

# Introducing Windows Server 2008

*Mitch Tulloch with the  
Microsoft Windows Server  
Team*

[Purchase select Microsoft Press books at a discount](#)  
(available in the United States only)

To learn more about this book, visit Microsoft Learning at  
<http://www.microsoft.com/MSPress/books/11163.aspx>

9780735624214  
Publication Date: May 2007

**Microsoft®**  
*Press*

## Additional Resources for IT Professionals

Published and Forthcoming Titles from Microsoft Press

### → Windows Server

Microsoft® Windows Server® 2003  
*Resource Kit*

Microsoft MVPs and Partners with  
Microsoft Windows Server Team  
978-0-7356-2232-6

Microsoft Windows Server 2003  
*Administrator's Companion*  
Second Edition

Charlie Russel, Sharon Crawford,  
and Jason Gerend  
978-0-7356-2047-6

Microsoft Windows Server 2003  
*Inside Out*

William R. Stanek  
978-0-7356-2048-3

Microsoft Windows Server 2003  
*Administrator's Pocket Consultant*  
Second Edition

William R. Stanek  
978-0-7356-2245-6

### → Windows Client

Windows Vista™  
*Resource Kit*

Tulloch, Northrup, Honeycutt,  
Russel, and Wilson with the  
Microsoft Windows Vista Team  
978-0-7356-2283-8

Windows Vista  
*Administrator's Pocket Consultant*

William R. Stanek  
978-0-7356-2296-8

Microsoft Windows® XP  
Professional  
*Resource Kit*  
Third Edition

The Microsoft Windows Team with  
Charlie Russel and Sharon Crawford  
978-0-7356-2167-1

Microsoft Windows XP  
Professional  
*Administrator's Pocket Consultant*  
Second Edition

William R. Stanek  
978-0-7356-2140-4

Microsoft Windows Command-Line  
*Administrator's Pocket Consultant*

William R. Stanek  
978-0-7356-2038-4

### → SQL Server 2005

Microsoft SQL Server™ 2005  
*Administrator's Pocket Consultant*

William R. Stanek  
978-0-7356-2107-7

Microsoft SQL Server 2005  
*Administrator's Companion*

Whalen, Garcia, et al.  
978-0-7356-2198-5

Inside Microsoft SQL Server 2005:  
*The Storage Engine*

Kalen Delaney  
978-0-7356-2105-3

Inside Microsoft SQL Server 2005:  
*T-SQL Programming*

Itzik Ben-Gan, Dejan Sarka, and  
Roger Wolter  
978-0-7356-2197-8

### → Exchange Server 2007

Microsoft Exchange Server 2007  
*Administrator's Companion*

Walter Glenn and Scott Lowe  
978-0-7356-2350-7

Microsoft Exchange Server 2007  
*Administrator's Pocket Consultant*

William R. Stanek  
978-0-7356-2348-4

### → Scripting

Microsoft Windows PowerShell™  
*Step by Step*

Ed Wilson  
978-0-7356-2395-8

Microsoft VBScript  
*Step by Step*

Ed Wilson  
978-0-7356-2297-5

Microsoft Windows  
Scripting with WMI:  
Self-Paced Learning Guide

Ed Wilson  
978-0-7356-2231-9

Advanced VBScript for Microsoft  
Windows Administrators

Don Jones and Jeffery Hicks  
978-0-7356-2244-9

#### RELATED TITLES



Microsoft Office  
SharePoint® Server  
2007 *Administrator's Companion*  
Bill English with the  
Microsoft SharePoint  
Community Experts  
978-0-7356-2282-1



Microsoft Windows  
Security  
*Resource Kit*  
Second Edition  
Ben Smith and Brian  
Komar with the  
Microsoft Security  
Team  
978-0-7356-2174-9



Microsoft Windows  
Small Business  
Server 2003 R2  
*Administrator's Companion*  
Charlie Russel and  
Sharon Crawford  
978-0-7356-2280-7



Microsoft Internet  
Security and  
Acceleration (ISA)  
Server 2004  
*Administrator's Pocket Consultant*  
Bud Ratliff and Jason  
Ballard with the Microsoft  
ISA Server Team  
978-0-7356-2188-6

# Resources for IT Professionals



## Administrator's Pocket Consultant

- Practical, portable guide for fast answers when you need them
- Focus on core operations and support tasks
- Organized for quick, precise reference—to get the job done



## Administrator's Companion

- Comprehensive, one-volume guide to deployment and system administration
- Real-world insights, procedures, troubleshooting tactics, and workarounds
- Fully searchable eBook on CD



## Resource Kit

- In-depth technical information and tools from those who know the technology best
- Definitive reference for deployment and operations
- Essential toolkit of resources, including eBook, on CD



## Self-Paced Training Kit

- Two products in one: official exam prep guide + practice tests
- Features lessons, exercises, and case scenarios
- Comprehensive self-tests; trial software; eBook on CD

## Available in 2008 from Microsoft Press

### Windows Server

Windows Server® 2008  
*Resource Kit*  
978-0-7356-2361-3

Windows Server 2008  
Active Directory®  
*Resource Kit*  
978-0-7356-2515-0

Windows Server 2008  
Virtualization  
*Resource Kit*  
978-0-7356-2517-4

Windows Server 2008  
Security *Resource Kit*  
978-0-7356-2504-4

Windows® Administration  
*Resource Kit: Productivity  
Solutions For IT Professionals*  
978-0-7356-2431-3

Windows Server 2008  
Networking Guide  
978-0-7356-2422-1

Windows Server 2008 TCP/IP  
Protocols and Services  
978-0-7356-2447-4

Windows Server 2008  
*Inside Out*  
978-0-7356-2438-2

Windows Server 2008  
Terminal Services  
978-0-7356-2516-7

Windows Server 2008  
*Administrator's Companion*  
978-0-7356-2505-1

Windows Server 2008  
*Administrator's Pocket Consultant*  
978-0-7356-2437-5

Windows Group Policy Guide,  
Second Edition  
978-0-7356-2514-3

Understanding IPv6,  
Second Edition  
978-0-7356-2446-7

### Internet Information Services

Internet Information  
Services (IIS) 7.0  
*Administrator's Pocket Consultant*  
978-0-7356-2364-4

Internet Information  
Services (IIS) 7.0  
*Resource Kit*  
978-0-7356-2441-2

### Scripting

Windows PowerShell™  
Scripting Guide  
978-0-7356-2279-1

Windows PowerShell  
& Command-line  
*Administrator's Pocket Consultant*  
978-0-7356-2262-3

### Certification

MCITP Self-Paced Training Kit  
(Exams 70-640, 70-642,  
70-643, 70-646): Windows Server  
Administrator Core Requirements  
978-0-7356-2508-2

MCITP Self-Paced Training Kit  
(Exam 70-640): Configuring  
Windows Server 2008  
Active Directory  
978-0-7356-2513-6

MCITP Self-Paced Training Kit  
(Exam 70-642): Configuring  
Windows Server 2008  
Network Infrastructure  
978-0-7356-2512-9

MCITP Self-Paced Training Kit  
(Exam 70-643): Configuring  
Windows Server 2008  
Applications Platform  
978-0-7356-2511-2

MCITP Self-Paced Training Kit  
(Exam 70-646): Windows Server  
2008 Administrator  
978-0-7356-2510-5

MCITP Self-Paced Training Kit  
(Exam 70-647): Windows Server  
2008 Enterprise Administrator  
978-0-7356-2509-9

See our full line of learning resources at: [microsoft.com/mspress](http://microsoft.com/mspress) and [microsoft.com/learning](http://microsoft.com/learning)

**Microsoft®**

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

Copyright © 2007 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: 2007924650

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 QWT 2 1 0 9 8 7

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

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

Chapter 4 contains the “From the Experts: WMI Remote Connection” sidebar. Copyright © 2007 by Alain Lissor.

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](http://www.microsoft.com/mspress). Send comments to [tkinput@microsoft.com](mailto:tkinput@microsoft.com).

Microsoft, Microsoft Press, Active Directory, ActiveX, Aero, BitLocker, ClearType, Direct3D, Excel, Internet Explorer, Microsoft Dynamics, MSDN, MS-DOS, Outlook, PowerPoint, SharePoint, SQL Server, Terminal Services RemoteApp, Visual Basic, Visual Studio, Visual Web Developer, Win32, Windows, Windows CardSpace, Windows Live, Windows Media, Windows Mobile, Windows NT, Windows PowerShell, Windows Server, Windows Server System, Windows Vista, and WinFX are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. 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:** Denise Bankaitis

Body Part No. X13-72717

# Table of Contents

<i>Preface</i> .....	xiii
<b>1 Introduction</b> .....	<b>1</b>
What's Between the Sheets .....	3
Acknowledgments .....	4
One Last Thing—Humor .....	7
<b>2 Usage Scenarios</b> .....	<b>9</b>
Providing an Identity and Access Infrastructure .....	10
Ensuring Security and Policy Enforcement .....	10
Easing Deployment Headaches .....	11
Making Servers Easier to Manage .....	12
Supporting the Branch Office .....	13
Providing Centralized Application Access .....	13
Deploying Web Applications and Services .....	14
Ensuring High Availability .....	14
Ensuring Secure and Reliable Storage .....	15
Leveraging Virtualization .....	16
Conclusion .....	16
<b>3 Windows Server Virtualization</b> .....	<b>17</b>
Why Enterprises Love Virtualization .....	17
Server Consolidation .....	18
Business Continuity .....	18
Testing and Development .....	19
Application Compatibility .....	19
Virtualization in the Datacenter .....	19

 **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:

[www.microsoft.com/learning/booksurvey/](http://www.microsoft.com/learning/booksurvey/)

Virtualization Today .....	20
Monolithic Hypervisor .....	22
Microkernelized Hypervisor .....	22
Understanding Virtualization in Windows Server 2008 .....	24
Partition 1: Parent .....	25
Partition 2: Child with Enlightened Guest .....	26
Partition 3: Child with Legacy Guest .....	27
Partition 4: Child with Guest Running Linux .....	28
Features of Windows Server Virtualization .....	28
Managing Virtual Machines in Windows Server 2008 .....	29
System Center Virtual Machine Manager 2007 .....	36
SoftGrid Application Virtualization .....	36
Conclusion .....	37
Additional Reading .....	37
<b>4 Managing Windows Server 2008 .....</b>	<b>39</b>
Performing Initial Configuration Tasks .....	39
Using Server Manager .....	42
Managing Server Roles .....	44
ServerManagerCmd.exe .....	50
Remote Server Administration Tools .....	53
Other Management Tools .....	56
Group Policy .....	56
Windows Management Instrumentation .....	59
Windows PowerShell .....	64
Microsoft System Center .....	68
Conclusion .....	69
Additional Resources .....	69
<b>5 Managing Server Roles .....</b>	<b>71</b>
Understanding Roles, Role Services, and Features .....	71
Available Roles and Role Services .....	72
Available Features .....	83

Adding Roles and Features .....	95
Using Initial Configuration Tasks .....	97
Using Server Manager .....	104
From the Command Line .....	105
Conclusion .....	108
Additional Reading .....	108
<b>6 Windows Server Core .....</b>	<b>109</b>
What Is a Windows Server Core Installation? .....	109
Understanding Windows Server Core .....	111
The Rationale for Windows Server Core .....	115
Performing Initial Configuration of a Windows Server Core Server .....	118
Performing Initial Configuration from the Command Line .....	118
Managing a Windows Server Core Server .....	130
Local Management from the Command Line .....	130
Remote Management Using Terminal Services .....	137
Remote Management Using the Remote Server Administration Tools ...	140
Remote Administration Using Group Policy .....	141
Remote Management Using WinRM/WinRS .....	142
Windows Server Core Installation Tips and Tricks .....	143
Conclusion .....	147
Additional Resources .....	147
<b>7 Active Directory Enhancements .....</b>	<b>149</b>
Understanding Identity and Access in Windows Server 2008 .....	149
Understanding Identity and Access .....	149
Identity and Access in Windows 2000 Server .....	150
Identity and Access in Windows Server 2003 .....	151
Identity and Access in Windows Server 2003 R2 .....	152
Identity and Access in Windows Server 2008 .....	153
Active Directory Domain Services .....	158
AD DS Auditing Enhancements .....	158
Read-Only Domain Controllers .....	164
Restartable AD DS .....	168
Granular Password and Account Lockout Policies .....	169

Active Directory Lightweight Directory Services .....	172
Active Directory Certificate Services .....	176
Certificate Web Enrollment Improvements .....	176
Network Device Enrollment Service Support .....	177
Online Certificate Status Protocol Support .....	177
Enterprise PKI and CAPI2 Diagnostics .....	179
Other AD CS Enhancements .....	180
Active Directory Federation Services .....	182
Active Directory Rights Management Services .....	186
Conclusion .....	187
Additional Resources .....	187
<b>8 Terminal Services Enhancements .....</b>	<b>189</b>
Core Enhancements to Terminal Services .....	190
Remote Desktop Connection 6.0 .....	191
Single Sign-On for Domain-joined Clients .....	200
Other Core Enhancements .....	201
Installing and Managing Terminal Services .....	209
Terminal Services RemoteApp .....	216
Using TS RemoteApp .....	217
Benefits of TS RemoteApp .....	225
Terminal Services Web Access .....	226
Using TS Web Access .....	227
Benefits of TS Web Access .....	232
Terminal Services Gateway .....	232
Implementing TS Gateway .....	235
Benefits of TS Gateway .....	237
Terminal Services Licensing .....	238
Other Terminal Services Enhancements .....	243
Terminal Services WMI Provider .....	243
Windows System Resource Manager .....	246
Terminal Services Session Broker .....	247
Conclusion .....	249
Additional Resources .....	250



<b>9</b>	<b>Clustering Enhancements</b>	<b>251</b>
	Failover Clustering Enhancements	252
	Goals of Clustering Improvements	253
	Understanding the New Quorum Model	254
	Understanding Storage Enhancements	256
	Understanding Networking and Security Enhancements	259
	Other Security Improvements	261
	Validating a Clustering Solution	261
	Tips for Validating Clustering Solutions	266
	Setting Up and Managing a Cluster	267
	Creating a Highly Available File Server	269
	Performing Other Cluster Management Tasks	273
	Network Load Balancing Enhancements	278
	Conclusion	283
	Additional Resources	283
<b>10</b>	<b>Network Access Protection</b>	<b>285</b>
	The Need for Network Access Protection	286
	Understanding Network Access Protection	287
	What NAP Does	288
	NAP Enforcement Methods	289
	Understanding the NAP Architecture	297
	A Walkthrough of How NAP Works	299
	Implementing NAP	301
	Choosing Enforcement Methods	302
	Phased Implementation	303
	Configuring the Network Policy Server	307
	Configuring NAP Clients	317
	Troubleshooting NAP	319
	Conclusion	339
	Additional Resources	340

<b>11</b>	<b>Internet Information Services 7.0</b>	<b>341</b>
	Understanding IIS 7.0 Enhancements	341
	Security and Patching	342
	Administration Tools	351
	Configuration and Deployment	360
	Diagnostics	365
	Extensibility	368
	What's New in IIS 7.0 in Windows Server 2008	370
	The Application Server Role	371
	Conclusion	374
	Additional Resources	375
<b>12</b>	<b>Other Features and Enhancements</b>	<b>377</b>
	Storage Improvements	378
	File Server Role	378
	Windows Server Backup	381
	Storage Explorer	384
	SMB 2.0	386
	Multipath I/O	387
	iSCSI Initiator	390
	iSCSI Remote Boot	397
	iSNS Server	401
	Networking Improvements	402
	Security Improvements	407
	Other Improvements	414
	Conclusion	419
	Additional Resources	419
<b>13</b>	<b>Deploying Windows Server 2008</b>	<b>421</b>
	Getting Windows Server 2008	421
	Installing Windows Server 2008	422
	Manual Installation	422
	Unattended Installation	423

Using Windows Deployment Services .....	423
Multicast Deployment .....	424
TFTP Windowing .....	427
EFI x64 Network Boot Support .....	430
Solution Accelerator for Windows Server Deployment. ....	431
Understanding Volume Activation 2.0 .....	432
Conclusion .....	439
Additional Resources .....	440
<b>14 Additional Resources .....</b>	<b>441</b>
Product Home Page .....	441
Microsoft Windows Server TechCenter .....	442
Microsoft Download Center .....	442
Microsoft Connect.....	443
Microsoft TechNet.....	445
Beta Central .....	445
TechNet Events.....	446
TechNet Virtual Labs.....	448
TechNet Community Resources .....	448
TechNet Columns.....	451
TechNet Magazine.....	451
TechNet Flash Newsletter.....	451
MSDN .....	451
Blogs .....	452
Blogs by MVPs .....	453
Channel 9 .....	454
Microsoft Press Books.....	454
Conclusion .....	455
<b>Index .....</b>	<b>457</b>



**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:

[www.microsoft.com/learning/booksurvey/](http://www.microsoft.com/learning/booksurvey/)

# Windows Server Core

**In this chapter:**

<b>What Is a Windows Server Core Installation? .....</b>	<b>109</b>
<b>Performing Initial Configuration of a Windows Server Core Server .....</b>	<b>118</b>
<b>Managing a Windows Server Core Server .....</b>	<b>130</b>
<b>Windows Server Core Installation Tips and Tricks .....</b>	<b>143</b>
<b>Conclusion .....</b>	<b>147</b>
<b>Additional Resources .....</b>	<b>147</b>

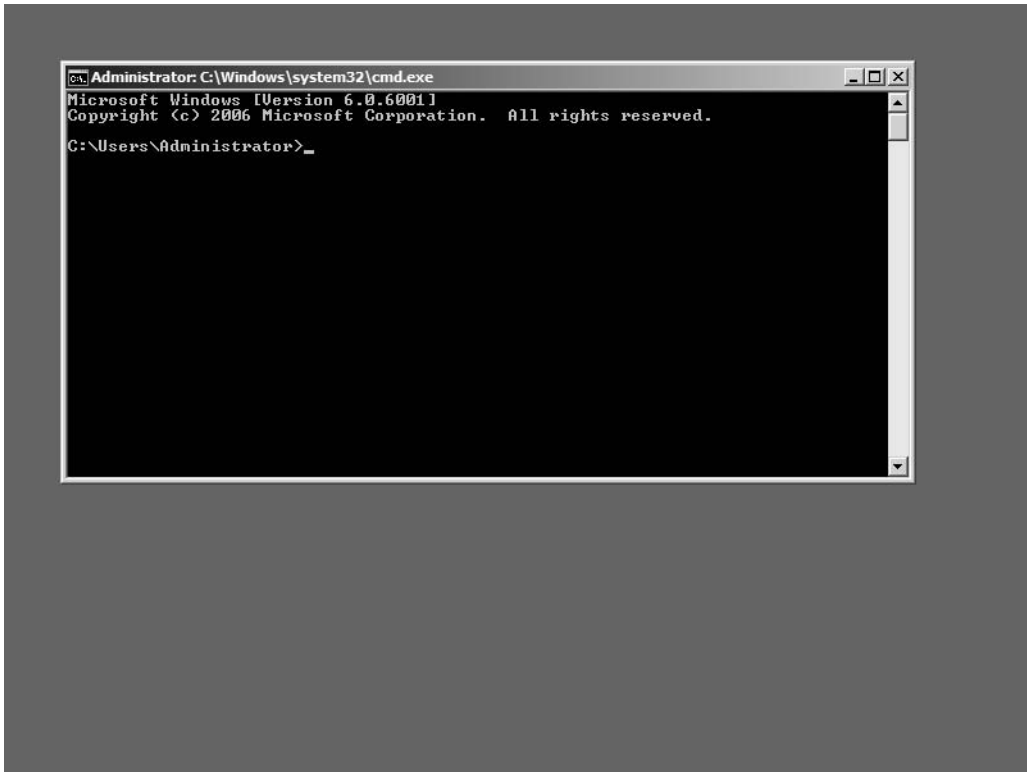
When you try to install Microsoft Windows Server 2008 manually from media on a system, you're presented with two installation options to choose from:

- A full installation of the Microsoft Windows Server 2008 operating system
- A Windows server core installation of the Windows Server 2008 operating system

Selecting the first option means you get the type of Windows server you're used to, with its full slate of GUI tools, support for the .NET Framework, and support for a wide range of possible roles and features you can install on your machine. But what if you select the second option? What's a Windows server core installation of Windows Server 2008? And how does this differ from a full installation of the product? Well, that's what this chapter is all about—read on!

## What Is a Windows Server Core Installation?

The best way of learning about the Windows server core installation option is to simply install it and log on. Here's what you see when you first log on to a Windows server core server.



That's it? Where's the task bar and Start menu? There is no task bar or Start menu. How do you start Windows Explorer then? You can't—the tool is not available in a Windows server core installation. Where's the Initial Configuration Tasks screen? It's not there. How can I open Server Manager to add roles and features? Sorry, Server Manager is unavailable on a Windows server core installation. Well, what can I do with this thing then? Am I stuck with only a command prompt to work with?

You can do a lot with a Windows server core installation, as we'll see in a moment. And no, you're not just stuck with a command prompt. But if you were, would it be bad? Ever hear a Unix admin complain about "being stuck" with having to use the command line to administer a server? Isn't command-line administration of servers a *good* thing because it means you can automate complex management tasks using batch files and scripts and there is no graphical UI taking resources away from server tasks?

And that's one of the things that a Windows server core installation is all about—scripted administration of Windows servers in enterprise (and especially datacenter) environments. But why remove the desktop and all the GUI management tools? Doesn't that cripple the server? Not at all—in fact, just the opposite!

## Understanding Windows Server Core

Windows server core is a “minimal” installation option for Windows Server 2008. What this means is that when you choose this option during setup (or when using unattended setup), Windows Server 2008 installs a minimum set of components on your machine that will allow you to run certain (but not all) server roles. In other words, selecting the Windows server core installation option installs only a subset of the binaries that are installed when you choose the full installation option for Windows Server 2008.

Here are some of the Windows Server 2008 components that are *not* installed when you specify the Windows server core installation option during setup:

- No desktop shell (which means no glass, wallpaper, or screen savers either)
- No Windows Explorer or My Computer (we already said no desktop shell, right?)
- No .NET Framework or CLR (which means no support for managed code, which also means no PowerShell support)
- No MMC console or snap-ins (so no Administrative tools on the Start menu—whoops! I forgot, no Start menu!)
- No Control Panel applets (with a few small exceptions)
- No Internet Explorer or Windows Mail or WordPad or Paint or Search window (no Windows Explorer!) or GUI Help and Support or even a Run box.

Wow, that sounds like a lot of stuff that’s missing in a Windows server core installation of Windows Server 2008! Actually though, it’s not—compare the preceding list to the following list of components that *are* available on a Windows server core server.

First, you’ve still got the kernel. You always need the kernel.

Then you’ve got hardware support components such as the Hardware Abstraction Layer (HAL) and device drivers. But it’s only a limited set of device drivers that supports disks, network cards, basic video support, and some other stuff. A lot of in-box drivers have been removed from the Windows server core installation option, however—though there is a way to install out-of-box drivers if you need to, as we’ll see later in this chapter.

Next, you’ve still got all the core subsystems that are needed by Windows Server 2008 in order to function. That means you’ve got the security subsystem and Winlogon, the networking subsystem, the file system, RPC and DCOM, SNMP support, and so on. Without these subsystems, your server simply wouldn’t be able to do anything at all, so they’re a necessity for a Windows server core installation.

Then you’ve got various components you need to configure different aspects of your server. For example, you have components that let you create user accounts and change passwords, enable DHCP or assign a static IP address, rename your server or join a domain, configure Windows Firewall, enable Automatic Updates, choose a keyboard layout, set the time and date, enable Remote Desktop, and so on. Many of these configuration tasks can be performed

using various command-line tools included in a Windows server core installation (more about tools in a moment), but a few of them use scripts or expose minimal UI.

There are some additional infrastructure components present as well on a Windows server core installation. For instance, you still have the event logs plus a command-line tool for viewing, configuring, and forwarding them using Windows eventing. You've got performance counters and a command-line tool for collecting performance information about your server. You have the Licensing service, so you can activate and use your server as a fully licensed machine. You've got IPSec support, so your server can securely communicate on the network. You've got NAP client support, so your server can participate in a NAP deployment. And you've got support for Group Policy of course.

Then there are various tools and infrastructure items to enable you to manage your Windows server core server. As we saw in our screen shot earlier, you've got the command prompt `cmd.exe`, so you can log on locally to your server and run various commands from a command-prompt window. In fact, as we saw, a command-prompt window is already open for you when you first log on to a Windows server core server. What happens, though, if you accidentally close this window? Fortunately, a Windows server core installation still includes Task Manager, so if you close your command window you can start another by doing the following:

1. Press CTRL+SHIFT+ESC, to open Task Manager.
2. On the Applications tab, click New Task.
3. Type `cmd` and click OK.

In addition to the command prompt, of course, there are dozens (probably over a hundred, and more when different roles and features are installed) of different command-line tools available on Windows Server 2008 for both full and server core installation options. What I'm talking about is `Arp`, `Assoc`, `At`, `Attrib`, `BCDEdit`, `Caccls`, `Certutil`, `Chdir`, `chkdsk`, `Cls`, `Copy`, `CScript`, `Defrag`, `Dir`, and so on. A lot of the commands listed in the "Windows Command-Line Reference A-Z," found on Microsoft TechNet, are available on a Windows server core server—not all, mind you, but a lot of them.

You can also enable Remote Desktop on a Windows server core installation, and this lets you connect to it from another machine using Remote Desktop Connection (RDC) and start a Terminal Services session running on it. Once you've established your session, you can use the command prompt to run various commands on your server, and you can even use the new Remote Programs feature of RDC 6.0 to run a remote command prompt on a Windows server core server from an administrative workstation running Windows Vista. (We'll learn more about that soon.)

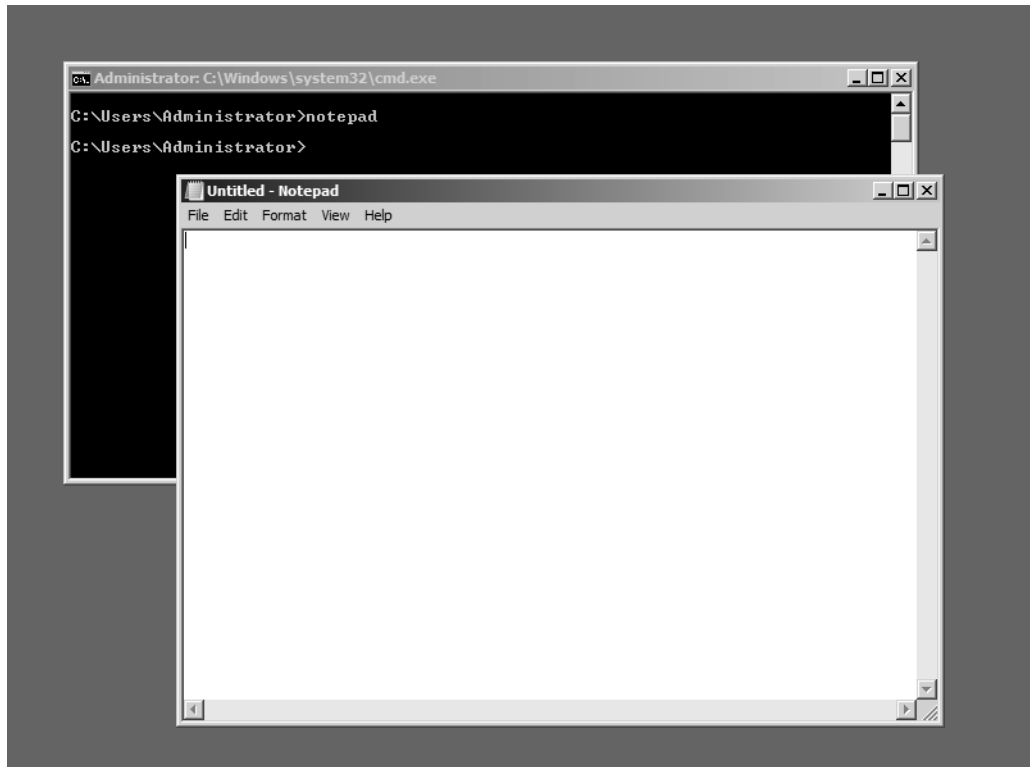
There's also a WMI infrastructure on your Windows server core server that includes many of the usual WMI providers. This means you can manage your Windows server core server either by running WMI scripts on the local machine from the command prompt or by scheduling their operation using `schtasks.exe`. (There's no Task Schedule UI available, however.) Or you can manage your server remotely by running remote WMI scripts against it from another machine. And having WMI on a Windows server core server means that remote UI tools

such as MMC snap-ins running on other systems (typically, either a full installation of Windows Server 2008 or an administrator workstation running Windows Vista with Remote Server Administration Tools installed) can connect to and remotely administer your Windows server core server. Plus there's also a WS-Management infrastructure on a Windows server core installation. WS-Management is a new remote-management infrastructure included in Windows Vista and Windows Server 2008, and involves Windows Remote Management (WinRM) on the machine being managed and the Windows Remote Shell (WinRM) for remote command execution from the machine doing the managing. We'll talk about remote management of Windows server core servers later in this chapter.

Then there are various server roles and optional features you can install on a Windows server core server so that the machine can actually do something useful on your network, like be a DHCP server or a domain controller or print server. We'll look later at exactly which roles and features are available for installing on a Windows server core server and which roles/features you can't install.

Then there are a few necessary GUI tools that actually *are* present on a Windows server core server. For example, we already saw that the command prompt (cmd.exe) is available, and so is Task Manager. Another useful tool on a Windows server core server is Regedit.exe, which can be launched either from the command line or from Task Manager. Then there's Notepad.

Notepad?





Yes, Notepad. The reason for including Notepad on a Windows server core installation option of Windows Server 2008 is simple: Microsoft listens to its customers. I'm not kidding! (Plus I'm serious about Microsoft listening to customers.) During the early stages of developing and testing Windows Server 2008, one of the most common requests from participants in the Microsoft Technology Adoption Program (TAP) for Windows Server 2008 was this: We need a tool on Windows server core servers that we can use to view logs, edit scripts, and perform other essential administrative tasks. Give us Notepad! We want Notepad!

Who ever expected that the lowly and oft-maligned Notepad would be so important to administrators who work in enterprise environments?

Anyway, before we move on and talk a bit about the rationale behind why Microsoft decided to offer the Windows server core installation option in Windows Server 2008, let's hear from one of our experts about how the Windows server core product team managed to make this thing work. After all, Windows components have a lot of dependencies with one another and especially with the desktop shell and Internet Explorer, so it will be interesting to hear how they took so many components out of this installation option for the product without causing it to break. Plus we'll also learn a bit about how we can try to get applications that we need to have running on a Windows server core server running properly. And finally, we'll learn something about getting Notepad to run properly on a Windows server core server:

### **From the Experts: Shimming Applications in Windows Server Core**

The primary goal of the Windows server core installation option is to minimize the disk and servicing footprint. Thus, a number of Windows components—such as Media Player and Internet Explorer—are not installed as part of a Windows server core installation. This means that because of their dependencies on parts of Internet Explorer, the common dialog boxes are not functional in a Windows server core installation. Thus, the file open and save dialog boxes in Notepad, for example, will not work.

A Windows server core installation leverages the application compatibility shim infrastructure in Windows to develop a clever solution to this problem. A *shim* is a thin layer of code that sits between an application and a Windows API. The shimming infrastructure redirects the API call made by the application to the shim code, which can then make some changes to the parameters, call the original API, or do something else entirely.

A Windows server core installation installs two shims. The first one is called *RegEditImportExportLoadHive* and is a specialized shim that allows *RegEdit* to import and export registry files. The second shim is called *NoExplorerForGetFileName*. It's a general shim for file open and save dialog boxes and is currently used by Notepad. This second shim changes some parameters to the API call that displays the file open or save dialog so that the old-style dialog box from pre-Windows 95 is displayed, instead of the new Explorer-style dialog box.

The shimming engine allows the end user to apply existing shims to other applications. The tool used to do this is the Application Compatibility Toolkit. Copy the sysmain.sdb database located at %SYSTEMROOT%\AppPatch (or %SYSTEMROOT%\AppPatch\AppPatch64 on x64 machines) on the Windows server core machine to a Windows Server 2008 machine. Use the Application Compatibility Toolkit to edit the database. Copy the new database back to the Windows server core machine, and install it using sdbinst.exe, located at %SYSTEMROOT%\System32.

–Rahul Prasad

*Software Development Engineer, Windows Core Operating System Division*

## The Rationale for Windows Server Core

The need for something like the Windows server core installation option of Windows Server 2008 is pretty obvious. Windows Server today is frequently deployed to support a single role in an enterprise or to handle a fixed workload. For example, organizations often deploy the DHCP Server role on a dedicated Windows Server 2003 machine to provide dynamic addressing support for client computers on their network. Now think about that for a moment—you’ve just installed Windows Server 2003 with all its various services and components on a solid piece of hardware, just to use the machine as a DHCP server and nothing more. Or maybe as a file server as part of a DFS file system infrastructure you’re setting up for users. Or as a print server to manage a number of printers on your network. The point is, you’ve got Windows Server 2003 with all its features doing only one thing. Why do you need all those extra binaries on your machine then? And think about when you need to patch your system—you’ve got to apply all new software updates to the machine, even though the functionality that many of those updates fix will never actually be used on that particular system. Why should you have to patch IIS on your server if the server is not going to be used for hosting Web sites? And might not having IIS binaries on your server make it more vulnerable even though the IIS component is not actually being used on it or is even installed? The more stuff you’ve got on a box, the more difficult it is to secure (or to be sure that it’s secure) and the more complex it is to maintain.

Enter the Windows server core installation option of Windows Server 2008. Now, instead of installing all of Windows Server 2008 on your box while using only a portion of it, you can install a minimal subset of Windows Server 2008 binaries and you need to maintain only those particular binaries. The value proposition for enterprises of the Windows server core installation option is plain to see:

- Fewer binaries mean a reduced attack surface and, hence, a greater degree of protection for your network.
- Less functionality and a role-based paradigm also mean fewer services running on your machine and, therefore, again less attack surface.

- Fewer binaries also mean a reduced servicing surface, which means fewer patches, making your server easier to service and orienting your patch management cycle according to roles instead of boxes. Estimates indicate that using the Windows server core installation option can reduce the number of patches you need to apply to your server by as much as 50 percent compared with full installations of Windows Server 2008.
- Fewer roles and features also mean easier management of your servers and enable different members of your IT staff to specialize better according to the server roles they need to support.
- Finally, fewer binaries also mean less disk space needed for the core operating system components, which is a plus for datacenter environments in particular.

The Windows server core installation option of Windows Server 2008 is all of these and more, and it's included in the Standard, Enterprise, and Datacenter editions of Windows Server 2008. Windows server core is not a separate product or SKU—it's an installation option you can select during manual or unattended install. And it's available on both the x86 and x64 platforms of Windows Server 2008. (It's not available on IA64 and on the Web edition SKU of Windows Server 2008.) The bottom line? The Windows server core installation option of Windows Server 2008 is more secure and more reliable, and it requires less management overhead than using a full installation of Windows Server 2008 for an equivalent purpose in your enterprise.

A Windows server core server provides you with minimal server operating system functionality and a low attack surface for targeted roles. To give you a better idea of the functionality that is (and isn't) available in the Windows server core installation option, Table 6-1 shows included and excluded roles and Table 6-2 shows included and excluded optional features.

**Table 6-1 Included/Excluded Roles in the Windows Server Core Installation Option of Windows Server 2008**

<b>Roles available</b>	<b>Roles unavailable</b>
Active Directory	Active Directory Certificate Services
Active Directory LDS	Active Directory Federation Services
DHCP Server	Active Directory RMS
DNS Server	Application Server
File Services (includes DFSR and NFS)	Fax Server
Print Services	Network Policy and Access Services
Streaming Media Services	Terminal Services
Windows Server Virtualization	UDDI Services
	Web Server (IIS)
	Windows Deployment Services
	Windows SharePoint Services

**Table 6-2 Included/Excluded Features in the Windows Server Core Installation Option of Windows Server 2008**

Features available	Features unavailable
BitLocker Drive Encryption	.NET Framework 3.0
Failover Clustering	BITS Server Extensions
Multipath I/O	Connection Manager Administration Kit
Removable Storage Management	Desktop Experience
SNMP Services	Internet Printing Client
Subsystem for UNIX-based Applications	Internet Storage Naming Server
Telnet Client	LPR Port Monitor
Windows Server Backup	Message Queuing
WINS Server	Network Load Balancing
	Peer Name Resolution Protocol
	Remote Assistance
	Remote Server Administration Tools
	RPC over HTTP Proxy
	Simple TCP/IP Services
	SMTP Server
	Storage Manager for SANs
	Telnet Server
	TFTP Client
	Windows Internal Database
	Windows Process Activation Service
	Windows System Resource Manager (WSRM)
	Wireless Networking

## Performing Initial Configuration of a Windows Server Core Server

In Chapter 5, “Managing Server Roles,” we saw how to perform the initial configuration of a Windows Server 2008 server using the Initial Configuration Tasks screen. Of course, many of these initial configuration tasks can also be performed using an `unattend.xml` answer file during an unattended installation.

The Windows server core installation option of Windows Server 2008 can also have its initial configuration done in two ways: from the command line after a manual install, or by doing an unattended installation. In this chapter, we’re going to look only at the first method (using the command line after a manual install). For more information on unattended installation of Windows Server 2008, see Chapter 13, “Deploying Windows Server 2008.”

### Performing Initial Configuration from the Command Line

Some of the initial configuration tasks you will want to perform on a Windows server core server include the following:

- Set a password for the Administrator account.
- Set the date, time, and time zone.
- Configure networking, which might mean assigning a static IP address, subnet mask, and default gateway (unless DHCP is being used) and pointing the DNS settings to a domain controller.
- Changing the server’s name and joining the domain.

Other initial configuration tasks can include activating your server, enabling Automatic Updates, downloading and installing any available software updates, enabling Windows Error Reporting and the Customer Experience Improvement Program, and so on.

Let’s see how to perform some of these tasks.

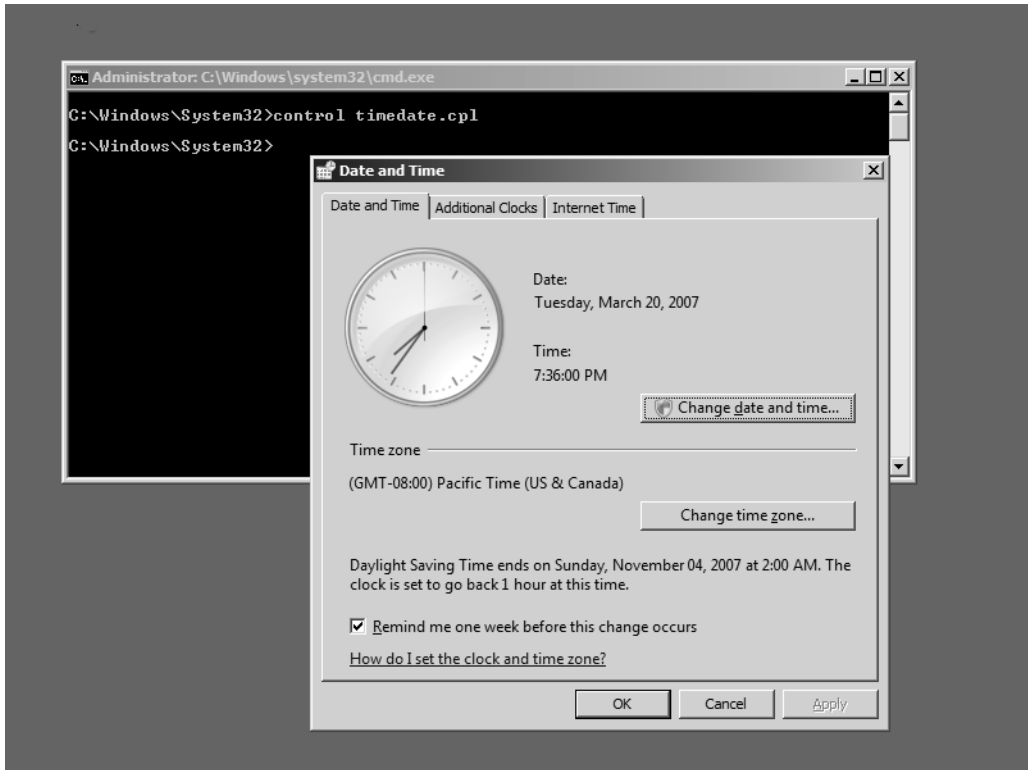
#### Changing the Administrator Password

There are two ways you can change the Administrator password on a Windows server core server:

- Press `CTRL+ALT+DEL`, click Change Password, and enter your old and new password.
- Type `net user administrator *` at the command prompt, and enter your new password twice.

## Setting Date, Time, and Time Zone

To set the time zone for your server, type **control timedate.cpl** at the command prompt. This opens the same Date And Time applet that can be opened from Control Panel in the full installation of Windows Server 2008:



The reason for using a Control Panel applet to do these tasks is simply that it's easier for admins to do it this way than to try and do it from the command line. And because it's a task that is likely to be performed only occasionally (even just once), and because there are no dependencies between the Date And Time applet and other system components that have been removed from the Windows server core installation option, the product team decided to leave this in as one of the few GUI tools still available in the Windows server core installation option of Windows Server 2008. Of course, you can also specify these settings in an `unattend.xml` answer file if you're performing an unattended installation of your server. And by the way, `control.exe` by itself doesn't work on a Windows server core installation. Only the two included `.cpls` work.

Before we go further, let's briefly hear from one of our experts on the Windows Server 2008 product team at Microsoft concerning configuring the Windows server core installation option of Windows Server 2008:

### **From the Experts: Shell-less vs. GUI-less**

If you have been working with a Windows server core installation, you might have noticed that there is some GUI support in a Windows server core installation of Windows Server 2008. To be completely accurate, the GUI of a Windows server core server is shell-less, not entirely GUI-less. There are several low-level GUI DLLs that are included because of current dependencies, such as `gdi32.dll` and `shlwapi.dll`. In a future release we hope to be able to remove the dependencies and also remove these files. However, including them does provide some advantages for making a Windows server core server easier to manage using the current tools.

In Beta 1, we didn't include any text editor. Although you could remotely connect to a Windows server core server to view logs, edit scripts, and so on, we heard lots of feedback that there should be an on-the-box text editor. Therefore, we added Notepad. However, because of the reduced environment the Windows server core installation option provides, not all of Notepad is functional—for example, help doesn't work.

In addition, the Windows server core installation option also includes two control panels, which you can access using the following commands:

- `Control timedate.cpl`
- `Control intl.cpl`

`Timedate.cpl` lets you set the time zone for your server, while `intl.cpl` lets you change your keyboard for different layouts.

—Andrew Mason  
Program Manager, Windows Server

## Configuring Networking

Now let's configure networking for our server. First let's run **ipconfig /all** and see the server's current networking settings:

```
C:\Windows\System32>ipconfig /all
Windows IP Configuration

Host Name . . . . . : LH-3TBCQ4I10NRA
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No


Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . :
    Description . . . . . : Intel 21140-Based PCI Fast Ethernet Adapter
    (Emulated)
    Physical Address. . . . . : 00-03-FF-27-88-8C
    DHCP Enabled. . . . . : Yes
    Autoconfiguration Enabled . . . . : Yes
    Link-local IPv6 Address . . . . . : fe80::c25:d049:5b0c:1585%2(Preferred)
    Autoconfiguration IPv4 Address. . : 169.254.21.133(Preferred)
    Subnet Mask . . . . . : 255.255.0.0
    Default Gateway . . . . . :
    DHCPv6 IAID . . . . . : 67109887
    DNS Servers . . . . . : fec0:0:0:ffff::1%1
                           fec0:0:0:ffff::2%1
                           fec0:0:0:ffff::3%1
    NetBIOS over Tcpip. . . . . : Enabled


Tunnel adapter Local Area Connection*:

    Connection-specific DNS Suffix  . :
    Description . . . . . : isatap.{B4B31F3D-B6C8-4303-BA3C-5A54B05F2FDD}
    Physical Address. . . . . : 00-00-00-00-00-00-E0
    DHCP Enabled. . . . . : No
    Autoconfiguration Enabled . . . . : Yes
    Link-local IPv6 Address . . . . . : fe80::5efe:169.254.21.133%3(Preferred)
    Default Gateway . . . . . :
    DNS Servers . . . . . : fec0:0:0:ffff::1%1
                           fec0:0:0:ffff::2%1
                           fec0:0:0:ffff::3%1
    NetBIOS over Tcpip. . . . . : Disabled
```

Note that **ipconfig /all** displays two network interfaces on the machine: a physical interface (NIC) and an ISATAP tunneling interface. Before we can use netsh.exe to modify network



settings, we need to know which interface we need to configure. To determine this, we'll use the **netsh interface ipv4 show interfaces** command as follows:

```
C:\Windows\System32>netsh interface ipv4 show interfaces
```

Idx	Met	MTU	State	Name
2	20	1500	connected	Local Area Connection
1	50	4294967295	connected	Loopback Pseudo-Interface 1

From this, we can see that our physical interface Local Area Connection has index number 2 (first column). Let's use this information to set the TCP/IP configuration for this interface. Here's what we want the settings to be:

- IP address: 172.16.11.162
- Subnet mask: 255.255.255.0
- Default gateway: 172.16.11.1
- Primary DNS server: 172.16.11.161
- Secondary DNS server: none

To do this, we can use two netsh.exe commands as follows:

```
C:\Windows\System32>netsh interface ipv4 set address name="2" source=static
address=172.16.11.162 mask=255.255.255.0 gateway=172.16.11.1

C:\Windows\System32>netsh interface ipv4 add dnsserver name="2" address=172.16.11.161
index=1
```

Now let's run **ipconfig /all** again and check the result:

```
C:\Windows\System32>ipconfig /all
Windows IP Configuration

Host Name . . . . . : LH-3TBCQ4I10NRA
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . :
Description . . . . . : Intel 21140-Based PCI Fast Ethernet Adapter
(Emulated)
Physical Address. . . . . : 00-03-FF-27-88-8C
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::c25:d049:5b0c:1585%2(Preferred)
```

```

IPv4 Address. . . . . : 172.16.11.162(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 172.16.11.1
DNS Servers . . . . . : 172.16.11.161
NetBIOS over Tcpip. . . . . : Enabled

Tunnel adapter Local Area Connection*:

Connection-specific DNS Suffix . :
Description . . . . . : isatap.{B4B31F3D-B6C8-4303-BA3C-5A54B05F2FDD}
Physical Address. . . . . : 00-00-00-00-00-00-E0
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::5efe:172.16.11.162%3(Preferred)
Default Gateway . . . . . :
DNS Servers . . . . . : 172.16.11.161
NetBIOS over Tcpip. . . . . : Disabled

```

So far, so good. Let's move on.

## Changing the Server's Name

Next let's change the name of our server. When you install a Windows server core server manually from media, the server is assigned a randomly generated name. We want to change that, and we can use `netdom.exe` to do this. First let's see what the current name is, and then let's change it to `DNSSRV` because we're planning on using this particular machine as a DNS server on our network:

```

C:\Windows\System32>hostname
LH-3TBCQ4I10NRA

C:\Windows\System32>netdom renamecomputer %computename% /NewName:DNSSRV
This operation will rename the computer LH-3TBCQ4I10NRA
to DNSSRV.

Certain services, such as the Certificate Authority, rely on a fixed machine
name. If any services of this type are running on LH-3TBCQ4I10NRA,
then a computer name change would have an adverse impact.

Do you want to proceed (Y or N)?
y
The computer needs to be restarted in order to complete the operation.

The command completed successfully.

```

We can restart the server using the `shutdown /r /t 0` command. Once the machine is restarted, typing `hostname` shows that the server's name has been successfully changed:

```

C:\Windows\System32>hostname
DNSSRV

```

## Joining a Domain

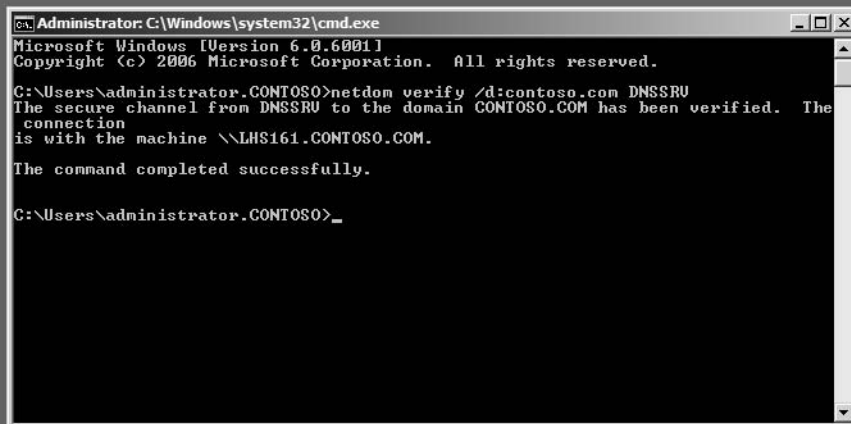
Now let's join our server to our domain. We'll use `netdom.exe` again to do this, and we're going to join our server to a domain named `contoso.com`. Here's how we do this:

```
C:\Windows\System32>netdom join DNSSRV /domain:CONTOSO /userd:Administrator /
passwordd:*
Type the password associated with the domain user:

The computer needs to be restarted in order to complete the operation.

The command completed successfully.
```

Again, we'll use `shutdown /r /t 0` to restart the machine. Once it's restarted, we'll log on as a domain admin this time and use `netdom.exe` again to verify that our server has established a secure channel to the domain controller.



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Users\administrator.CONTOSO>netdom verify /d:contoso.com DNSSRV
The secure channel from DNSSRV to the domain CONTOSO.COM has been verified. The
connection
is with the machine \\LHS161.CONTOSO.COM.

The command completed successfully.

C:\Users\administrator.CONTOSO>_
```

## Activating the Server

To activate our server, we can use a built-in script named `slmgr.vbs` found in the `%windir%\System32` directory. (This script is also in Windows Vista and in full installations of Windows Server 2008, and it can be run remotely from those platforms to activate a Windows server core installation.) Typing `cscript slmgr.vbs /?` shows the available syntax for this command:

```
C:\Windows\System32>cscript slmgr.vbs /?
Windows Software Licensing Management Tool
Usage: slmgr.vbs [MachineName [User Password]] [<Option>]
        MachineName: Name of remote machine (default is local machine)
        User:        Account with required privilege on remote machine
        Password:    password for the previous account

Global Options:
-ipk <Product Key>
    Install product key (replaces existing key)
-upk
    Uninstall product key
-ato
    Activate Windows
-dli [Activation ID | All]
    Display license information (default: current license)
-dlv [Activation ID | All]
    Display detailed license information (default: current license)
-xpr
    Expiration date for current license state

Advanced Options:
-cpky
    Clear product key from the registry (prevents disclosure attacks)
-ilc <License file>
    Install license
-rilc
    Re-install system license files
-rearm
    Reset the licensing status of the machine
-dti
    Display Installation ID for offline activation
-atp <Confirmation ID>
    Activate product with user-provided Confirmation ID
```

Let's first use the `-xpr` option to display the expiration date for the current license state:

```
C:\Windows\system32>cscript slmgr.vbs -xpr
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

Initial grace period ends 3/31/2007 1:13:00 AM
```

Now let's use **-dli** to display more info concerning the server's current license state:

```
C:\Windows\system32>cscript slmgr.vbs -dli
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

Name: Windows(TM) Server 2008, ServerEnterpriseCore edition
Description: Windows Operating System - Windows Server 2008, RETAIL channel
Partial Product Key: XHKDR
License Status: Initial grace period
Time remaining: 14533 minute(s) (10 day(s))
```

Now let's activate the server using the **-ato** option:

```
C:\Windows\system32>cscript slmgr.vbs -ato
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

Activating Windows(TM) Server 2008, ServerEnterpriseCore edition
(f00d81ce-df2c-47cb-a359-36d652296e56) ...
Product activated successfully.
```

Finally, let's try the **-xpr** and **-dli** options again and see the result:

```
C:\Windows\system32>cscript slmgr.vbs -xpr
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

The machine is permanently activated.

C:\Windows\system32>cscript slmgr.vbs -dli
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

Name: Windows(TM) Server code name "Longhorn", ServerEnterpriseCore edition
Description: Windows Operating System - Server code name "Longhorn", RETAIL channel
Partial Product Key: XHKDR
License Status: Licensed
```

## Enabling Automatic Updates

To enable Automatic Updates on our server, we'll use another built-in script named `scregedit.wsf`. This script is unique to the Windows server core installation option of Windows Server 2008, and it's one of the few binaries on a Windows server core server that is

not found on a full installation of Windows Server 2008. To view the syntax of this script, type `cscript scregedit.wsf /?` at the command prompt:

```
C:\Windows\System32>cscript scregedit.wsf /?
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

Automatic Updates - Manage Automatic Windows Updates
These settings can be used to configure how Automatic Updates are applied to the
Windows system. It includes the ability to disable automatic updates and to set the
installation schedule.

/AU [/v][value]

/v    View the current Automatic Update settings
value  value you want to set to.

Options:
4 - Enable Automatic Updates
1 - Disable Automatic Updates

Windows Error Reporting Settings
Windows can send descriptions of problems on this server to Microsoft. If you choose
to automatically send generic information about a problem, Microsoft will use the
information to start working on a solution.

This setting might be overridden by the following Group Policy:
    Key : Software\Policies\Microsoft\Windows\Windows Error Reporting\Consent,
    Value : DefaultConsent

/ER [/v][value]
/v    View the current Windows Error Reporting settings
value  value you want to set to.

Opt-in Settings:
2 - Automatically send summary reports (Recommended)
3 - Automatically send detailed reports
1 - Disable Windows Error Reporting

For more information on what data information is collected, go to
http://go.microsoft.com/fwlink/?linkid=50163

Terminal Service - Allow Remote Administration Connections
This allows administrators to connect remotely for administration purposes.

/AR [/v][value]

/v    View the Remote Terminal Service Connection setting
value  (0 = enabled, 1 = disabled)

Terminal Service - Allow connections from previous versions of Windows
```

This setting configures CredSSP based user authentication for Terminal Service connections

/CS [/v][value]

/v View the Terminal Service CredSSP setting  
value (0 = allow previous versions, 1 = require CredSSP)

IP Security (IPSEC) Monitor - allow remote management

This setting configures the server to allow the IP Security (IPSEC) Monitor to be able to remotely manage IPSEC.

/IM [/v][value]

/v View the IPSEC Monitor setting  
value (0 = do not allow, 1 = allow remote management)

DNS SRV priority - changes the priority for DNS SRV records

This setting configures the priority for DNS SRV records and is only useful on Domain Controllers.

For more information on this setting, search TechNet for LdapSrvPriority

/DP [/v][value]

/v View the DNS SRV priority setting  
value (value from 0 through 65535. The recommended value is 200.)

DNS SRV weight - changes the weight for DNS SRV records

This setting configures the weight for DNS SRV records and is useful only on Domain Controllers.

For more information on this setting, search TechNet for LdapSrvWeight

/DW [/v][value]

/v View the DNS SRV weight setting  
value (value from 0 through 65535. The recommended value is 50.)

Command Line Reference

This setting displays a list of common tasks and how to perform them from the command line.

/CLI

First let's see what the current setting for Automatic Updates is on the machine:

```
C:\Windows\system32>cscript scregedit.wsf /au /v
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update AUOptions
Value not set.
```

Looks like Automatic Updates is not yet configured, so let's enable it:

```
C:\Windows\system32>cscript scregedit.wsf /au 4
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

Registry has been updated.
```

Now let's verify by using our previous command:

```
C:\Windows\system32>cscript scregedit.wsf /au /v
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update AUOptions
View registry setting.
4
```

Note that on a Windows server core server you can configure Automatic Updates only to download and install updates automatically. You can't configure it to download updates and prompt you to install them later.

There are other initial configuration tasks we could do, but let's move on. Actually, let's hear first from one of our experts concerning a configuration task that's *not* easy to do from the command line:

### From the Experts: Configuring Display Resolution

Although there is no tool on a Windows server core server to allow you to change your display resolution, you can configure this by using an unattend file. However, it is possible to change the display resolution so that you can run at a higher resolution than what you might have ended up with at the end of setup. Doing this requires editing the registry; however, if you pick a resolution your video card or monitor cannot display, you might have to reinstall—although you should still be able to boot and remotely modify the settings in the registry.



To do this, you need to open regedit.exe and navigate to the following location:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Video

Under this will be a list of GUIDs, and you need to determine which one corresponds to your video card/driver. You might have to experiment to determine the right one. Under the GUID, you can set

\0000\DefaultSettings.XResolution

\0000\DefaultSettings.YResolution

to the resolution you would like to use. If these don't exist, you can create them. You must log off and log back on again for the change to take effect. Be careful doing this because if you specify an unsupported display resolution, you might need to reinstall your machine or remotely connect to the registry from another computer to change it, and remotely reboot.

—Andrew Mason

*Program Manager, Windows Server*

## Managing a Windows Server Core Server

Once we've performed initial configuration of our Windows server core server, we can then add roles and optional features so that it can provide needed functionality to our network. In this section, we're going to examine how to perform such common tasks, and we'll also look at different ways of managing a Windows server core server, including using the following:

- Local administration from the command prompt
- Remote administration using Terminal Services
- Remote administration using Remote Server Administration Tools
- Remote administration using Group Policy
- Remote administration using WinRM/WinRS

### Local Management from the Command Line

When we log on to the console of a Windows server core server, a command prompt appears. From this command prompt, we can do a lot of things:

- Run common tools such as netsh.exe and netdom.exe to perform various tasks, as we saw previously.
- Use special tools such as oclist.exe and ocsetup.exe to install roles and optional features on our server to give it more functionality.

- Run in-box scripts such as `slmgr.vbs` and `scregedit.wsf`, as we saw earlier, to perform certain kinds of tasks.
- Create our own scripts using Notepad, and run them using `Cscript.exe` and the supported WMI providers.
- Use the WMI command line (WMIC) to do almost anything from the command line that you can do by writing WMI scripts.

As we mentioned before, however, one thing you can't do is run PowerShell commands to administer your server. The reason for this omission is that PowerShell is managed code that requires the .NET Framework in order to work, and the .NET Framework is not included in the Windows server core installation option. Why? Because the .NET Framework has dependencies across the whole spectrum of different Windows components, and leaving it in would have increased the size of the Windows server core installation option until it was very nearly the size of a full installation of Windows Server 2008. For future versions of the Windows server core installation, however, a slimmed-down .NET Framework might be available that can provide PowerShell cmdlet functionality without the need of increasing the footprint significantly. But we'll have to see, as that's something that would happen after RTM. Note that you can however use PowerShell remotely to manage a Windows server core installation if the script strictly uses only WMI commands and not cmdlets.

Let's look how to perform two important tasks from the command line: adding server roles and adding optional features.

## Installing Roles

Let's start by seeing what roles are currently installed on our server and what roles are available to install. We'll use the `oclist.exe` command to do this:

```
C:\Windows\System32>oclist
```

```
Use the listed update names with Ocsetup.exe to install/uninstall a server role or optional feature.
```

```
Adding or removing the Active Directory role with OCSetup.exe is not supported. It can leave your server in an unstable state. Always use DCPromo to install or uninstall Active Directory.
```

```
=====
Microsoft-Windows-ServerCore-Package
=====
```

```
Not Installed:BitLocker
Not Installed:BitLocker-RemoteAdminTool
Not Installed:ClientForNFS-Base
Not Installed:DFSN-Server
Not Installed:DFSR-Infrastructure-ServerEdition
Not Installed:DHCPServerCore
Not Installed:DirectoryServices-ADAM-ServerCore
Not Installed:DirectoryServices-DomainController-ServerFoundation
```

```
Not Installed:DNS-Server-Core-Role
Not Installed:FailoverCluster-Core
Not Installed:FRS-Infrastructure
Not Installed:MediaServer
Not Installed:Microsoft-Windows-MultipathIo
Not Installed:Microsoft-Windows-RemovableStorageManagementCore
Not Installed:NetworkLoadBalancingHeadlessServer
Not Installed:Printing-ServerCore-Role
    |
    |--- Not Installed:Printing-LPDPrintService
    |
Not Installed:ServerForNFS-Base
Not Installed:SIS
Not Installed:SNMP-SC
Not Installed:SUACore
Not Installed:TelnetClient
Not Installed:WindowsServerBackup
Not Installed:WINS-SC
```

Note that the `oclist.exe` command displays information about both roles and features installed and not installed on the machine. We can see from the command output that the DNS Server role is not presently installed on the machine. We can also verify this by typing **net start** in the command line:

```
C:\Windows\System32>net start
These Windows services are started:

Application Experience
Background Intelligent Transfer Service
Base Filtering Engine
COM+ Event System
Computer Browser
Cryptographic Services
DCOM Server Process Launcher
DHCP Client
Diagnostic Policy Service
Diagnostic System Host
Distributed Transaction Coordinator
DNS Client
Group Policy Client
IKE and AuthIP IPsec Keying Modules...
```

In fact, the only DNS binaries presently installed are those for the DNS client:

```
C:\Windows\System32>dir dns*.*
Volume in drive C has no label.
Volume Serial Number is FC68-BDF4

Directory of C:\Windows\system32

02/09/2007  10:00 PM          163,840  dnsapi.dll
02/09/2007  09:59 PM           24,064  dnscacheugc.exe
02/09/2007  10:00 PM          84,480  dnsrslvr.dll
               3 File(s)      272,384 bytes
               0 Dir(s)  27,578,523,648 bytes free
```

Now let's install the DNS Server role using the `ocsetup.exe` command as follows:

```
C:\Windows\System32>start /w ocsetup DNS-Server-Core-Role
```

After a short while, the command prompt appears again. The reason we used the `/w` switch with **start** is because that way control is not returned to the command prompt until the **ocsetup** command finishes its work. (By the way, note that **ocsetup** is case sensitive.) Now if we type **oclist**, we should see that the DNS Server role has been added to our server:

```
C:\Windows\System32>oclist
...
Not Installed:DirectoryServices-ADAM-ServerCore
Not Installed:DirectoryServices-DomainController-ServerFoundation
    Installed:DNS-Server-Core-Role
Not Installed:FailoverCluster-Core
Not Installed:FRS-Infrastructure
...
```

We can also see that three additional binaries for DNS are now present on the server:

```
C:\Windows\System32>dir dns*.*
Volume in drive C has no label.
Volume Serial Number is FC68-BDF4

Directory of C:\Windows\system32

03/20/2007  11:59 PM    <DIR>          dns
02/09/2007  11:42 AM          484,864  dns.exe
02/09/2007  10:00 PM          163,840  dnsapi.dll
02/09/2007  09:59 PM           24,064  dnscacheugc.exe
02/09/2007  11:42 AM          162,816  dnscmd.exe
02/09/2007  11:42 AM           13,312  dnsperf.dll
02/09/2007  10:00 PM          84,480  dnsrslvr.dll
               6 File(s)      933,376 bytes
               1 Dir(s)  27,576,926,208 bytes free
```

And if we type **net stop dns**, we can now stop the DNS Server service without getting an error because the service is now present on the machine. Now that our machine is a DNS Server, we can use the `dnscmd.exe` command to further configure this role if we want from the command line.

Installing other server roles is similar to what we just did and uses the `ocsetup.exe` command, with the exception being that the process installs the Active Directory role. This is because `Dcpromo.exe` in Windows Server 2008 now installs the Active Directory binaries during promotion and uninstalls the binaries during demotion, so you should *not* use `ocsetup.exe` to add or remove the Active Directory role as then the promotion/demotion will not take place and your server may not function correctly.

Anyway, to add or remove the Active Directory role, you therefore have to use the `dcpromo.exe` tool, but you also have to run it in unattended mode because the GUI form of this tool (the Active Directory Installation Wizard) can't run on a Windows server core server because of the lack of a desktop shell to run it in. The syntax for running `dcpromo.exe` in unattended mode is **dcpromp /unattend:unattend.txt**, and a sample `unattend.txt` file you could use (or further customize) for doing this is as follows:

```
[DCInstall]
ReplicaOrNewDomain = Domain
NewDomain=Forest
NewDomainDNSName = contoso.com
AutoConfigDNS=Yes
DNSDelegation=Yes
DNSDelegationUserName=dnsuser
DNSDelegationPassword=p@ssword!
RebootOnSuccess = NoAndNoPromptEither
SafeModeAdminPassword = p@ssword!
```

For more information on using `dcpromo` in unattended mode, type **dcpromo /?:unattend** at the command prompt.

## Installing Optional Features

Installing optional features is very similar to installing roles. Type **oclist** to display a list of installed and uninstalled features and to determine the internal name of each feature. For example, the Failover Cluster feature is named `FailoverCluster-Core`, and we need to use this internal form of the name when we run `ocsetup` to install this feature. You can also remove features by adding an **/uninstall** switch to your **ocsetup** command. You can remove roles that way too, but be sure to stop the role's services before you remove the role.

## Other Common Management Tasks

There are lots of other common management tasks you might need to perform on a Windows server core server. The following is just a sampling of some of these tasks.

First, you can add new hardware to your server. Windows server core servers include support for Plug and Play. So if your new device is PnP and there's an in-box driver available for your device, you can just plug the device in and the server will recognize it and automatically install a driver for it. But we did mention earlier that the Windows server core server installation option of Windows Server 2008 does not include that many in-box drivers. So what do you do if your device is not supported by an in-box driver because of its date of manufacture? In that case, follow this procedure:

1. Copy the driver files from the driver media for the device to a temporary directory on your server.
2. Change your current directory to this temporary directory, and type **pnputil -i -a <driver>.inf** at the command prompt.
3. Reboot your server if prompted to do so.

Note that if you want to find what drivers are currently installed on your server, you can type **sc query type= driver** at a command prompt.

What if you want to install some application on your server? First of all, beware—any application that has a GUI might not function properly when you install it. Obviously, that means we can't install Microsoft Exchange Server, Microsoft SQL Server, or other Windows Server System products on a Windows server core server, because these products all have GUI management tools (and more importantly, a Windows server core server is missing a lot of components needed by these products such as the .NET Framework for running managed code).

What kinds of applications might you want to install on a Windows server core server? The usual stuff—antivirus agents, network backup agents, system management agents, and so on. Most agents like this are GUI-less and should install fine and work properly on a Windows server core server. And the Windows Installer service is yet another feature that's still present on a Windows server core server—and if you need to install an agent manually, you should try and do so in quiet mode using `msiexec.exe` with the **/qb** switch to display the basic UI only. For example, you can do this by typing **msiexec /qb <package>** at the command prompt.

If you need to configure Windows Firewall, the NAP client, or your server's IPSec configuration, you can use `netsh.exe` to do this. I won't go into all the details here, as you can just check TechNet for the proper `netsh.exe` syntax to use for each task.

What about patch management? We already described how to enable Automatic Updates on the server, and if you have Windows Server Update Service (WSUS) deployed, you can manage patches for your server using that as well. For Windows server core servers that you want

to manually perform patch management on, however, you can use the `wusa.exe` command to install and remove patches from the command prompt. To do this, first download the patch from Windows Update and expand to get the `.msu` file. Then copy the `.msu` file to your server, and type **`wusa <patch>.msu /quiet`** at the command prompt to install the patch. You can also remove installed patches from your server by typing **`pkgmgr /up /m:<package>.cab /quiet`** at the command prompt.

Let's hear more about patch management on a Windows server core installation of Windows Server 2008 from one of our experts:

### From the Experts: Servicing Windows Server Core

When using Windows server core, the new minimal installation option for Windows Server 2008, a common topic of discussion is servicing. First a little background and then some methods to make dealing with patches easier.

With Windows Server 2008, each patch that is released contains a set of applicability rules. When a patch is sent to a server, either by Windows Update or another automated servicing tool, the servicing infrastructure examines the patch to determine if it applies to the system based on the applicability rules. If not, it is ignored and nothing is changed on the server.

If you have already downloaded a set of patches and want to determine if they apply to a Windows server core installation, you can do the following:

1. Run **`wusa <patch_name>`**.
2. If the dialog box that appears asks if you want to apply the patch, click No. This means that the patch applies, and you should move on to the next step. Otherwise, the dialog box will state that the patch doesn't apply and you can ignore the patch.
3. Run **`wusa <patch_name> /quiet`** to apply the patch.

After applying patches, you can run either the `wmic qfe` command or `systeminfo.exe` to see what patches are installed.

—Andrew Mason

*Program Manager, Windows Server*

What else can you do in terms of managing your Windows server core installation of Windows Server 2008? Lots! For example, if you need to manage your disks and file system on your server, you can use commands such as `diskpart`, `defrag`, `fsutil`, `vssadmin`, and so on. And if you need to manage permissions and ownership of files, you can use `icacls`.

You can also manage your event logs from the command line using the `wevtutil.exe` command, which is new in Windows Vista and Windows Server 2008. This powerful command can be used to query your event logs for specific events and to export,

archive, clear, and configure your event logs as well. For example, to query your System log for the most recent occurrence of a shutdown event having source USER32 and event ID 1074, you can do this:

```
C:\Windows\system32>wevtutil qe System /c:1 /rd:true /f:text /
q:*[System[(EventID=1074)]]
Event[0]:
  Log Name: System
  Source: USER32
  Date: 2007-03-20T22:26:36.000
  Event ID: 1074
  Task: N/A
  Level: Information
  Opcode: N/A
  Keyword: Classic
  User: S-1-5-21-3620207985-2970159875-1752314906-500
  User Name: DNSSRV\Administrator
  Computer: DNSSRV
  Description:
    The process C:\Windows\system32\shutdown.exe (DNSSRV) has initiated the restart of
    computer DNSSRV on behalf of user DNSSRV\Administrator for the following reason: No
    title for this reason could be found
    Reason Code: 0x840000ff
    Shutdown Type: restart
    Comment:
```

To create and manage data collectors for performance monitoring, you can use the logman.exe command. You can also use the relog.exe command to convert a performance log file into a different format or change its sampling rate. And you can use the tracerpt.exe command to create a remote from a log file or a real-time stream of performance-monitoring data.

To manage services, you can use the sc command, which is a very powerful command that provides even more functionality than the Services.msc snap-in.

What else can you do? Lots. Let's move on now to remote management.

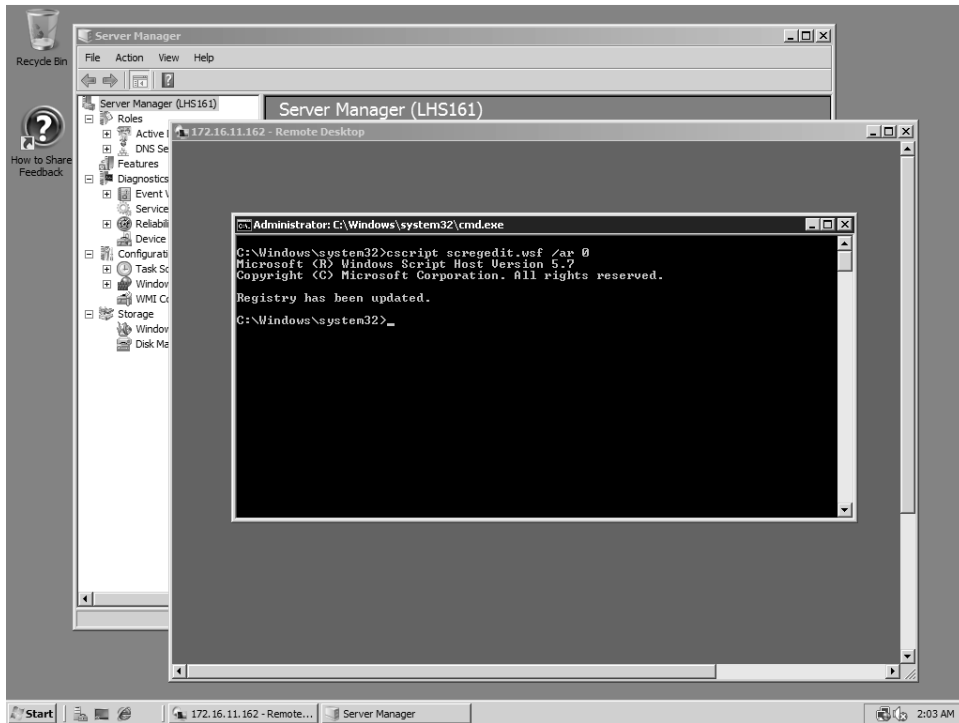
## Remote Management Using Terminal Services

You can also manage Windows server core servers from another computer using Terminal Services. To do this, you first have to enable Remote Desktop on your server, and because we can't right-click on Computer and select Properties to do this, we'll have to find another way. Here's how—use the scregedit.wsf script we looked at previously. The syntax for performing this task is **cscript scregedit.wsf /ar 0** to enable Remote Desktop and **cscript scregedit.wsf /ar 1** to disable it again. To view your current Remote Desktop settings, type **cscript scregedit.wsf /ar /v** at a command prompt. Note that in order to allow pre-Windows Vista



versions of the TS client to connect to a Windows server core installation, you need to disable the enhanced security by running the **cscript scregedit.wsf /cs 0** command.

Once you've enabled Remote Desktop like this, you can connect to your Windows server core server from another machine using Remote Desktop Connection (mstsc.exe) and manage it as if you were logged on interactively at your server's console. In this figure I'm logged on to a full installation of Windows Server 2008 and have a Terminal Services session open to my remote Windows server core server to manage it.



There's more! Later in Chapter 8, "Terminal Services Enhancements," we'll describe a new feature of Terminal Services in Windows Server 2008 that lets you remote individual application windows instead of entire desktops. Let's hear now from one of our experts concerning how this new Terminal Services functionality can be used to make managing Windows server core servers easier.

### **From the Experts: Enabling Remote Command Line Access on Server Core**

There are several ways to administer a Windows server core installation, ranging from using the local console to remote administration from a full Windows Server 2008 server using MMC. A really cool mechanism is to manage the Windows server core installation using Terminal Services RemoteApp to make the command line console available. This allows command-line administration without having to be physically present at the box, and without having a full-blown terminal server session. (After all, a Windows server core installation does not need the full desktop; it just needs the console, and Terminal Services RemoteApp is perfect for this.) A full Windows Server 2008 machine is necessary, along with the Windows server core installation that is to be administered.

On the Windows Server 2008 machine, add the Terminal Server Role using the Server Manager administrative tool. Only the Terminal Server role itself is needed, not the TS Licensing role, TS Session Broker role, TS Gateway role, or TS Web Access role. After the TS role is installed, start MMC and add the TS RemoteApp Manager snap-in, providing the name of the Windows server core machine to the snap-in. Once the snap-in is installed, connect to the Windows server core machine and click Add Remote Apps. Navigate to the %SYSTEMROOT%\System32 folder using the administrative share, select cmd.exe, and complete the wizard. Select the cmd.exe entry in the RemoteApp pane, click Create .rdp File, and follow the wizard to save the RDP file. Ensure that TS is enabled on the Windows server core machine. (Use the scregedit.wsf script.) You can now copy the RDP file to any client machine and connect to the Windows server core installation through it. The console will be integrated into the task bar of the client, like a local application. For more information on Terminal Services and TS RemoteApp, please see Chapter, “Terminal Services Enhancements.”

–Rahul Prasad

*Software Development Engineer, Windows Core Operating System Division*

And here's another expert from the product team at Microsoft sharing some additional tips on managing Windows server core servers using Terminal Services:

### **From the Experts: Tips for Using Terminal Services with Windows Server Core**

When you're using Terminal Services in a Windows server core server without the GUI shell, some common tasks require you to do things a little differently.

#### **Logging off of a Terminal Services Session**

On a Windows server core server, there is no Start button and therefore no GUI option to log off. Clicking the X in the corner of the Terminal Services window disconnects your

session, but the session will still be using resources on the server. To log off, you need to use the Terminal Services logoff command. While in your Terminal Services session, you simply run logoff. If you disconnect your session, you can either reconnect and use logoff, use the logoff command remotely, or use the Terminal Services MMC to log off the session.

### **Restarting the Command Prompt**

When logged on locally, if you accidentally close the command prompt you can either log off and log on, or press CTRL+ALT+DEL, start Task Manager (or just press CTRL+SHIFT+ESC), click file, and run cmd.exe to restart it. You can also configure the Terminal Services client to have the Windows keys pass to the remote session when not maximized so that you can use CTRL+SHIFT+ESC to start task manager and run cmd.exe.

### **Working with Terminal Services Sessions**

If you ever need to manage Terminal Services sessions from the command line, the query command is the tool to use. Running query sessions (which can also be used remotely) will tell you what Terminal Services sessions are active on the box, as well as who is logged in to them. This is handy if you need to restart the box and want to know if any other administrators are logged on. Query has some other useful options, and there are a variety of other Terminal Services command-line tools.

—Andrew Mason

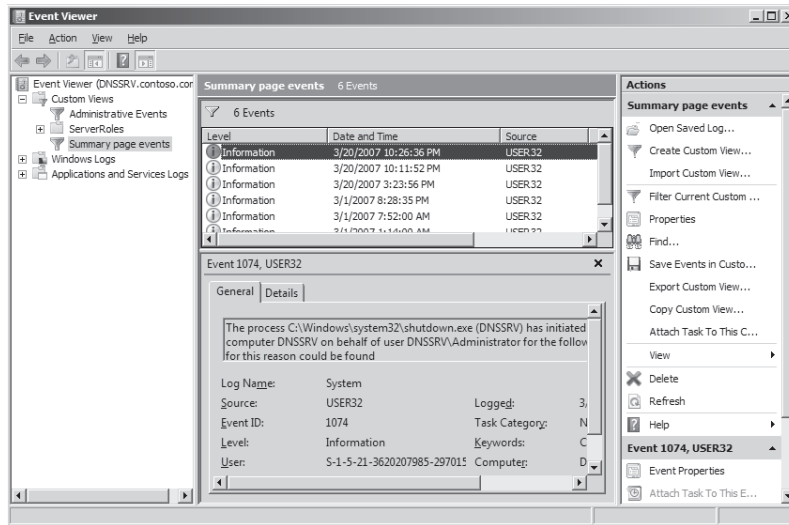
*Program Manager, Windows Server*

## **Remote Management Using the Remote Server Administration Tools**

Although you can manage file systems, event logs, performance logs, device drivers, and other aspects from the command line, there's no law that says you have to. For example, the syntax for wvetutil.exe is quite complex to learn and understand, especially if you want to use this tool to query event logs for specific types of events. It would be nice if you could just use Event Viewer to display, query, and filter your event logs on a Windows server core server. You can! But you have to do it remotely from another computer running either Windows Vista or Windows Server 2008 and with the appropriate Remote Server Administration Tools (RSAT) installed on it.

We talked about RSAT earlier in Chapter 4, “Managing Windows Server 2008,” and it's basically the Windows Server 2008 equivalent of the Adminpak.msi server tools on previous versions of Windows Server. So if you want to use MMC snap-in tools to administer a Windows server core server from a Windows Vista computer or a machine running a full installation of Windows Server 2008, you might or might not need to install the RSAT on this machine because both Windows Vista and full installations of Windows Server 2008 already include many MMC snap-in tools that can be accessed from the Start menu using Administrative

Tools. Event Viewer is one such built-in tool, and here it is running on a full installation of Windows Server 2008, showing the previously mentioned shutdown event in the System event log on our remote Windows server core server.



## Remote Administration Using Group Policy

Another way of remotely administering Windows server core servers is by using Group Policy. For example, although the **netsh advfirewall** context commands can be used to configure Windows Firewall, doing it this way can be tedious. It's much easier to use the following policy setting:

Computer Configuration\Windows Settings\Security Settings\Windows Firewall With Advanced Security

By creating a GPO that targets your Windows server core servers, either by placing these servers in an OU and linking the GPO to that OU or by using a WMI filter to target the GPO only at Windows server core servers, you can remotely configure Windows Firewall on these machines using Group Policy. For example, you can use the *OperatingSystemSKU* property of the *Win32\_OperatingSystem* WMI class to determine whether a given system is running a Windows server core installation of Windows Server 2008 by checking for the following return values:

- 12 – Datacenter Server Core Edition
- 13 – Standard Server Core Edition
- 14 – Enterprise Server Core Edition

You can use this property in creating a WMI filter that causes a GPO to target only Windows server core servers.

## Remote Management Using WinRM/WinRS

Finally, you can also manage Windows server core servers remotely using the Windows Remote Shell (WinRS) included in Windows Vista and the full installation of Windows Server 2008. WinRS uses Windows Remote Management (WinRM), which is Microsoft's implementation of the WS-Management protocol developed by the Desktop Management Task Force (DMTF). WinRM was first included in Windows Server 2003 R2 and has been enhanced in Windows Vista and Windows Server 2008.

To use the Windows Remote Shell to manage a Windows server core server, log on to the Windows server core server you want to remotely manage and type **WinRM quickconfig** at the command prompt to create a WinRM listener on the machine:

```
C:\Windows\System32>WinRM quickconfig
WinRM is not set up to allow remote access to this machine for management.
The following changes must be made:

Create a WinRM listener on HTTP://* to accept WS-Man requests to any IP on this
machine.

Make these changes [y/n]? y

WinRM has been updated for remote management.

Created a WinRM listener on HTTP://* to accept WS-Man requests to any IP on this
machine.
```

Now on a different machine running either Windows Vista or the full installation of Windows Server 2008, type **winrs -r:<server\_name> <command>**, where <server\_name> is your Windows server core server and <command> is the command you want to execute on your remote server. Here's an example of the Windows Remote Shell at work:

```
C:\Users\Administrator>winrs -r:DNSSRV "cscript C:\Windows\System32\slmgr.vbs -dli"
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

Name: Windows(TM) Server Windows Server 2008, ServerEnterpriseCore edition
Description: Windows Operating System - Windows Server 2008, RETAIL channel
Partial Product Key: XHKDR
License Status: Licensed
```

You can also run WinRM quickconfig during unattended installation by configuring the appropriate answer file setting for this service.

# Windows Server Core Installation Tips and Tricks

Finally, let's conclude this chapter with a list of 101 things (well, not really 101) you might want to know about or do with a Windows server core installation of Windows Server 2008. Some of these are tips or tricks for configuring or managing a Windows server core server; others are just things you might want to make note of. They're all either interesting, useful, or both. Here goes....

First, if you want quick examples of a whole lot of administrative tasks you can perform from the command line, just type **cscript scregedit.wsf /cli** at the command prompt:

```
C:\Windows\System32>cscript scregedit.wsf /cli
Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation. All rights reserved.

To activate:
    Cscript slmgr.vbs -ato

To use KMS volume licensing for activation:
    Configure KMS volume licensing:
        cscript slmgr.vbs -ipk [volume license key]
    Activate KMS licensing

        cscript slmgr.vbs -ato
    Set KMS DNS SRV record
        cscript slmgr.vbs -skma [KMS FQDN]
Determine the computer name, any of the following:
    Set c

    Ipconfig /all
    Systeminfo

Rename the Server Core computer:
    Domain joined:
        Netdom renamecomputer %computername% /NewName:new-name
        /UserD:domain-username /PasswordD:*

    Not domain joined:
        Netdom renamecomputer %computername% /NewName:new-name

Changing workgroups:
    Wmic computersystem where name="%computername%" call
    joindomainorworkgroup name="[new workgroup name]"

Install a role or optional feature:
    Start /w Ocsetup [packagename]

    Note: For Active Directory, run Dcpromo with an answer file.
View role and optional feature package names and current installation state:
    oclist
Start task manager hot-key:
    ctrl-shift-esc
```

```

Logoff of a Terminal Services session:
    Logoff

To set the pagefile size:
    Disable system pagefile management:
        wmic computersystem where name="%computername%" set
        AutomaticManagedPagefile=False

    Configure the pagefile:
        wmic pagefileset where name="C:\\pagefile.sys" set
        InitialSize=500,MaximumSize=1000
Configure the timezone, date, or time:

    control timedate.cpl
Configure regional and language options:

    control intl.cpl
Manually install a management tool or agent:
    Msiexec.exe /i [msipackage]

List installed msi applications:
    Wmic

    product
Uninstall msi applications:

    Wmic product get name /value
    Wmic product where name="[name]" call uninstall
To list installed drivers:
    Sc query type= driver
Install a driver that is not included:
    Copy the driver files to Server Core
    Pnputil -i -a [path]\\[driver].inf
Determine a file's version:
    wmic datafile where name="d:\\windows\\system32\\ntdll.dll" get version
List of installed patches:
    wmic qfe list
Install a patch:
    Wusa.exe [patchame].msu /quiet
Configure a proxy:
    Netsh winhttp proxy set [proxy_name]:[port]
Add, delete, query a Registry value:
    reg.exe add /?
    reg.exe delete /?
    reg.exe query /?

```

Now here are a bunch of random insights into and tips for running a Windows server core installation of Windows Server 2008:

The SMS 2005 and MOM 2005 agents should run fine on Windows server core servers, but for best systems management functionality you probably want to use the upcoming Microsoft System Center family of products instead.

You can deploy the Windows server core installation option using Windows Deployment Services (WDS) just like the full installation option of Windows Server 2008. It's the same product—just a different setup option to choose.

To install the Windows server core installation option on a system, the system needs a minimum of 512 MB RAM. That's not because Windows server core servers need that much RAM, however—in fact, they need just over 100 MB of RAM to run with no roles installed. But the setup program for installing Windows Server 2008 requires 512 MB or more of memory or setup will fail. You *can* install the Windows server core installation option on a box with 512 MB RAM and then after installation pull some of the RAM, but at the time of this writing, this procedure is not supported.

The Windows server core installation option uses much less disk space than a full installation of Windows Server 2008. We're talking roughly 1 MB vs. 5 MB here, and that shows you how much stuff has been pulled out of Windows server core to slim it down.

When patching Windows server core servers, you actually don't need to presort patches into those that apply to the Windows server core installation option and those that don't apply. Instead, you can just go ahead and patch, and only updates that apply to Windows server core servers will actually be applied.

You can manage Windows server core servers remotely using the RSAT, but you can't install the RSAT on Windows server core to manage the server locally.

The Windows server core installation option does support Read Only Domain Controllers (RO DC). This support makes Windows server core servers ideal for branch office scenarios, especially with BitLocker installed as well.

You won't get any User Account Control (UAC) prompts if you log on to a Windows server core server as a nonadministrator and try to perform an administrative task. Why not? UAC needs the desktop shell to function.

One way of seeing how slimmed-down Windows server core is is to compare the number of installed and running services on the two platforms. Table 6-3 shows a rough comparison, assuming no roles have been installed.

**Table 6-3 Comparison of default number of services for server core installation vs. full installation**

Feature compared	Server core	Server
Number of services installed by default	~40	~75
Number of services running by default	~30	~50

If you're trying to run the Windows Remote Shell from another machine and use it to manage a Windows server core server and it doesn't work, you might not have the right credentials on the Windows server core server to manage it. If this is the case, first try connecting to the



Windows server core server from your machine using the **net use \\<server\_name>\ipc\$ /u:<domain>\<user\_name>** command using a user account that has local admin privileges on the Windows server core server. Then try running your WinRS commands again. Note that this tip also applies to using MMC admin tools to remotely manage a Windows server core installation since the MMC doesn't let you specify different credentials for connecting remotely.

If you're trying to use Computer Management on another machine to manage the disk subsystem on your Windows server core server using Disk Management and you can't, type **net start vds** at the command prompt on your Windows server core server to start the Virtual Disk Service on the server. Then you should be able to manage your server's disks remotely using Disk Management.

If you've enabled Automatic Updates on your Windows server core server and you want to check for new software updates immediately, type **wuauctl /detectnow** at the command prompt.

And yes, the Windows server core installation option does support clustering. A clustered file server running on Windows server core servers would be cool.

Our last tip will be provided by one of our experts:

### From the Experts: What Time Is It?

Here is a flash back to the old MS-DOS days. Because Windows server core does not have the system tray, there is no clock. If you are used to having the time available on the screen, you can add it to your prompt in the command prompt window.

Entering the following:

```
prompt [%t]%%p$g
```

will display:

```
[14:27:06.28] C:\users\default>
```

*-Andrew Mason*

*Program Manager, Windows Server*

## Conclusion

We're used to Microsoft piling features into products, not stripping features out of them. The Windows server core installation option of Windows Server 2008 is a new direction Microsoft is pursuing in its core product line, but it's a direction being driven by customer demand. When I said that Microsoft listened to their customers, I was serious. And Windows server core is a good example of this.

## Additional Resources

You'll find a brief description of the Windows server core installation of Windows Server 2008 at [http://www.microsoft.com/windowsserver/Windows Server 2008/evaluation/overview.msp](http://www.microsoft.com/windowsserver/Windows%20Server%202008/evaluation/overview.msp). By the time you read this chapter, this page will probably be expanded or the URL will redirect you to somewhere that has a lot more content on the subject.

If you have access to the Windows Server 2008 beta program on Microsoft Connect (<http://connect.microsoft.com>), you can get some great documentation from there, including these:

- Microsoft Windows Server Code Name 2008 Server Core Step-By-Step Guide
- Live Meeting on Server Core
- Live Chat on Server Core

There's also a TechNet Forum where you can ask questions and help others trying out the Windows server core installation option of Windows Server 2008. See <http://forums.microsoft.com/TechNet/ShowForum.aspx?ForumID=582&SiteID=17> for this forum. (Windows Live registration is required.)

There's a Windows server core blog on TechNet that is definitely something you won't want to miss. See [http://blogs.technet.com/server\\_core/](http://blogs.technet.com/server_core/).

Finally, be sure to turn to Chapter 14, "Additional Resources," for more sources of information concerning the Windows server core installation option, and also for links to webcasts, whitepapers, blogs, newsgroups, and other sources of information about all aspects of Windows Server 2008.