Cordingley) the *Oracle DBA Quick Reference Series*. He has also written numerous white papers on Microsoft.com. Charlie lives in beautiful British Columbia with one dog, varying numbers of cats, and a delightful, if somewhat distracting, view of Pender Harbour.

CRAIG ZACKER is a writer, editor, and educator whose computing experience began in the days of teletypes and paper tape. After making the move from minicomputers to PCs, he worked as a network administrator and PC support technician while operating a freelance desktop publishing business. After earning a Master's Degree in English and American Literature from New York University, Craig worked extensively on the integration of Microsoft Windows operating systems into existing internetworks, supported fleets of Windows workstations, and was employed as a technical writer, content provider, and webmaster for the online services group of a large software company. Since devoting himself to writing and editing full-time, Craig has authored or contributed to dozens of books on operating systems, networking topics, and PC hardware. He has also developed educational texts for college courses, developed online training courses for the Web, and published articles with top industry publications. His latest book is *Windows Small Business Server 2008 Administrator's Pocket Consultant*, published by Microsoft Press.

Get Certified—Windows Server 2008

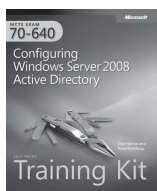
Ace your preparation for the skills measured by the Microsoft® certification exams—and on the job. With 2-in-1 *Self-Paced Training Kits*, you get an official exam-prep guide + practice tests. Work at your own pace through lessons and real-world case scenarios that cover the exam objectives. Then, assess your skills using practice tests with multiple testing modes—and get a customized learning plan based on your results.



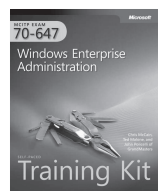
EXAMS 70-640, 70-642, 70-646
MCITP Self-Paced Training Kit: Windows Server® 2008 Server Administrator Core Requirements
ISBN 9780735625082



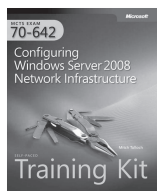
EXAMS 70-640, 70-642, 70-643, 70-647
MCITP Self-Paced Training Kit: Windows Server 2008 Enterprise Administrator Core Requirements
ISBN 9780735625723



EXAM 70-640
MCTS Self-Paced Training Kit: Configuring Windows Server 2008 Active Directory®
Dan Holme, Nelson Ruest, and Danielle Ruest
ISBN 9780735625136



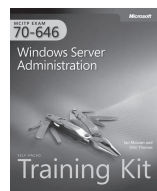
EXAM 70-647
MCITP Self-Paced Training Kit: Windows® Enterprise Administration
Orin Thomas, et al.
ISBN 9780735625099



EXAM 70-642
MCTS Self-Paced Training Kit: Configuring Windows Server 2008 Network Infrastructure
Tony Northrup, J.C. Mackin
ISBN 9780735625129



EXAM 70-643
MCTS Self-Paced Training Kit: Configuring Windows Server 2008 Applications Infrastructure
J.C. Mackin, Anil Desai
ISBN 9780735625112



EXAM 70-646
MCITP Self-Paced Training Kit: Windows Server Administration
Ian McLean, Orin Thomas
ISBN 9780735625105

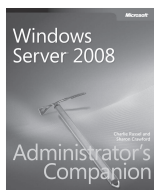
ALSO SEE

Windows Server 2008 Administrator's Pocket Consultant
William R. Stanek
ISBN 9780735624375

Windows Server 2008 Administrator's Companion
Charlie Russel, Sharon Crawford
ISBN 9780735625051

Windows Server 2008 Resource Kit
Microsoft MVPs with Windows Server Team
ISBN 9780735623613

Windows Server 2008— Resources for Administrators



Windows Server® 2008 Administrator's Companion

Charlie Russel and Sharon Crawford
ISBN 9780735625051

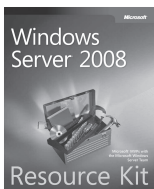
Your comprehensive, one-volume guide to deployment, administration, and support. Delve into core system capabilities and administration topics, including Active Directory®, security issues, disaster planning/recovery, interoperability, IIS 7.0, virtualization, clustering, and performance tuning.



Windows Server 2008 Administrator's Pocket Consultant

William R. Stanek
ISBN 9780735624375

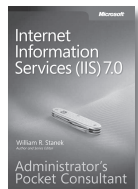
Portable and precise—with the focused information you need for administering server roles, Active Directory, user/group accounts, rights and permissions, file-system management, TCP/IP, DHCP, DNS, printers, network performance, backup, and restoration.



Windows Server 2008 Resource Kit

Microsoft MVPs with Microsoft
Windows Server Team
ISBN 9780735623613

Six volumes! Your definitive resource for deployment and operations—from the experts who know the technology best. Get in-depth technical information on Active Directory, Windows PowerShell™ scripting, advanced administration, networking and network access protection, security administration, IIS, and more—plus an essential toolkit of resources on CD.



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

William R. Stanek
ISBN 9780735623644

This pocket-sized guide delivers immediate answers for administering IIS 7.0. Topics include customizing installation; configuration and XML schema; application management; user access and security; Web sites, directories, and content; and performance, backup, and recovery.



Windows PowerShell Step by Step

Ed Wilson
ISBN 9780735623958

Teach yourself the fundamentals of the Windows PowerShell command-line interface and scripting language—one step at a time. Learn to use *cmdlets* and write scripts to manage users, groups, and computers; configure network components; administer Microsoft® Exchange Server 2007; and more. Includes 100+ sample scripts.

ALSO SEE

Windows Server 2008 Hyper-V™ Resource Kit

ISBN 9780735625174

Internet Information Services (IIS) 7.0 Resource Kit

ISBN 9780735624412

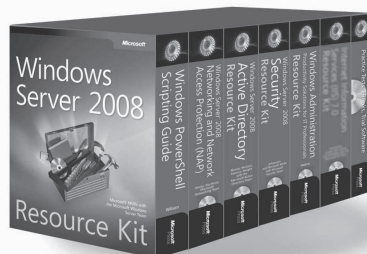
Windows® Administration Resource Kit: Productivity Solutions for IT Professionals

ISBN 9780735624313

Windows Server 2008 Security Resource Kit

ISBN 9780735625044

Windows Server 2008 Resource Kit— Your Definitive Resource!

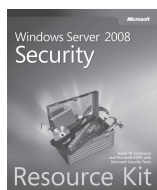


Windows Server® 2008 Resource Kit

Microsoft® MVPs with
Microsoft Windows Server Team
ISBN 9780735623613

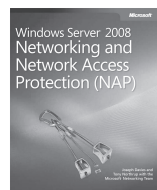
Your definitive reference for deployment and operations—from the experts who know the technology best. Get in-depth technical information on Active Directory®, Windows PowerShell™ scripting, advanced administration, networking and network access protection, security administration, IIS, and other critical topics—plus an essential toolkit of resources on CD.

ALSO AVAILABLE AS SINGLE VOLUMES



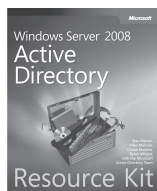
Windows Server 2008 Security Resource Kit

Jesper M. Johansson et al. with
Microsoft Security Team
ISBN 9780735625044



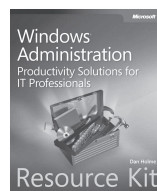
Windows Server 2008 Networking and Network Access Protection (NAP)

Joseph Davies, Tony Northrup,
Microsoft Networking Team
ISBN 9780735624221



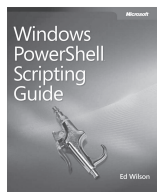
Windows Server 2008 Active Directory Resource Kit

Stan Reimer et al. with
Microsoft Active Directory Team
ISBN 9780735625150



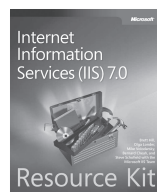
Windows® Administration Resource Kit: Productivity Solutions for IT Professionals

Dan Holme
ISBN 9780735624313



Windows Powershell Scripting Guide

Ed Wilson
ISBN 9780735622791



Internet Information Services (IIS) 7.0 Resource Kit

Mike Volodarsky et al. with
Microsoft IIS Team
ISBN 9780735624412

microsoft.com/mspress

Microsoft®
Press

What do you think of this book?

We want to hear from you!

To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

Tell us how well this book meets your needs—what works effectively, and what we can do better. Your feedback will help us continually improve our books and learning resources for you.

Thank you in advance for your input!

Microsoft
Press

Stay in touch!

To subscribe to the *Microsoft Press® Book Connection Newsletter*—for news on upcoming books, events, and special offers—please visit:

microsoft.com/learning/books/newsletter