Set up, configure, and manage Microsoft Windows Server 2008

Provide feature-rich web and FTP services with IIS 7.0

Use the all-new Server Manager, PowerShell, and Deployment Services

Optimize availability, scalability, security, and efficiency

Steve Seguis
Praise for *Microsoft Windows Server 2008 Administration*

Steve Seguis’ *Microsoft Windows Server 2008 Administration* is a wonderful read by a brilliant and skillful writer. The book is written in concise and easy-to-understand terms that will benefit both new and experienced administrators.

The book includes hands-on exercises, chapter summaries, and plenty of images. The hands-on exercises allow you to put into practice what you have just learned or read. The exercises are written in a step-by-step manner so that you can perform the tasks at hand without the need to reread the accompanying text. The chapter summaries are brief chapter overviews and are a handy way to refresh your memory about the contents of the chapter. The images that accompany the book are great for seeing where you need to be when reading the content.

I recommend this book in part because of the new improvements and enhancements that Microsoft has added to their flagship Server Operating System. I also recommend this book because it will make a great addition to your technical library.

---Don Hite, Microsoft MVP, Systems Management Server, IBM Global Services

If you’re a professional Windows Server administrator, this book is a must-have. The hands-on exercises alone set this book apart from any other Windows Server management guide I’ve read in a long time. You can tell that Steve has spent a great deal of time with Windows Server 2008. I highly recommend it.

---Stuart B. Renes, Microsoft MVP, Windows Server System

Whether you are new to Windows Server 2008 or not, this book will give you the background to understand the new technologies and get you up to speed quickly. Although I primarily work with small to medium businesses, this book will serve me equally well in these smaller environments as well as the larger enterprise environments. An excellent reference for anyone!

---Kevin Royalty, MCSE 2000/2003, Microsoft MVP, Small Business Server Managing Partner, Total Care Computer Consulting
This page intentionally left blank
ABOUT THE AUTHOR

Steve Seguis is a Windows Systems Engineer in the financial industry who has been managing Microsoft Windows environments for more than 10 years. He was a Microsoft Most Valuable Professional (MVP) for Windows Server Admin Frameworks from 2004 to 2007, and is a contributing writer and technical editor for Scripting Pro VIP (formerly Windows Scripting Solutions) magazine. His specialty is in systems management and automation.

About the Technical Editor

Richard Lewis is a Windows Systems Engineer who has been involved in Windows systems design and automation for more than 11 years and is currently a consultant to the aerospace industry at Lewis Technology (www.lewistech.com). He has been a Microsoft Certified Systems Engineer (MCSE) and Microsoft Certified Trainer (MCT) since 1996 and is a contributing author and technical editor for Windows ITPro magazine and Scripting Pro VIP. Richard has penned more than 200 articles on Windows training, administration, scripting, and system automation.
We hope you enjoy this McGraw-Hill eBook! If you’d like more information about this book, its author, or related books and websites, please click here.
For my wife Annalene who never fails to support and believe in me!
This page intentionally left blank
AT A GLANCE

▼ 1  Getting Started with Windows Server 2008  ...  1
▼ 2  Server Core  .................................. 25
▼ 3  Server Manager  ................................. 51
▼ 4  Active Directory Domain Services .......... 95
▼ 5  Windows Deployment Services ............... 145
▼ 6  Internet Information Services 7.0 .......... 177
▼ 7  Resource Management and Performance Monitoring .................................. 213
▼ 8  Network Policy and Access Services ......... 253
▼ 9  Terminal Services  ............................... 285
▼ 10 Windows DNS, BitLocker Drive Encryption, and Itanium Support .................. 331
▼ 11 Routing and Remote Access ................... 353
▼ 12 Enterprise Public Key Infrastructure ........ 401
▼ 13 Windows PowerShell ......................... 433

▼  Index  ............................................ 459
This page intentionally left blank
CONTENTS

Acknowledgments ........................................... xvi
Introduction .................................................... xix

1 Getting Started with Windows Server 2008 ............... 1
   System Requirements ........................................ 2
   Installation and Configuration ......................... 3
   Post-Installation Configuration and Initial
      Configuration Tasks .................................. 8
   Boot Configuration Data .................................. 10
   BCD Store .................................................. 10
   BCD Object ................................................. 13
   BCD Elements ............................................. 16
   BCD Modification Methods .............................. 16
   Chapter Summary ......................................... 23

2 Server Core .................................................. 25
   Roles Supported by Server Core ..................... 26
   The Ups and Downs of Server Core ................. 27
   Installing Server Core ................................ 27
      Requirements ......................................... 27
      Post-Installation Tasks ............................ 30
5 Windows Deployment Services ........................................... 145
Benefits of Using Windows Deployment Services .................. 146
Scenarios for Windows Deployment Services ......................... 147
Components ............................................................... 148
WDS Installation ......................................................... 148
WDS Properties ......................................................... 151
Creating an Operating System Image for WDS ....................... 152
Loading Your Install Image to Your Clients Using WDS ........... 162
Unattended Install Using WDS .......................................... 164
Windows System Image Manager ....................................... 165
Chapter Summary ...................................................... 174

6 Internet Information Services 7.0 ................................. 177
IIS 7.0 Features .......................................................... 178
Unattended Installation .................................................. 181
IIS Management Console ............................................... 187
Remote IIS Administration ............................................. 192
Administration Using APPCMD.EXE ................................ 194
Delegated Administration ................................................ 200
Server and Application Health and Performance ................. 204
  Runtime Status & Control API ..................................... 204
  Automatic Failed Request Tracing ................................ 205
Xcopy Deployment ....................................................... 211
Chapter Summary ...................................................... 212

7 Resource Management and Performance Monitoring .............. 213
Data Is Good! ............................................................ 214
Windows System Resource Manager .................................. 215
  WSRM Architecture ................................................ 215
  Managed vs. Unmanaged Processes ................................ 216
  WSRM Service ....................................................... 216
  The WSRM Management Interface ................................. 218
Process Matching Criteria ............................................ 219
Resource Allocation Policies .......................................... 222
Calendar ................................................................. 228
Accounting ............................................................... 231
Conditions .............................................................. 235
Resource Monitor ...................................................... 236
Reliability and Performance Monitor ................................ 239
  Data Collector Sets .................................................. 242
  Reliability Monitor ................................................ 246
  Reports ............................................................... 248
Chapter Summary ...................................................... 252
8 Network Policy and Access Services .......................... 253
   Network Access Protection .................................. 254
   NAP Components ........................................... 256
       IPSec Enforcement ....................................... 256
       802.1X Enforcement ..................................... 257
       VPN Enforcement ....................................... 257
       DHCP Enforcement ...................................... 258
   Network Policy Server/RADIUS ................................ 258
   NAP Agent .................................................... 258
   System Health Agent ....................................... 258
   NAP Administration Server .................................. 258
   System Health Validator .................................... 259
   Health Policy .............................................. 259
   Accounts Database ......................................... 259
   Health Registration Authority .............................. 259
   Remediation Server ........................................ 259
   Dispelling NAP Myths ....................................... 259
   Architecture ............................................... 260
   NAP Client Architecture ................................... 261
       Enforcement Clients ..................................... 262
       System Health Agent ..................................... 262
   NAP Server Architecture ................................... 262
       Enforcement Servers .................................... 263
   Communications Flow ....................................... 263
       Requirements ............................................ 265
       Preparation .............................................. 265
       Installing the Network Policy Server ................... 265
       Configuring the Network Policy Server .................. 266
       Installing and Configuring DHCP ....................... 271
       Configuring the Client ................................... 281
   Testing the NAP Client ...................................... 283
   Chapter Summary ............................................ 284

9 Terminal Services .............................................. 285
   Terminal Services Core Functionality ...................... 286
       Remote Desktop Connection 6.0 ........................ 287
   Single Sign-On .............................................. 287
   Installing Terminal Services ................................ 291
   Terminal Services Licensing ................................ 294
       License Types ............................................ 294
       Installing and Configuring TS Licensing ................ 295
   Terminal Services Gateway .................................. 302
       TS Gateway Architecture ............................... 302
       TS Gateway and NAP ..................................... 317
Terminal Services Remote Programs ........................................... 317
Requirements ........................................................................... 318
Installing Applications ............................................................. 318
Terminal Server Web Access ..................................................... 323
Program Placement and Performance ....................................... 329
Chapter Summary .................................................................... 330

10 Windows DNS, BitLocker Drive Encryption, and Itanium Support .... 331
Domain Name System ................................................................. 332
  Background Zone Loading ......................................................... 333
  IPv6 Support .......................................................................... 334
  GlobalNames Zone .................................................................. 334
  Read-Only DNS Zone ............................................................... 334
  Windows Link-Local Multicast Name Resolution ....................... 335
Windows BitLocker Drive Encryption ........................................... 336
Requirements ........................................................................... 336
BitLocker Architecture ............................................................... 337
Initializing BitLocker .................................................................. 344
BitLocker Recovery .................................................................... 350
Turning Off or Uninstalling BitLocker Drive Encryption .................. 351
Windows Server 2008 Itanium Support ........................................ 351
Chapter Summary .................................................................... 352

11 Routing and Remote Access .................................................... 353
Routing Services ....................................................................... 354
  Routing Basics ......................................................................... 354
  Dynamic Routing ...................................................................... 356
  Routing Configuration with RRAS ............................................. 358
  Configuring Network Interfaces for Routing ............................... 359
Routing Protocols ....................................................................... 361
Remote Access ........................................................................... 381
Dial-Up Networking .................................................................... 381
Virtual Private Networks .............................................................. 383
DHCP Integration with RRAS ..................................................... 389
Configuring RRAS Server Properties ........................................... 389
Chapter Summary .................................................................... 398

12 Enterprise Public Key Infrastructure ........................................ 401
PKI Uses ................................................................................... 402
Digital Signatures ....................................................................... 403
Digital Certificates ...................................................................... 404
Certification Authorities .............................................................. 404
Types of CAs ............................................................................. 405
  Enterprise CAs ........................................................................ 405
  Stand-alone CAs ...................................................................... 405
This book wouldn’t exist without the concerted efforts of many individuals working together from different disciplines, who made sure that the final product is something of which we can all be proud. First off, I want to thank my literary agent, David Fugate, who initially approached me to ask if I would be interested in putting together a proposal for this book project. He opened up the door for me to write my first book.

Jane Brownlow was the sponsoring editor for this book and came up with the initial concept, got the publisher’s approval, and got the ball rolling. Jane took maternity leave shortly after I started writing this book, so Megg Morin (acquisitions editor) and Carly Stapleton (acquisitions coordinator) kept this project going, making sure we stayed focused on meeting our objectives and promptly answering any questions I had. After Jane returned from maternity leave (Congratulations, Jane!), they all worked together to help me finish up the book. Lisa Theobald was the copy editor for this book and together with Janet Walden, the editorial supervisor, put a lot of work into making my writing much clearer and making me look better in the process. I want to thank them all very much for their professionalism and dedication to this project.
Richard Lewis was the book’s technical editor, and he painstakingly went through several iterations of each chapter as I worked through writing various lab exercises to ensure technical accuracy of both the general content and the hands-on exercises. He also provided lots of good feedback that I believe helped improve the book tremendously. Thanks for paying attention to the details. The effort you put into this project, especially toward the end to ensure that we hit our deadlines, is very much appreciated.

Finally, many people didn’t directly participate in the writing of this book but were directly impacted during the time of its writing, and those people are my family. As is the case with most technical writers, I have a regular full-time day job in addition to writing this book. I want to thank my family for being patient and understanding while I spent countless hours night after night, weekend after weekend, month after month, locked away in my lab painstakingly researching and writing instead of spending quality time with them. More specifically, I would like to thank my wife, Annalene, for understanding why I was too busy for the past few months to spend quality time with her and take her to the movies, and for understanding why we had to reschedule every vacation we had planned for so long just to get this book done. She’s always believed in me and has stood by every decision I’ve made in my career. Thanks for being my best friend! I also want to thank my parents, Romeo and Lourdes Seguis, for being great role models, raising me with a good head on my shoulders, and giving me opportunities that helped shape my career and my life. I love you all very much!
have read hundreds of books throughout my career, as I’m sure many of you have, and I’ve found three general categories of technical books: On one end of the spectrum are books geared toward beginners that help readers get a basic understanding of each topic but are only skin deep. On the other extreme are highly technical reference books that try to cover every imaginable aspect of the subject (but typically fail to do so). Those types of books go into great detail about every subject, but—let’s face it—there’s no such thing as a book that covers absolutely everything. Those books in the middle of the spectrum cover the basics regarding things you should know, but go into greater detail about things you really need to know. This book was purposely written to be more of a book in the middle, and I’ll tell you why.

While I consider myself to be highly technical, I don’t like more complicated explanations than are necessary. This has been my approach while writing this book. My goal was to write a book that satisfies your need for technical details without making your head spin in the process. This book is clearly targeted to professionals, so I have made the assumptions that you already have a healthy understanding of servers and how they work and have managed a Microsoft Windows Server–based operating system in the past (even better if you are currently doing so).
In each chapter, I start off with a few basics on each topic, and in some cases a quick review of the subject matter, before diving into specifics of how things work in Windows Server 2008. I hope this greatly enhances the reader experience, since it makes sure that every reader is on the same page (no pun intended) before going into product-specific information.

You will also notice that I use plenty of hands-on exercises throughout each chapter. I think that understanding theory and general concepts is a good thing, but most people learn best while actually completing tasks. I hope that you will find the inclusion of many hands-on exercises to be of use to you. One of the major goals of these exercises is to force you to use Windows Server 2008 and its many features. Although each exercise offers step-by-step instructions on how to accomplish a specific task, there is always more than one way to perform a task, so feel free to experiment and try to find other ways to work. One thing you will appreciate with Windows Server 2008 is the flexibility it offers you as an administrator to interact with various elements of the operating system. Take advantage of this and don’t assume that the way I wrote it is necessarily the best way, since I sometimes had to choose steps that were easier to follow rather than faster to do.

I also don’t hold back on screenshots. These are not page fillers but serve a specific purpose of showing what you can expect the screens to look like as you work through the exercises. I can’t tell you how many times I’ve read a book and scratched my head while reading some of the step-by-step guides because either the description wasn’t clear or a miscommunication was written about what I should be looking at versus what I actually saw. By providing the screenshots, I hope to clear up a lot of the confusion associated with many of those purely text-based exercises.

This book was initially written when Microsoft Windows Server 2008 (then called Windows Server codename “Longhorn”) was still in Beta 2. As you can very well understand, Beta 2, which wasn’t made available to the general public, was still quite rough around the edges, and many features and graphical elements didn’t function the way one might expect. I finished writing the first draft of the book just as Release Candidate 1 was released to the general public. After Windows Server 2008 Release Candidate 1 was made available, we went back and updated every chapter and making changes where appropriate; we recaptured all of the screenshots since Microsoft had thankfully done a wonderful job polishing up the user interface and in many cases fixed major bugs that caused me many sleepless nights. We did our very best to make sure that you got the most accurate information you can get up until product launch so as you read this book, please keep in mind that the screenshots and exercises were taken from Windows Server 2008 Release Candidate 1, and while Microsoft generally doesn’t make any major functionality changes other than bug fixes prior to launch, the screenshots and some of the wording on the screen can potentially be different from that of the final product.

This book, *Microsoft Windows Server 2008 Administration*, is a book written by a Windows administrator for Windows administrators. I know how frustrating it is to read a book and not be able to answer the question of “How do I do that?”. From the ground up, I focused on one thing and one thing alone, and that is to provide you
with the information you need to not only answer the question “What can do I in Windows Server 2008?”, but also “How do I do that in Windows Server 2008?”. It’s a direct hands-on approach loaded with step-by-step guides and real examples. Unfortunately, there’s no way to do that and cover every possible feature inside this new operating system. However, this book will equip you to make good decisions about how you can use Windows Server 2008 in your environment and take advantage of its many new features.
This page intentionally left blank
CHAPTER 1

Getting Started with Windows Server 2008
When Microsoft started development of Windows Server 2008, the company took the time to collect user feedback and incorporate this information into the product’s features. It is the first operating system built by Microsoft under its new strict security development guidelines. The security “theme” permeates every aspect of this operating system and can’t be missed. Although future system updates are inevitable with any OS release, this new architecture allows you to minimize the attack surface immediately, thereby mitigating the risks. Microsoft has also vastly improved the user experience by simplifying the installation process and providing a new integrated Server Manager tool for more effective server management. Before you can take advantage of any of these features though, your first step is to install Windows Server 2008. Let’s cut to the chase and see what it takes to get Windows Server 2008 installed.

### SYSTEM REQUIREMENTS

To ensure proper installation of Windows Server 2008, you will need to make sure the server hardware meets these minimum and recommended hardware levels:

<table>
<thead>
<tr>
<th>Component</th>
<th>Minimum</th>
<th>Recommended</th>
<th>Optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processor</strong></td>
<td>1GHz</td>
<td>2GHz</td>
<td>3GHz or faster</td>
</tr>
<tr>
<td><em>Intel Itanium 2 processor required for Windows Server 2008 for Itanium-based systems</em></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>512MB RAM</td>
<td>1GB RAM</td>
<td>2GB RAM (Full installation) or 1GB RAM (Server Core installation) or more</td>
</tr>
<tr>
<td><em>Maximum (32-bit): 4GB (Standard) or 64GB (Enterprise and Datacenter)</em></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Maximum (64-bit): 32GB (Standard) or 2TB (Enterprise, Datacenter, and Itanium-based systems)</em></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Disk Space</strong></td>
<td>8GB</td>
<td>40GB (Full installation) or 10GB (Server Core installation)</td>
<td>80GB (Full installation) or 40GB (Server Core installation) or more</td>
</tr>
<tr>
<td><strong>Drive</strong></td>
<td>DVD-ROM drive</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Display</strong></td>
<td>SVGA (800×600) or higher resolution</td>
<td>Keyboard</td>
<td></td>
</tr>
<tr>
<td>Microsoft mouse or compatible pointing device</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
INSTALLATION AND CONFIGURATION

Windows Server 2008 offers two general types of installations: a typical Full server installation and Server Core. Server Core is a stripped down version of Windows Server 2008 that doesn’t include a GUI or any other unneeded services. Instead, the server installs only key features that are related to the role that it supports—for example, Active Directory or Domain Name System (DNS). Chapter 2 provides more details about Server Core. The following paragraphs discuss a typical Windows Server 2008 installation.

One of server engineers’ biggest gripes about the manual Windows Server installation process in the past was that they had to babysit the server as it went through the installation, because they had to key in bits of information at different times throughout the process—license information, components to install, and network configuration, for example. Of course, the easy solution to all this is to perform an unattended installation, but for the one-offs that require manual installation, the process was far from being “set and forget.”

In Windows Server 2008, this problem has been addressed by reducing the number of interactive steps required to get your server up and running. All the necessary questions for the installation are asked up front, before you begin the actual installation process of copying the files and performing the initial server configuration. By doing this, the installation process no longer has to stop for additional information before it can proceed. Once the server software installation is complete, installation of components and the configuration of the server can proceed under the new integrated management tool called Server Manager.

Hands-On Exercise: Interactive Installation of Windows Server 2008

1. Start the computer and bootup using the Windows Server 2008 installation media. Select the installation language, time and currency format, and keyboard layout, and then click Next (Figure 1-1).

2. Click Install Now to begin the installation process. As you can see in Figure 1-2, you can access system recovery tools by clicking the Repair Your Computer option at the bottom of the screen.

3. Enter the product key. If you don’t want to activate Windows as soon as you’re computer goes online (for example, if you are simply testing the installation or evaluating Windows Server 2008), you can uncheck the Automatically Activate Windows When I’m Online checkbox (Figure 1-3). Click Next.

4. Now select whether to install Windows Server 2008 Enterprise (Full Installation) or Windows Server 2008 Enterprise (Core Installation). For now, select Windows Server 2008 Enterprise (Full Installation) (as shown in Figure 1-4), and then click Next.
Figure 1-1. Installation language, time and currency, and keyboard layout screen

Figure 1-2. Installation screen
Chapter 1: Getting Started with Windows Server 2008

Figure 1-3. Product key screen

Figure 1-4. Operating system selection screen
5. If you accept the terms of the license agreement, check the I Accept the License Terms checkbox (required to use Windows), and then click Next (Figure 1-5).

6. Select the type of installation you want to perform. In this case, you are performing a clean install, so you should select Custom (Advanced). You’ll notice that you can’t select Upgrade unless you initiated the setup from an existing Windows Server installation (Figure 1-6).

7. If your hard drive is automatically detected, you can create and format a partition as necessary for the installation. If your drive isn’t detected, most likely the device driver for your controller isn’t built into Windows, in which case you can click Load Driver (at the bottom-left of the screen) to load it. Click Next after you have created the partition to which you are going to install (Figure 1-7).

8. Now that Windows Server 2008 has all the basic information it needs to proceed with the installation, it begins the installation process and displays the status of the install, as shown in Figure 1-8. This is where setup significantly differs from previous Windows Server builds, as you will not be prompted for any further details until the installation is complete and Windows fully starts up. This is a great enhancement, since you can walk away from the server while the installation proceeds without having to worry about additional dialog boxes asking for further information to complete the install.
Chapter 1: Getting Started with Windows Server 2008

Figure 1-6. Installation type selection screen

Figure 1-7. Installation partition selection screen
9. When setup has completed installing Windows and has rebooted as many times as necessary to install and configure everything, you will automatically be logged in to Windows Server 2008 under the Administrator account, where the Initial Configuration Tasks screen is loaded.

**IMPORTANT** By default, the Administrator Password field is blank and should be changed immediately. Until you set a password, Windows Server 2008 will autologon with the Administrator account and a blank password. On the first password change, remember that the old password field is left blank because the password is indeed blank.

**Post-Installation Configuration and Initial Configuration Tasks**

After the installation has completed, you are prompted for the initial configuration tasks (Table 1-1). Many of these options would have typically been part of the initial installation options in previous Windows Server versions—such as setting the administrative password, configuring network options, and specifying computer name and domain membership information.
<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the Administrator Password</td>
<td>Lets you set the password for the Administrator account and rename the account.</td>
</tr>
<tr>
<td>Set Time Zone</td>
<td>Sets the time zone for the server.</td>
</tr>
<tr>
<td>Configure Networking</td>
<td>Opens the Network Connections Control Panel applet so you can configure your various network interfaces.</td>
</tr>
<tr>
<td>Provide Computer Name and Domain</td>
<td>Lets you change the computer name as well as join a domain.</td>
</tr>
<tr>
<td>Enable Automatic Updating and Feedback</td>
<td>Lets you specify how you want to configure Windows Update, Windows Error Reporting, and the Customer Experience Improvement Program (CEIP). You should compare the Windows Error Reporting information as well as the CEIP settings against your organization’s policies, since both features send usage information back to Microsoft.</td>
</tr>
<tr>
<td>Download and Install Updates</td>
<td>Lets you download and install updates. You should do this unless you have an alternative patch-management tool, since you want your system to be up to date with all critical security patches before opening it up to your network. You should manually set the configuration of the updates based on your own policies to prevent updates from automatically restarting your server. You should also keep checking for updates after each reboot until all the updates have been installed.</td>
</tr>
<tr>
<td>Add Roles</td>
<td>Lets you add roles to this server—that is, Dynamic Host Configuration Protocol (DHCP), DNS, Internet Information Services (IIS), and so on.</td>
</tr>
<tr>
<td>Add Features</td>
<td>This new interface replaces the Add/Remove Programs Control Panel applet in previous versions of Windows and provides a much easier means of adding additional Windows components.</td>
</tr>
<tr>
<td>Enable Remote Desktop</td>
<td>Lets you configure remote desktop.</td>
</tr>
<tr>
<td>Configure Windows Firewall</td>
<td>Turns on or turns off the Windows Firewall.</td>
</tr>
</tbody>
</table>
TIP  If you change the administrative password by pressing `CTRL-ALT-DEL` and then select Change a Password on the Change the Password screen below the Confirm Password line, you’ll see a Create a Password Reset Disk selection, which is the entry point to the Welcome to the Forgotten Password Wizard. This same wizard is also available in Control Panel by clicking User Accounts | Prepare for a Forgotten Password. After launching the wizard, you will be prompted to insert a formatted floppy disk, which is used to create a password recovery disk. After this disk is created, it can be used to recover from a forgotten password even if the password has been changed. Consequently, this floppy disk should be physically secured, as it could be used for unauthorized server access.

Once you close out of the Initial Configuration Tasks interface, the Server Manager tool automatically launches. This is an integrated interface you can use to configure various items on your computer. You’ll read details about managing your server using Server Manager in Chapter 3.

**BOOT CONFIGURATION DATA**

All Windows Server builds since Windows NT have been using NT Loader (NTLDR) and `boot.ini` to control the boot process as well as manage multi-boot environments. With Windows Server 2008 (as well as Windows Vista), the entire boot process has been re-engineered, resulting in the creation of the Boot Configuration Data (BCD). The BCD replaces NTLDR completely in its functionality, and, rather than store the boot configuration in a text file such as `boot.ini`, everything is now stored in a binary format that can be manipulated only using one of the following editing methods: `BCDEdit.exe` or coding using Windows Management Interface (WMI).

The BCD is physically stored in one of two locations. For BIOS-based operating systems, the BCD is stored in the `\Boot\BCD` directory of the active partition. For Extensible Firmware Interface (EFI)–based operating systems, the BCD is stored on the EFI system partition (NVRAM). For those of you who may not be familiar with EFI, you’ll see it implemented in 64-bit systems. Currently, these are the only two systems supported by BCD; however, in technical terms, it would be possible for Microsoft to extend the BCD to other boot systems in the future. The internal structure of the BCD is that of a registry hive, which makes sense due to the hierarchal nature of the data being stored there; however, you should never attempt to manipulate the BCD using tools designed for the registry.

The BCD architecture is a hierarchy, which is exactly why it made sense to reuse the registry hive format for this data store. It is composed of three distinct components: stores, objects, and elements, as described in Table 1-2. The component hierarchy is shown in Figure 1-9.

**BCD Store**

The BCD store is the physical binary file that is stored either on the active partition or on the EFI system partition (ESP). It stores all the information that describes the bootup environment for each Windows instance on the system or other boot loaders such as NTLDR.
Each system can have more than one BCD store; however, only one store can be the active system store. A simple example of an additional BCD store would be a backup of the active system store. For BIOS-based systems, this file is stored under the active partition’s \BOOT folder, whereas for EFI-based systems, it is stored under \Windows\Boot\EFI.
Since the system store knows all about the installed operating systems on the computer, if it detects a multi-boot environment, it is also responsible for displaying the Windows Boot Manager OS selection menu, as shown in Figure 1-10. Each system store contains, minimally, two BCD objects as well as additional options (Table 1-3).

Although it all sounds complicated, it really isn’t. You can take apart a simple boot.ini file, such as the one shown here, and translate it quickly to a BCD format (Table 1-4).

```plaintext
[boot loader]
timeout=3
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS=Microsoft windows XP Professional /nologin
```

![Figure 1-10. Windows Boot Manager showing multi-boot screen and Windows Memory Diagnostic option](image-url)
Chapter 1: Getting Started with Windows Server 2008

BCD Object

Each BCD object is identified uniquely using a 128-bit GUID that contains a 32-bit description about the type of object it represents. The three object categories are application objects, inheritable objects, and device objects. The application objects type is the most common type and is the object type for the Windows Boot Manager, Windows boot loader objects including NTLDR, Windows resume loader, and Windows memory tester. Windows resume loader is invoked when you turn on the computer from hibernate mode.

<table>
<thead>
<tr>
<th>BCD Object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Boot Manager</td>
<td>Think of this as the [boot loader] section of the original boot.ini file. It contains things like the default boot OS as well as the timeout before the default OS is launched. The BCD can store multiple Windows Boot Managers, but only one can hold the global unique identifier (GUID) that designates the active boot manager. This GUID is aliased as {bootmgr} and is used in BCDEdit.exe to make changes to the store.</td>
</tr>
<tr>
<td>Windows Boot Loader</td>
<td>The store must contain at least one Windows Boot Loader objects. The Windows Boot Loader contains information regarding the boot environment for each instance of Windows Server 2008 installed on the system. Each boot loader contains a number of BCD elements that describe additional boot parameters such as no-execute, page-protection policies and debugger options. Two special aliases relate to the Windows Boot Loader. The first is called {current} and points to the currently active boot loader. The other is called {default} and points to the default boot loader if nothing is explicitly selected by the user.</td>
</tr>
<tr>
<td>Windows NTLDR</td>
<td>This special object points to the old NTLDR if you have an older Windows installation on the system. This special GUID is referenced by the alias {ntldr}.</td>
</tr>
<tr>
<td>Optional boot applications</td>
<td>These special applications perform other boot-related tasks. For example, Windows Server 2008 includes a Windows Memory Diagnostic tool, an optional boot application used to perform various memory checks on the system.</td>
</tr>
</tbody>
</table>

Table 1-3. BCD Objects
Each application object contains an image type and an application type. The image type tells the system whether it should be loaded as a firmware, boot, NTLDR-based, or real-mode application. The application type is a bit more detailed on what the application does. The most common application types are listed on the next page and in Table 1-5.

<table>
<thead>
<tr>
<th>Description</th>
<th>Alias</th>
<th>GUID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Boot Manager</td>
<td>{bootmgr}</td>
<td>9dea862c-5cdd-4e70-acc1-f32b344d4795</td>
</tr>
<tr>
<td>Firmware Boot Manager</td>
<td>{fwbootmgr}</td>
<td>a5a30fa2-3d06-4e9f-b5f4-a01df9d1fcba</td>
</tr>
<tr>
<td>Windows memory tester</td>
<td>{memdiag}</td>
<td>b2721d73-1db4-4c62-bf78-c548a880142d</td>
</tr>
<tr>
<td>Windows resume application</td>
<td>None</td>
<td>147aa509-0358-4473-b83b-d950dda0615</td>
</tr>
<tr>
<td>Legacy Windows Loader</td>
<td>{ntldr}</td>
<td>466f5a88-0af2-4f76-9038-095b170dc21c</td>
</tr>
<tr>
<td>Current boot entry</td>
<td>{current}</td>
<td>fa926493-6f1c-4193-a414-58f0b2456d1e</td>
</tr>
<tr>
<td>Default boot entry</td>
<td>{default}</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 1-5. Most Commonly Used Application Objects’ Aliases and GUIDs
Firmware Boot Manager (for EFI-based systems)
■ Windows Boot Manager
■ Windows boot loader
■ Windows resume application
■ Windows memory tester
■ NT Loader
▲ Boot sector (can be used to load non-Windows-based systems)

BCD inheritable objects are a way to generalize certain settings and flags so that they can be reused in more than one BCD object. Rather than having separate instances of an object, it is globally defined and then referenced by other BCD objects as needed. Some examples of these inheritable objects are listed in Table 1-6.

<table>
<thead>
<tr>
<th>Alias</th>
<th>GUID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{badmemory}</td>
<td>5189b25c-5558-4bf2-bca4-289b11bd29e2</td>
<td>Global RAM defect list</td>
</tr>
<tr>
<td>{bootloadersettings}</td>
<td>6efb52bf-1766-41db-a6b3-0ee5eff72bd7</td>
<td>Settings that should be inherited by all Windows boot loaders</td>
</tr>
<tr>
<td>{dbgsettings}</td>
<td>4636856e-540f-4170-a130-a84776f4c654</td>
<td>Debugger settings that can be inherited by any boot application</td>
</tr>
<tr>
<td>{emssettings}</td>
<td>0ce4991b-e6b3-4b16-b23c-5e0d9250e5d9</td>
<td>Emergency Management Services settings that can be inherited by any boot application</td>
</tr>
<tr>
<td>{globalsettings}</td>
<td>7ea2e1ac-2e61-4728-aaa3-896d90a9f0e</td>
<td>Settings that should be inherited by all boot applications</td>
</tr>
<tr>
<td>{resumeloadersettings}</td>
<td>1afa9c49-16ab-4a5c-901b-212802da9460</td>
<td>Settings that should be inherited by all resume applications</td>
</tr>
</tbody>
</table>

Table 1-6. Examples of Inheritable Objects
As you can tell from the sample list, the objects are typically general global settings that propagate to multiple objects. In addition to this, each inheritable object is classified under two classes: library class and application class. Library class inheritable objects can be inherited by any BCD object, whereas application class inheritable objects can be inherited only by specified BCD applications.

BCD device objects contain BCD elements for complex devices, unlike simple devices such as partitions, which can be defined as simple BCD elements. BCD device objects are most commonly used for describing booting RAM disks created from Windows Image (WIM) files, as this type of device type can contain the location of the WIM file in addition to any relevant port information if loaded from the network.

**BCD Elements**

Unlike the older boot.ini system, BCD elements have distinct data types associated with the data values. For example, an element can contain a String, Object, Integer, or Boolean data type. In addition to this, BCD elements are limited by their class type. Library elements can be applied to all boot environment applications; application elements can be applied only to specific application class types; and device elements can be applied only to device objects.

**BCD Modification Methods**

As fun as it was describing the BCD architecture and explaining the technical nuances of each component, I’ll bet you have this burning question in your mind. How do I actually manipulate the BCD? You can manipulate the BCD in four ways, as shown in Table 1-7.

**Using BCDEdit**

Since this tool is critical to the manipulation of BCD data, you should take the time to understand it. As with all command-line tools, the best way to learn about available command switches and general functionality is by running it with the /? switch to display the help screen for the command and, in this case, the primary switches the tool supports. If you want to get into more specifics about a particular command-line switch, you can type in `BCDEdit.exe /? <command>` where `command` is any of the available switches. For example, if you want to learn more about the `export` switch, you can type this:

`BCDEdit.exe /? /export`

The most basic command you’ll need to know lets you retrieve your current configuration:

`BCDEdit.exe /enum`

This command shows your global Windows Boot Manager settings along with settings associated with each of your Windows OS Loaders. You can see the output of this command on a dual-boot Windows Server 2003 and Windows Server 2008 computer in Figure 1-11. You can clearly see the display order for the menu items, the default boot
loader, and the timeout. For each of the boot loaders, you can see their unique identifier, device path, and any options (BCD elements) that have been specified.

You can specify additional parameters with the /enum switch to control what is displayed, such as displaying only the Windows Boot Manager section or getting information about a particular boot loader. One of the most useful additional switches to /enum is the /v switch. This switch shows all entry identifiers in full GUID form rather than their user-friendly aliases. The identifiers are in GUID format—for example, {0f732d04-e6b2-11da-b631-b722247cd703}. The aliases are those values in the output that are enclosed in curly braces that are not GUIDs—that is, {ntldr}, {current}, {bootmgd}, and so on. As an additional shortcut, if you simply run BCDEdit.exe without any switches, it defaults to running the following:

```
BCDEdit.exe /enum ACTIVE
```
The most common changes most administrators will make to the BCD will be around the Windows Boot Manager, since that controls the boot sequence, default Windows loader, display order, and timeout before the default selection is made. The help messages give you all the information you’ll ever need, but it’s much easier to understand this command by looking at some simple examples.

**Modifying the Boot Sequence** You can do four things with the `/bootsequence` switch:

- List the identifiers for each loader in the order in which you want the boot sequence to appear.
- Add a loader to the top of the list, or if it’s already on the list, move it to the top.
- Add a loader to the bottom of the list, or if it’s already on the list, move it to the bottom.
- Remove a loader from the list completely.
Chapter 1: Getting Started with Windows Server 2008

The following example shows how you would define the boot sequence explicitly with the NT Loader booting first, followed by the OS Loader with the identifier \{0f732d04-e6b2-11da-b631-b722247cd703\} (which in this case is an instance of Windows Server 2008):

```
bcdedit /bootsequence {ntldr} {0f732d04-e6b2-11da-b631-b722247cd703}
```

The example shown here demonstrates how to add or move the OS loader with the identifier \{0f732d04-e6b2-11da-b631-b722247cd703\} to the top of the boot sequence:

```
bcdedit /bootsequence {0f732d04-e6b2-11da-b631-b722247cd703} /addfirst
```

The following example shows how to add or move the OS loader with the identifier \{0f732d04-e6b2-11da-b631-b722247cd703\} to the bottom of the boot sequence:

```
bcdedit /bootsequence {0f732d04-e6b2-11da-b631-b722247cd703} /addlast
```

Finally, if you want to remove an OS loader from the boot sequence completely—for example, if you want to remove NT Loader from the sequence if you no longer use the older Windows version—you could run this command:

```
bcdedit /bootsequence {ntldr} /remove
```

**Setting the Default Boot Entry** To specify which of the boot menu items will be the default boot selection, you use the `/default` switch. For example, to set NT Loader as the default boot loader selection, you would run this:

```
bcdedit /default {ntldr}
```

Simply replace `{ntldr}` with the identifier for whatever OS Loader you want to use as the default.

**Setting the Menu Display Order** When more than one boot loader is available, a menu is automatically displayed allowing you to select one. To set the order in which those entries are displayed, you use the `/displayorder` switch. As you can with the `/bootsequence` switch, you can explicitly define the menu order, add or move an item to the top, add or move an item to the bottom, or remove an item from the menu completely. In fact, the syntax for the `/displayorder` switch is the same as that for `/bootsequence`—except, of course, you would replace `/bootsequence` with `/displayorder`.

For example, to set up the menu order so that the OS loader entry with the identifier \{0f732d04-e6b2-11da-b631-b722247cd703\} is followed by the NT Loader, you would run this:

```
bcdedit /displayorder {0f732d04-e6b2-11da-b631-b722247cd703} {ntldr}
```

Similarly, to add or move the NT Loader to the top of the menu, you would run this:

```
bcdedit /displayorder {ntldr} /addfirst
```
As you can see, the syntax follows the /bootsequence commands exactly, so the /addlast and /remove switches would work the same way.

**Setting the Boot Manager Timeout**  By default, the timeout for the boot manager is 30 seconds. This is probably more time than you will ever need before a selection is made. In practice, this value is typically set from 3 to 5 seconds. You can even set this timeout to 0 so that the menu won’t be displayed. To set the timeout period to 5 seconds, run the following command:

```
Bcdedit /timeout 5
```

Simply replace 5 with whatever timeout period you want in terms of seconds, and it will set it accordingly.

**Setting the Tools Display Order**  If you go through the help menu for BCDEdit as well as the boot manager configuration, you will see an entry for toolsdisplayorder. If you recall the discussion about BCD objects, you will remember that not all objects have to be boot loaders. In fact, the object can be any application designed to run during the boot process. Out of the box, Windows Server 2008 comes with the Windows Memory Diagnostic tool, which can be selected from the boot menu. For a typical Windows installation, you would have only one item in the tools display menu, and that is for the memory diagnostic tool designated by the alias {memdiag}. If Microsoft or a third-party company builds additional tools that can be added to this menu, you can then use BCDEdit to set the order by which these tools are presented in that menu.

For example, if a BCD object functioned as a tool with the identifier {073332d04-e6b2-11da-b631-cdd1327cd703} and you wanted that tool to appear before {memdiag}, you would run the following command:

```
Bcdedit /toolsdisplayorder {073332d04-e6b2-11da-b631-cdd1327cd703} {memdiag}
```

I hope you’re starting to see a pattern here. I bet you’ve already guessed what’s coming next. Yes, the same additional switches that were available in the /bootsequence switch are all available here as well, specifically /addfirst, /addlast, and /remove. The syntax is the same, just replace /bootsequence in those commands with /toolsdisplayorder.

**Backing Up and Restoring the BCD**  The next critical task an administrator will need to ensure is the ability to back up and restore the BCD. In pre–Windows Server 2008 days, you could simply back up the boot.ini file since it was a simple text file. The BCD, on the other hand, is a binary file, and the active BCD file is locked and marked as in-use, so it can’t be copied outright. The correct way to back up and restore a BCD is through the /export and /import switches of BCDEdit. This is all very painless, since the /export switch requires only the destination file name to export the data to, while the /import switch requires only the source file name to import the data from.
Here is an example of backing up the BCD. It will actually create two files after it runs—one is the backup data file and the other is the backup log file:

```
Bcdedit /export "C:\backup\BCD-backup"
```

The following is an example for importing the data that was just backed up. Please be aware that this deletes all the entries in the BCD system store and replaces them with whatever data is in the import file.

```
Bcdedit /import "C:\backup\BCD-backup"
```

**CAUTION**  If you don’t import the right data, your system may become nonbootable after your next reboot when it reads this data, so double-check to make sure you are importing the correct file before you issue the command.

### Manipulating BCD Entries

So far, you’ve read about the most common BCD commands. Sometimes you will need to manipulate BCD entries themselves. This includes the need to create and delete an entry, copy entries within the store, and set entry options. Let’s say you wanted to create a Windows Loader entry manually in the current BCD. This would be necessary if you wanted to have a separate set of boot options for the same install—you could have a normal boot option and one with debugging enabled. Let’s see this in practice since it demonstrates many of the commands for manipulating BCD entries.

Let’s assume your current Windows Server 2008 installation is loaded by the Windows Boot Loader, with the identifier `{0f732d04-e6b2-11da-b631-b722247cd703}`. The first step would be to make a copy of this entry:

```
Bcdedit /copy {0f732d04-e6b2-11da-b631-b722247cd703} /d "Windows Server 2008 (with debug)"
```

This creates a new entry in the BCD system store with the description “Windows Server 2008 (with debug).” By default, this entry is added to the bottom of every list including boot sequence and boot menu order. The output of the previous command is the identifier `{8496b610-6ec8-11db-9581-0003ffaf0a2b}`. If you accidentally closed the window or did not write down the resulting identifier, all you need to do is run `BCDEdit.exe /v` and it will output all the entries on your system. Look for the one with the description you specified, and the ID will be right there.

You can now manipulate this new entry with the debug information you want using the following command:

```
Bcdedit /set {8496b610-6ec8-11db-9581-0003ffaf0a2b} debugtype USB
Bcdedit /set {8496b610-6ec8-11db-9581-0003ffaf0a2b} targetname DBG1
```

The combined effect of the two previous commands is to modify the newly created entry from your copy command, identifier `{8496b610-6ec8-11db-9581-0003ffaf0a2b}`,
with debugtype set to USB and the USB targetname set to DBG1. This is an example, of course, and you would adjust the entry options based on whatever values you really needed.

If you want to delete the entry options you just created, you would run this command:

```plaintext
Bcdedit /deletevalue {8496b610-6ec8-11db-9581-0003ffaf0a2b} debugtype
Bcdedit /deletevalue {8496b610-6ec8-11db-9581-0003ffaf0a2b} targetname
```

It’s important that you specify the identifier in this command; otherwise, it will delete the BCD option in whatever BCD entry the alias `{current}` points to. If you wanted to delete the copy of the BCD entry you created, you would simply run this:

```plaintext
Bcdedit /delete {8496b610-6ec8-11db-9581-0003ffaf0a2b}
```

Three important switches are associated with the `bcdedit /delete` command. First, if you’re trying to delete an entry with a well-known identifier—for example, `{current}`—you also need to specify the `/f` switch to force the deletion. The other two switches that accompany the `/delete` switch are `/cleanup` and `/nocleanup`. If you don’t specify either, the default is `/cleanup`, which not only deletes the entry from the BCD, but also deletes any references to it, such as entries in the boot sequence and boot menu order. If you insist that you want these entries to stay (not generally recommended), you can specify the `/nocleanup` switch that deletes only the entry for the identifier you specified and nothing else.

A few more switches for BCDEdit haven’t been covered here, but they aren’t frequently used and are all listed in the BCDEdit help message. Microsoft did an exceptional job with the help message for this command by providing detailed descriptions of each command along with some easy to understand examples.

---

**TIP** If you want to learn more about BCDEdit, read through the entire help message for this command.

**Manipulating the BCD Using WMI** As a heavy proponent of automation and scripting, I was glad to see that Microsoft had built-in support for WMI to help manage the BCD. The BCD WMI provider is written as a COM object and exposes a number of scriptable classes that can be used to manipulate the BCD using any programming or scripting language that can access COM (which is almost anything mainstream—C++, VBScript, Visual Basic, JScript, and so on). This is not a scripting book, and more than just a simple discussion is necessary to elaborate fully on using WMI, but if you already know how to script with WMI, you can visit the BCD documentation on MSDN (http://msdn2.microsoft.com/en-US/library/aa362692.aspx) to view all the available classes and methods for working with the BCD. If you’re unfamiliar with VBScript or using WMI, visit Microsoft’s Scripting Center (www.microsoft.com/technet/scriptcenter/default.mspx) or pick up a good book on the subject. If you’re serious about administering Windows Server 2008, this is a skill you will definitely want to have under your belt.
CHAPTER SUMMARY

This chapter went through a straightforward installation of Windows Server 2008 and covered in great detail the new Windows Boot Manager called the Boot Configuration Data or BCD. Now that you understand how to install and configure Windows Server 2008, you will need to understand how to take advantage of the new features in this operating system, including the new management tools available, as well as how to incorporate these into your existing environment. We will tackle all this in upcoming chapters.

If you’ve worked with previous versions of Windows Server, you will have undoubtedly noticed a more streamlined installation process. However, don’t be fooled by the marketing hype, because you will still need to perform some significant configuration tasks after installation has completed. The two initial configuration tasks that you should never defer until later are setting the Administrator password and installing all the latest patches. During the installation process covered in this chapter, you read about the option to install Windows Server 2008 as a Server Core installation. This is a very different type of installation that should be used whenever appropriate, since it minimizes the potential attack surface. In the next chapter, you will read in great detail how to install and configure a Server Core installation.
CHAPTER 2

Server Core
With previous versions of Microsoft Windows Server, critical Windows system updates for services were often required to be installed on the server even if they weren’t being used. For example, in 2005, Microsoft released a system update to address the Universal Plug and Play (UPnP) denial-of-service vulnerability across Windows operating systems. Although an available workaround meant that you didn’t necessarily have to install the patch, if your Windows server was performing the function of only a single role—that is, as a Domain Name System (DNS) Server, for example—it shouldn’t be using UPnP in the first place, so the patch would’ve been unnecessary if the service wasn’t installed in the first place, thereby exposing the operating system to unneeded vulnerabilities.

In Windows Server 2008, Microsoft addresses this issue by introducing an installation option called Server Core. This installation option installs the most basic Windows Server component for the role the Windows Server will perform. There is one caveat, however, in that currently, Microsoft supports the Server Core installation for only a handful of predefined roles, such as domain controller (DC), DNS Server, Dynamic Host Configuration Protocol (DHCP) Server, and file server. This basic installation of Windows Server 2008 doesn’t even install Windows Explorer, so you have no desktop with which to interact. Instead, the system must be managed completely through the command line or via Terminal Services. Microsoft realized that if a server is performing a very distinct infrastructure role, excess services need not be installed on it—not even a full graphical user interface (GUI). This minimizes the server’s attack surface and will hopefully help reduce downtime by reducing the need to install system updates on the server.

**ROLES SUPPORTED BY SERVER CORE**

Microsoft intended the Server Core installation method to be used for infrastructure-related services. Because of all this, Microsoft supports only the following seven roles in the Server Core installations:

- ▾ Active Directory Domain Services (AD DS)
- ■ Active Directory Lightweight Directory Services (AD LDS)
- ■ File Server
- ■ DHCP Server
- ■ DNS Server
- ■ Print Server
- ▲ Streaming Media Services

These roles are not mutually exclusive. A Server Core instance can have one or more roles installed and configured without encountering any serious issues.
THE UPS AND DOWNS OF SERVER CORE

Using a Server Core installation offers many useful benefits. It reduces the potential vulnerability footprint by not installing any unneeded services and binaries. As a result, it also reduces the amount of servicing that needs to be done to the operating system and therefore reduces the amount of management overhead required to maintain these servers. The downside is that a Server Core installation doesn’t provide much of a user interface to work with, other than the command prompt. The only way to manage a Server Core installation is through command-line tools and scripts, Microsoft Management Console (MMC) snap-ins, or other tools that support remote administration and Terminal Services (although your Terminal Services session will have only a command prompt anyway). This is quite cumbersome, especially for those who have been spoiled over the years by point-and-click administration techniques. Luckily, if you do it right, you will need to run only a minimal number of commands to set up remote management through some kind of management console.

INSTALLING SERVER CORE

As you might expect, installation of Server Core is not much different from installation of the regular Windows Server 2008. In fact, both installations share the same steps, except that at the end of the Server Core installation process, rather than facing an Initial Configuration Tasks screen, you are presented with a command prompt. You will make all your configuration changes using this command prompt. If you close the command prompt, you will have to press ctrl-alt-del, click Start Task Manager, click File, then click Run and enter cmd.exe to open a new command prompt.

Requirements

Windows Server 2008 Server Core shares the same minimum requirements with the regular Windows Server 2008 installation—with a few caveats. In addition to having the Windows Server 2008 installation media and a valid product key, you will also need to perform a clean installation. You cannot upgrade from previous versions of Windows to a Server Core installation, you cannot upgrade from a regular Windows Server 2008 installation to Server Core, and you cannot move from Server Core to a regular Windows Server 2008 installation. Server Core must be installed from scratch. You should also have Internet access so that the server can be activated after the installation completes. Also, since fewer binaries are installed as part of Server Core, the hard disk space requirements are much lower for Server Core than they are for the regular Windows Server 2008 installation. You will need only 1GB of disk space for the actual Server Core installation and 2GB of disk space for regular server operations.
Hands-On Exercise: Interactive Installation of Server Core

1. Start the computer and boot up using the Windows Server 2008 installation media. Select the installation language, time and currency format, and keyboard layout. Then click Next.

2. Click Install Now to begin the installation process.

3. Enter your product key, and then click Next. If you don’t want to activate Windows as soon as your computer goes online (for example, if you are simply testing the installation or evaluating Windows Server 2008), you can uncheck the Automatically Activate Windows When I’m Online checkbox.

4. Now select whether to install Windows Server 2008 Enterprise (Full Installation) or Windows Server 2008 Enterprise (Server Core Installation). Select Windows Server 2008 Enterprise (Server Core Installation), as shown in Figure 2-1, and then click Next.
5. If you accept the license agreement, check the I Accept the License Terms (required to use Windows) checkbox, and then click Next.

6. Select the type of installation you would like to perform. In this case, you’ll perform a clean install and you can select Custom (Advanced).

7. If your hard drive is automatically detected, you can create and format partitions as necessary for the installation. If your drive isn’t detected, most likely the device driver for your controller isn’t built into Windows, in which case you can click Load Driver to load it. Click Next after you have created the partition to which you are going to install.

8. Now that Windows Server 2008 has all the basic information it needs to proceed with the installation, it begins to go through the installation process and displays the status of the install.

9. Once the installation completes, you will be prompted to press Ctrl-Alt-Del to log in.

10. Click the Other User button as shown in Figure 2-2 to initiate login. Enter Administrator as the username, leave the password blank, and then click the arrow button to log in (or simply press ENTER).
11. When logging in for the first time, you will be prompted to change your password. Leave the current password field blank and enter your new password in the New Password and Confirm Password fields. Click OK when your password change has been confirmed.

12. Once you’re logged in, you will see a command prompt and nothing else (Figure 2-3). At this point, you can manage this server only by using these command prompts. Remote administration is disabled by default. Your next step will be to perform initial configuration tasks using the command prompt, as discussed in detail in the next section.

Post-Installation Tasks
Installing Server Core for Windows Server 2008 is the easy part. Without a real user interface to assist you in configuring the server, you will need to get used to working with the command prompt if you don’t already work with it. Your first order of business after
installing Server Core is to run through the initial configuration tasks, except this time without the help of a handy screen to walk you through it:

1. Set the Administrator password.
2. Configure your network interfaces.
3. Activate the server.
4. Rename the server and join it to a domain (if applicable).
5. Configure Automatic Updates.
6. Enable remote administration (unless you like sitting in front of the server every time you need to work on it).
7. Configure the Windows Firewall.

**Setting the Administrator Password**

You were already prompted to change the password the first time you logged on, however, you can change the administrator password locally in two ways. The easiest way is to press `ctrl-alt-del` and then select Change a Password. You can accomplish the same thing straight from the command prompt:

```
Net user Administrator P@ssword
```

Simply replace `P@ssword` with whatever password you want to use. The main difference between these two methods is that the graphical method requires you to enter the old password and then the new password twice before changing the password; in the command-line method, the password is changed immediately. Because no confirmation prompt appears after you change a password from the command line, it is crucial that you proceed very carefully and record your new password to reduce the possibility of a typographical error.

**Configuring Your Network Interfaces**

By default, your new Server Core installation uses DHCP to acquire an IP address. If you will be using a static IP address for the server, you will need to assign this using the `Netsh` command. This requires more than one command sequence since you will need to take a number of steps. Your first step is to list all your network adapters. This is important, because most servers come with more than one network interface, plus the default loopback interface. When you configure the IP address, you will need to specify which interface you are going to modify. To list all your network adapters, enter the following:

```
Netsh interface ipv4 show interfaces
```

Although IPv6 is not currently widely implemented, except probably in test labs, Windows Server 2008 natively supports it. An equivalent command for IPv6 is as simple as replacing `ipv4` in this command with `ipv6`. The output of the command on my Server Core installation is shown in Figure 2-4.
The first column of the command’s output shows a parameter called \textit{Idx}. This is the unique number assigned by the system to identify each network interface. Note the \textit{Idx} number of the interface you are interested in modifying. On my test server, I have only one network interface, excluding the loopback interface, so that’s what I will be modifying in this example.

I will set my network interface to have the static IP address 192.168.100.75 with a subnet mask of 255.255.255.0 and a default gateway of 192.168.100.1. If I look at the \textit{Idx} number for my Local Area Connection in Figure 2-4, I can see that the value is 2 for my network interface. Putting all this information together, I can now run the following command to set these values:

\texttt{Netsh interface ipv4 set address name=2 source=static address=192.168.100.75 mask=255.255.255.0 gateway=192.168.100.1}

Since DNS is so critical to Windows Server 2008, especially in an Active Directory domain, I would also need to configure the DNS Servers for this server. In this case,
I want to set this interface to use the DNS Server with the IP address 192.168.100.40. To set this value, I run the following command:

```
Netsh interface ipv4 add dnsserver name=2 address=192.168.100.40
```

If more than one interface needs to be configured, I would simply repeat this process for every interface. If you are trying to set up network interface card (NIC) teaming or failover, you should consult your vendor’s documentation to determine how to accomplish this task in Server Core, since most vendors supply graphical interfaces to configure these advanced options, and those will not run on a Server Core installation.

**Activating Your Server**

If you’re setting up a server that will be running Windows Server 2008 for more than 14 days, you will want to activate your server or it will no longer function once the trial period has elapsed. No graphical method can be used to activate your server in Windows Server 2008; instead, you will have to rely on the nifty Windows Software License Management Tool, otherwise known as slmgr.vbs, that sits in the %WINDIR%\system32 directory. To activate your server, simply run this command:

```
Slmgr.vbs -ato
```

It can’t get any easier than that. In fact, the slmgr.vbs script is so powerful you can actually use it to initiate the activation of a new Windows Server 2008 installation remotely from an existing Windows Server 2008 server. Let’s say, for example, that you wanted to activate a new Windows Server 2008 installation called Utopia that had a local

---

**Netsh Up Close and Personal**

*Netsh* is the ultimate command-line shell for managing all aspects of the network components of Windows Server 2008. This command was available in previous Windows versions but is now an even more critical tool for Windows Server 2008. It can be used to query and manage everything from a network interface, Windows Firewall, and DHCP Server parameters including defining scopes and exclusions, to defining routing and remote access policies. The ability to do all these things from the command line makes this tool highly useful for Windows administrators when they want to script various network service-related tasks. However, many administrators neglect to learn *netsh* well, since everything they can do in *netsh* can be done more easily with any of the Windows GUIs. Server Core makes it necessary for Windows administrators to learn how to use this tool rather than make it an afterthought. Although many core network services that a Server Core instance can provide can be managed remotely using an MMC snap-in, many key tasks cannot be accomplished without *netsh*, especially with regard to configuring network interfaces, such as setting a static IP address or listing DNS Servers to use.
administrator password of password123. This could be easily accomplished remotely by running the following command from an existing Windows Server 2008 installation:

```
Slmgr.vbs Utopia Administrator password123 -ato
```

**Rename the Server and Add It to Your Domain**

Since the Windows Server 2008 installation process doesn’t ask for the computer name before proceeding with the install, the server is given a computer-generated name. This unintuitive random name is practically useless in most environments, so you’ll need to rename the server to something more meaningful before joining it to the domain. Microsoft’s documentation tells you to use the `netdom` command to rename a computer. The problem with this command, however, is that you can’t rename a computer until it has joined the domain. To rename the computer before it joins the domain without having to run a third-party tool, you need to use Windows Management Interface (WMI). Rather than writing a script and then executing it, the easier way is simply to run WMI Command-line (WMIC). This command-line tool is specifically designed to run WMI commands and is ideal for straightforward commands like this. For example, to rename your server to WINSRV1, you would run this command:

```
wmic computersystem where name="%computername%" rename name="WINSRV1"
```

This command should result in a `ReturnValue=0` to indicate a successful rename, as shown in Figure 2-5. Before going on, make sure you reboot the server for the new computer name to take effect.

After you rename the computer, you can join it to the domain using the `netdom join` command. You will need to know three pieces of information to complete this command: the name of the domain, a username of an account that has rights to join computers to the domain, and of course the password for that user account. For example, if you wanted to add this server to the TESTLAB domain using an account called SysAdmin with the password P@ssword, you would run the following:

```
Netdom join %computername% /domain:TESTLAB /userd:SysAdmin /password:P@ssword
```

If you don’t want to type the password explicitly like this because people around you can view the console, you can replace `/passwordd:P@ssword` with `/passwordd:"` in which case it will prompt you to type in the password instead. You will need to restart the computer after it has been joined to the domain.

**NOTE** Don’t be concerned if this command takes a while to complete. Depending on your network environment, it could take a minute or two before the command can complete successfully.

**Configure Automatic Updates**

You would think that Microsoft would have at least provided an easy way to initiate and configure Automatic Updates, but without Windows Explorer or even Internet Explorer,
getting updates installed can be quite tricky. You’ll have to rely on a Windows script file called scregedit.wsf, which is located in the %WINDIR%\System32 directory. Unfortunately, with Server Core, it’s all or nothing when it comes to Automatic Updates. You either enable or disable it completely. Since there’s no GUI, you have no way of controlling which updates to install. Of course, the workaround to all this is to configure Automatic Updates using group policy in conjunction with a patch-management solution such as Windows Server Update Services to control exactly which patches your server will receive.

To enable Automatic Updates manually, you can run this command:

```
Cscript Scregedit.wsf /AU 4
```

To turn off Automatic Updates (the default), you would run this command:

```
Cscript Scregedit.wsf /AU 1
```
NOTE  A graphical warning message is displayed whenever you run Scregedit.wsf commands. To avoid this, make sure that when you open a command prompt to run these commands, you change the current directory to %WINDIR%\System32 and run the command using CScript. For example, you could run cscript.exe scregedit.wsf /AU 4.

Enable Remote Administration

Technically, you can already remotely manage your Server Core installation using the Computer Management MMC snap-in; however, access via Terminal Services in Remote Administration mode is disabled by default and you will need to turn it on if you want that capability. To do so, go back to the scregedit.wsf script and run the following:

Scregedit.wsf /AR 0

Yes, that is a zero. This is actually designed in reverse logic. The 0 means you want to enable Terminal Services in Remote Administration mode and 1 means you want to disable it. If you want to manage your Windows Server 2008 instance from a previous Windows version, you will need to allow these types of “legacy” connections explicitly, since by default, a higher level of security is built around the Terminal Services in Windows Server 2008, called Credential Security Service Provider (CredSSP). To allow terminal service connections from a previous Windows version, run this command:

Scregedit.wsf /CS 0

If you set CS to 1, this forces Terminal Services to use CredSSP, which is currently supported only by Windows Server 2008 and Windows Vista.

TIP  Since the Windows Firewall is enabled on all interfaces on all profiles by default, simply enabling Terminal Services in Remote Administration mode won’t allow you to control the server remotely using Remote Desktop Protocol (RDP). The right way is to explicitly open the Terminal Services port on the server. This can be achieved by adding a firewall rule to allow inbound TCP connections to port 3389 through netsh:

Netsh advfirewall firewall add rule name="TS Admin" protocol=TCP dir=in localport=3389 action=allow

Configure the Windows Firewall

The Windows Firewall is a host-based, bidirectional network traffic filter. Unlike the initial incarnation of the Windows Firewall that debuted in Windows XP SP2 and filtered only inbound traffic, the new Windows Firewall can control both inbound and outbound traffic. The current Windows Firewall is also network-aware, in that you can define policies depending on whether the server is on the network where it can authenticate to the domain, on a public network that is directly attached to the Internet, or on a private network explicitly defined. For example, you can configure policies to allow file and print sharing when in a domain network and then block it if on a public network.
Chapter 2: Server Core

Configuring the firewall involves either working with the **Netsh** command at the command prompt or using the Windows Firewall with Advanced Security MMC snap-in from a remote Windows Server 2008 server. Unless you’re absolutely hardcore and love playing with the command line, I strongly recommend using the Windows Firewall with Advanced Security MMC snap-in. However, before you can remotely manage the Server Core installation’s firewall using the MMC snap-in, you will have to enable remote management. To enable remote management of the firewall, enter the following:

```
Netsh advfirewall set current settings remotemanagement enable
```

Once remote management is enabled, you can go to another Windows Server 2008 installation and add the Windows Firewall with Advanced Security MMC snap-in and point it to the server you want to manage. Unfortunately, if only one Windows Server 2008 instance is on your network, you will need to configure the firewall using **Netsh**.

To view all the profile-specific properties in all profiles, you can run this command:

```
Netsh advfirewall show allprofiles
```

In the output, you’ll see the general properties of your domain, public and private profiles such as its state (whether it’s enabled or disabled), the general firewall policy such as whether it allows outbound connections but prevents inbound connections, and the name of the log file. If you want to enable a specific profile—for example, the domain profile—you can run this command:

```
Netsh advfirewall set domainprofile state on
```

Let’s say you want a rule to allow inbound TCP connections to port 80. This can be accomplished by running the following command:

```
Netsh advfirewall inbound add name="Port80 Allow" protocol=TCP localport=80 action=allow
```

The Windows Firewall allows you to create a blanket rule to allow or disallow any traffic to and from an application based on a particular executable. For example, if you had an application called myapp.exe in the C:\myapp directory that performed some kind of networking function by listening to several ports on the server, you could allow any connection to this application by running this:

```
Netsh advfirewall inbound add name="Allow Myapp" program="C:\myapp\myapp.exe" action=allow
```

You can view all your currently defined inbound rules by running this command:

```
Netsh advfirewall inbound show name=all verbose
```

The **verbose** parameter is optional, but if you omit it, you won’t see the path to the executable for any application-based rules you’ve defined.
This barely scratches the surface of all the `netsh` commands you can use to configure the Windows Firewall. To find out more about `netsh` firewall commands, view the `netsh advfirewall` help file by running this command:

```
Netsh advfirewall help
```

As you can tell, this method of manipulating the Windows Firewall can be quite tedious. It’s most useful when you are creating a script to define the firewall rules. In most cases, though, it’s best to use the Windows Firewall with Advanced Security MMC snap-in, as it offers a more intuitive and easier method for defining rules and configuring profiles.

## Installing and Configuring Server Roles

Up to this point, you have accomplished a base installation of Server Core. Just like the regular Windows Server 2008 installation, there are no roles installed by default in Server Core. If you want your Server Core installation to perform any of the six supported roles, you will need to install each of them individually from the command line. Since only six roles are supported by Server Core, you need to know only a handful of commands.

### Installing and Configuring the DNS Server Role

DNS is a key infrastructure component because it’s so critical to Active Directory. This role is an ideal candidate for Server Core, since once you set it up, you probably won’t touch it much other than to perform regular maintenance. To install the DNS Server role, you run this command:

```
Start /w ocsetup DNS-Server-Core-Role
```

It will take a few minutes to install and it won’t display a progress dialog box, so be patient. Remember that this installs only the DNS Server role, and nothing is really configured yet. You can configure the DNS Server using the DNS MMC snap-in from a different computer or by running `dnscmd` at the command prompt. To view the general parameters of your newly installed DNS Server, you can run this command:

```
Dnscmd /info
```

The most logical first step after installing a DNS Server would be to configure the DNS zones. For example, to add a zone called `testlab.local` as a primary zone, you can run this command:

```
Dnscmd /zoneadd "testlab.local" /Primary /file "testlab.local.dns"
```

Now if you want to add an A record for a host called `testpc` with the IP address `192.168.100.71` to the `testlab.local` zone, you’d enter this:

```
Dnscmd /recordadd testlab.local testpc A 192.168.100.71
```
The `/recordadd` switch can be used to add any record type you want to the DNS Server. You would simply replace the `A` before the IP address with whatever record type you wanted—for example, `CNAME` or `MX` followed by the parameters required by that record type. Run this command to see a list of available record types and their parameters:

```
Dnscmd /recordadd /?
```

If you want to view all the records of a particular zone, use the `/zoneprint` switch. For example, to list all the entries of your `testlab.local` zone, you would run this:

```
Dnscmd /zoneprint testlab.local
```

If you want to delete a record, you would run `dnscmd` with the `/recorddelete` switch. To delete the `A` record entry for the `testpc` record created earlier, you’d run this command:

```
Dnscmd /recordadd testlab.local testpc A 192.168.100.71 /f
```

The `/f` switch at the end indicates that you want to force the deletion of this record; otherwise, `dnscmd` will politely ask for confirmation before deleting the record.

There’s more to DNS than what you’ve learned so far, especially the new features of DNS in Windows Server 2008, which are covered in Chapter 10. `Dnscmd` is a powerful and useful command for configuring DNS on Windows Server 2008. It’s the only method to make changes to your DNS Server locally on the server, but it can also be executed remotely from a different server. Again, I would recommend using the DNS MMC snap-in whenever possible rather than `dnscmd`, since the snap-in is far more intuitive.

If you later decide that this Server Core instance will no longer provide DNS services, you can uninstall it by running the following:

```
Start /w ocsetup DNS-Server-Core-Role /uninstall
```

### Installing and Configuring the DHCP Server Role

Whether you are configuring a small environment or an enterprise-size network, you will most likely want to use DHCP to manage the IP addresses in your environment. Before you can do that with Windows Server Core, you will need to install this role using the following command:

```
Start /w ocsetup DHCPServerCore
```

Once installed, you will have the option to configure your DHCP scopes using either `netsh` or the DHCP MMC snap-in from a remote server. Also, if this DHCP Server is acting within an Active Directory domain, it must also be authorized in Active Directory before it can issue IP addresses. You can authorize a DHCP Server in the domain using the DHCP MMC snap-in, but it can also be done using `netsh`. For example, if your Server
Core instance is called WINDHCP1 and has the IP address 172.16.0.5, and you want to authorize this on your domain, log onto WINDHCP1 with domain credentials that have rights to authorize DHCP servers, and then run the following command:

```
Netsh dhcp add server WINDHCP1 172.16.0.5
```

Likewise, if you wanted to unauthorized the server, you can run this:

```
Netsh dhcp delete server WINDHCP1 172.16.0.5
```

If you later decide that this Server Core instance will no longer provide DHCP services, it can be uninstalled like so:

```
Start /w ocsetup DHCPServerCore /uninstall
```

### Installing and Configuring the File Server Role

By default, your basic File Server role is installed on Windows Server 2008, including Server Core. If you want to use some more advanced File Server roles, such as the following, they will need to be installed:

- File Replication
- Distributed File System (DFS)
- Distributed File System Replication
- Network File System (NFS)

It should come as no surprise that to install these additional roles you will use the `ocsetup` command as you did for the DNS and DHCP installations. Table 2-1 shows the command to install each File Server role.

Currently no command-line tools are used to manage these additional File Server roles, so you will need to resort to managing them remotely via the appropriate MMC snap-ins. To uninstall any of them, you can run the same command used to install them and add a `/uninstall` switch at the end.

<table>
<thead>
<tr>
<th>Role</th>
<th>Installation Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Replication</td>
<td><code>start /w ocsetup FRS-Infrastructure</code></td>
</tr>
<tr>
<td>Distributed File System</td>
<td><code>start /w ocsetup DFSN-Server</code></td>
</tr>
<tr>
<td>Distributed File System</td>
<td></td>
</tr>
<tr>
<td>Replication</td>
<td></td>
</tr>
<tr>
<td>Network File System</td>
<td><code>start /w ocsetup ServerForNFS-Base</code></td>
</tr>
<tr>
<td></td>
<td><code>start /w ocsetup ClientForNFS-Base</code></td>
</tr>
</tbody>
</table>

**Table 2-1.** Commands to Install File Server Roles
Installing and Configuring the Print Server Role

One of the most prevalent uses for Windows servers is to act as print servers. This is generally regarded as a core infrastructure role that makes perfect sense to belong in Server Core. In most environments, a print server acts as a print server and nothing else, and fits nicely into the Server Core model of having minimal additional services for key infrastructure roles. To install the Print Server role, simply run this command:

```
Start /w ocsetup Printing-ServerCore-Role
```

If you want to install the Line Printer Daemon (LPD) service, you can run this:

```
Start /w ocsetup Printing-LPDPrintService
```

Installing and Configuring the Streaming Media Services Role

Streaming media servers are generally deployed when you want to provide streaming audio or video content to your users. This doesn’t necessarily have to be aimed at the general public. In fact, many organizations use streaming media services internally to provide host training videos and other internally developed content that needs to be shared with the general user community. Like print services, streaming media servers are generally single purpose and make ideal Server Core candidates.

To install the Streaming Media Services role, perform these steps:

1. Download the Streaming Media Services installer file from KB934518 on Microsoft’s support site (http://support.microsoft.com/kb/934518) and copy it to your Server Core installation. Remember that you need to do this from a different server since Server Core doesn’t have a browser.
2. Run the downloaded MSI file.
3. Install the service role by running this command:

   ```
   Start /w ocsetup MediaServer
   ```

   Just as with the other services, you will need to manage your newly installed role remotely from another server or workstation using the Streaming Media Services MMC snap-in.

Installing and Configuring the Active Directory Domain Services Role

Of all the different roles included in a Server Core installation, this is by far the most complex. There’s no equivalent `ocsetup` command to use to install Active Directory; instead, you have to rely on `dcpromo.exe`, just as you did in Windows 2000/2003. Because of the way Server Core is set up, the `dcpromo.exe` GUI can’t be displayed. This forces you to install Active Directory via an unattended setup.

To install Active Directory, run the following command:

```
dcpromo /unattend:c:\unattend.txt
```
This assumes that c:\unattend.txt is your answer file for dcpromo. If it’s in a different location, enter the path to that file.

This may seem straightforward, but you’re probably wondering how to format an unattend.txt file for dcpromo. The dcpromo that’s built into Windows Server 2008 is for the most part the same as that built into Windows Server 2003 with some newly supported features. They’re similar enough that you can use the unattend.txt files for Windows Server 2003 on Windows Server 2008. See Figure 2-6 for a sample unattend.txt file. The unattend.txt file supports many more options than what are shown in the sample file. You need to specify only those options that you want to use. The sample in Figure 2-6 is the unattend.txt file I used to join my Server Core installation to my existing Windows Server 2008 domain called testlab.local.

Everything you can do in the graphical version of dcpromo can be done through an unattended installation. Table 2-2 lists all the possible parameters you can use in your unattend.txt file to install and configure a domain controller. If a parameter is not applicable to your installation, you don’t need to include it in your answer file. For example, you don’t need to enter parameters relating to domain controller demotion when you are promoting a standalone server to a domain controller role.

![Sample unattend.txt file for dcpromo](image)

**Figure 2-6.** Sample unattend.txt file for dcpromo
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdministratorPassword</td>
<td>&lt;password&gt;</td>
<td>When demoting a DC, this sets the default administrator password. If not specified, it will default to blank.</td>
</tr>
<tr>
<td>AllowDomainControllerRemove</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ApplicationPartitionsToReplicate</td>
<td>&lt;partitions&gt;</td>
<td>Specifies application partition to be replicated. If * is specified, all partitions will be replicated.</td>
</tr>
<tr>
<td>AutoConfigDNS</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ChildName</td>
<td>&lt;child_domain_name&gt;</td>
<td>The portion of the domain name that refers to the child domain. Applies only when installing a child domain.</td>
</tr>
<tr>
<td>ConfirmGC</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>CriticalReplicationOnly</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Database Path</td>
<td>&lt;path_to_database_file&gt;</td>
<td>The full path to where the domain database will be stored. The default is %SYSTEMROOT%\NTDS.</td>
</tr>
<tr>
<td>DemoteFSMO</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DisableCancelForDnsInstall</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 2-2. Parameters Available for a dcpromo Unattended Install
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNSDelegation</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DNSDelegationUserName</td>
<td>&lt;username&gt;</td>
<td>The username used for creating DNS delegation.</td>
</tr>
<tr>
<td>DNSDelegationPassword</td>
<td>&lt;password&gt;</td>
<td>Password of the username used for creating the DNS delegation.</td>
</tr>
<tr>
<td>DNSOnNetwork</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DomainNetBiosName</td>
<td>&lt;domain_netbios_name&gt;</td>
<td>Assigns the specified NETBIOS name to the domain. Use this option for new domains.</td>
</tr>
<tr>
<td>DomainLevel</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>ForestLevel</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>IgnoreIsLastDNSServerForZone</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>IgnoreIsLastDcInDomainMismatch</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>IsLastDCInDomain</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>LogPath</td>
<td>&lt;path_to_logfile&gt;</td>
<td>Path to store the domain log files. Defaults to %SYSTEMROOT%\NTDS.</td>
</tr>
<tr>
<td>NewDomain</td>
<td>Tree</td>
<td>Child</td>
</tr>
<tr>
<td>OnDemandAllowed</td>
<td>&lt;security_group&gt;</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 2-2. Parameters Available for a dcpromo Unattended Install (Continued)
### Table 2-2. Parameters Available for a dcpromo Unattended Install (Continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OnDemandDenied</td>
<td>&lt;security_group&gt;</td>
<td>None</td>
</tr>
<tr>
<td>ParentDomainDNSName</td>
<td>&lt;dns_name_of_domain&gt;</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td>&lt;password&gt;</td>
<td></td>
</tr>
<tr>
<td>RebootOnCompletion</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>RebootOnSuccess</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>RemoveApplicationPartitions</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>ReplicaDomainDNSName</td>
<td>&lt;dns_name_of_domain&gt;</td>
<td></td>
</tr>
<tr>
<td>ReplicaOrNewDomain</td>
<td>&lt;Replica&gt;</td>
<td>ReadOnlyReplica</td>
</tr>
<tr>
<td>ReplicationSourceDC</td>
<td>&lt;dns_name_of_DC&gt;</td>
<td></td>
</tr>
<tr>
<td>ReplicationSourcePath</td>
<td>&lt;replication_source_path&gt;</td>
<td></td>
</tr>
<tr>
<td>SafeModeAdminPassword</td>
<td>&lt;password&gt;</td>
<td></td>
</tr>
<tr>
<td>SiteName</td>
<td>&lt;site_name&gt;</td>
<td></td>
</tr>
<tr>
<td>SysKey</td>
<td>&lt;none&gt;</td>
<td>system key</td>
</tr>
<tr>
<td>SysVolPath</td>
<td>&lt;path_to_database_file&gt;</td>
<td></td>
</tr>
<tr>
<td>UserDomain</td>
<td>&lt;domain_name&gt;</td>
<td></td>
</tr>
<tr>
<td>UserName</td>
<td>&lt;username&gt;</td>
<td></td>
</tr>
</tbody>
</table>
You will notice the references to read-only DCs in the parameters listed in Table 2-2. This is a new feature for Windows Server 2008 domain controllers. You’ll read about all the new aspects of Active Directory for Windows Server 2008 in Chapter 4.

### Installing Optional Features

Once you have installed the Server Core and installed and configured all the roles, you can install optional features. The following optional features are available for installation on Server Core:

- ▼ Backup
- ■ Bitlocker Drive Encryption
- ■ Microsoft Failover Clustering (not available in Windows Server Standard Edition)
- ■ Multipath IO
- ■ Network Load Balancing
- ■ Removable Storage Management
- ■ Simple Network Management Protocol (SNMP)
- ■ Subsystem for UNIX-based applications
- ■ Telnet Client
- ▲ Windows Internet Name Service (WINS)

---

**NOTE** Some optional features require appropriate hardware. These features are Bitlocker Drive Encryption, Microsoft Failover Clustering, Multipath IO, Network Load Balancing, and Removable Storage Management.

All the optional features are installed using the familiar `ocsetup` command you used to install the other server roles. Table 2-3 lists the commands needed to install each of these features.

---

**NOTE** The commands to install the optional features are case-sensitive! To uninstall an optional feature, run the same command to install it and add a `/uninstall` switch at the end.

### Server Core Management

As you’ve seen so far, with the limited user interface of Server Core, it’s difficult to manage a Server Core instance locally or even remotely from Terminal Services, since you have to know all the manual commands to get anything done. For the most part, you’ll be doing all your management remotely using an MMC snap-in loaded on your workstation or from another Windows server. The only actual graphical application included as part of Server Core, besides Task Manager, is Notepad, and you will need that to edit
text files. You can install Windows Installer (MSI)–based packages, but the GUI can’t be displayed, so you will need to specify all the parameters the installation needs in order to make it a quiet install, and then specify the /qb switch. For example, if you had a third-party toolkit for Windows Server 2008 packaged in an MSI called mgmtpack.msi in the root of the C: drive, you would run this:

Msiexec /I c:\mgmtpack.msi /qb

You can run two Control Panel applets in Server Core: the time zone and international settings applets. The time zone applet lets you set the date, time, and time zone. To run this applet, enter the following:

Control timedate.cpl

The international settings applet sets the currency format, location, keyboards, and languages. It can be accessed by entering this:

Control intl.cpl

<table>
<thead>
<tr>
<th>Feature</th>
<th>Installation Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup</td>
<td>start /w ocsetup WindowsServerBackup</td>
</tr>
<tr>
<td>Bitlocker Drive Encryption</td>
<td>start /w ocsetup BitLocker</td>
</tr>
<tr>
<td>Microsoft Failover Clustering</td>
<td>start /w ocsetup FailoverCluster-Core</td>
</tr>
<tr>
<td>Multipath IO</td>
<td>start /w ocsetup MultipathIo</td>
</tr>
<tr>
<td>Network Load Balancing</td>
<td>start /w ocsetup NetworkLoadBalancing</td>
</tr>
<tr>
<td>Removable Storage Management</td>
<td>start /w ocsetup Microsoft-Windows-RemovableStorageManagementCore</td>
</tr>
<tr>
<td>Simple Network Management Protocol</td>
<td>start /w ocsetup SNMP-SC</td>
</tr>
<tr>
<td>Subsystem for UNIX-based applications</td>
<td>start /w ocsetup SUACore</td>
</tr>
<tr>
<td>Telnet Client</td>
<td>start /w ocsetup TelnetClient</td>
</tr>
<tr>
<td>Windows Internet Name Service</td>
<td>start /w ocsetup WINS-SC</td>
</tr>
</tbody>
</table>

Table 2-3. Installation Commands for Server Core Optional Features
You can also manage a Server Core installation using Windows Remote Shell. Using Windows Remote Shell is like running a command prompt remotely, just as you would telnet into a UNIX or Linux system. This isn’t enabled by default, so your first step is to enable Windows Remote Shell on the server using the following command:

```
WinRM quickconfig
```

You will be prompted if you want to accept the change. You can then connect using remote shell from another computer by running Winrs. For example, to connect to a Server Core instance called WINSRVCORE and open the command prompt, you would run this command:

```
Winrs -r:WINSRVCORE cmd
```

Winrs also supports additional switches such as specifying the username and password when connecting to the remote server and setting environment variables when the shell starts. This tool is useful if you want to run a command-line-based tool remotely without having to log in first using Terminal Services.

One piece of functionality that is difficult with a Server Core installation is managing hardware. If you attach a Plug and Play device to the server, the driver will be automatically loaded; however, if it is not an easily recognized device and requires loading of third-party drivers, extra steps are required. You need to perform two distinct steps. First, you need to copy the driver files to the server. Then you can load the driver using the INF file that came with the driver using the `drvload` command. For example, if you copied the drivers to `C:\TEMP\NEWDRIVERS` and the INF file for the driver was called `oemsetup.inf`, you would run this:

```
C:
    cd temp\newdrivers
    Drvload oemsetup.inf
```

If you want to query the list of all the drivers installed on the server, you would use the `SC` command (note the required space between the equal sign and the word `driver`):

```
Sc query type= driver
```

The `SC` command was historically part of the Windows Resource Kit, but is now a built-in command and can be used not only to query device drivers but also services. This command can be used to configure the startup type of devices and services from disabled all the way to automatic and even delete devices and services among other things. To find out more about the `SC` command, run the following:

```
SC /?
```
CHAPTER SUMMARY

In this chapter you got your arms around installing and configuring Server Core. Although you have no graphical tools to configure Server Core, the graphical installation option should be considered first, especially when the role of the server matches one of the roles supported by Server Core. The smaller installation footprint increases performance and reliability and decreases the potential for security vulnerabilities. This is probably one of the biggest changes to Windows Server. The reality is that the entire Windows Server 2008 build is based around componentization. Server core is just an extreme form of that, since it truly strips away all the unnecessary clutter that can be installed with a regular Windows Server installation. If you’re a Windows administrator who has tried to get away from the command prompt as much as possible, you should really reconsider and take the time to learn these tools. The command-line tools available natively in Windows Server 2008 provide rich functionality and when used properly can be more effective than the graphical tools, simply because of their ability to be automated through the use of scripts.
This page intentionally left blank
CHAPTER 3

Server Manager
If you’ve ever had to manage a Windows NT domain, you will remember using a tool called Server Manager to manage workstation and server accounts. Windows 2000 Server did away with Server Manager, since its functionality was replaced by Active Directory Users and Computers. Windows Server 2008 introduces a new Server Manager tool, but don’t think Microsoft is going back two steps. This is an entirely new tool that shares nothing with its predecessor other than its name. The new Server Manager was designed to be used as a single source for managing and monitoring most aspects of your server, offering the ability to install and configure components and view system status. You can think of Server Manager as a portal into your server, since it performs the exact same function as a portal. Rather than replace all the tools into which it offers views, Server Manager centralizes the presentation of key information and then provides links into the appropriate tools you’ll need to configure each item. Server Manager replaces the functions of Manage Your Server, Configure Your Server, and Add or Remove Windows Components from Windows Server 2003.

WHAT IS SERVER MANAGER?

It seems as though with every release of Windows Server, Microsoft is finding more and more ways to simplify server administration. Server Manager is a new MMC snap-in that represents a consolidation of all the various wizards and tools Microsoft has provided in previous Windows Server releases for server management. By default, Server Manager starts up automatically after you have completed the Initial Configuration Tasks screen that is displayed upon installing Windows Server 2008. It then runs automatically every time you log on to the server unless you check the Do Not Show Me This Console at Logon checkbox in the Server Summary section of Server Manager. Figure 3-1 shows what Server Manager looks like after a fresh installation of Windows Server 2008 and after a few configuration changes have been made, such as changing the server name and configuring automatic updates.

Server Manager is actually made up of several components—mostly wizards that allow you to add or remove roles and features in your Windows Server 2008 installation. Various role management home pages automatically scan your system for each installed role. A summary high-level view is then displayed so you can have a quick overview of each role, including the status of related services and links to various role-specific tools and resources.

If you close Server Manager or configure it not to start up automatically, it can still be accessed through a few different methods:

- Open the Start menu, and then click Server Manager at the top of the menu.
- Open the Start menu, and then choose Administrative Tools | Server Manager.
- Open the Start menu, right-click Computer, and then choose Manage.
Open the Start menu and click Control Panel. Double-click Administrative Tools, and you’ll see Server Manager there.

Click the Server Manager icon in the Quick Launch bar next to the Start button.

Open the Start menu and click Run. Type `mmc`, and then click OK. This will open a blank console. Choose File | Add/Remove Snap-in, and then choose Server Manager from the Available Snap-ins list. Click the Add button to add it to your Selected Snap-ins list, as shown in Figure 3-2, and then click OK.
Roles versus Features

Whenever you talk about Windows Server 2008, it’s impossible not to refer to roles and features. You need to understand exactly what these roles and features are. Roles inherently describe the primary function of a server. Features describe a supporting function of a server that typically augments the functionality of a role. Although it is possible to have a server perform only one role, every server can perform multiple roles, and you’re realistically limited only by the capacity of the server. For a small environment, having a server host multiple roles may be your only option, since servers are limited; for larger enterprises, you may want to spread your roles among multiple servers to ensure maximum performance and reliability. In addition, for each role, you should follow best practices—for example, if the server is going to perform the role of Active Directory Domain Services (AD DS), you probably don’t want it to host SharePoint as well, since you won’t want to compromise the security of your domain controllers.
Windows Server 2008 supports 16 roles and 35 features that can be managed using Server Manager. Some of the features, such as Failover Clustering and BitLocker Drive Encryption, require supporting hardware. Also, certain roles require other roles to be installed as well. For example, if you want to install the SharePoint role, you must install the Internet Information Services (IIS) role as well.

The following roles are supported:

- Active Directory Certificate Services
- Active Directory Federation Services (AD FS)
- Active Directory Rights Management Services (AD RMS)
- DHCP Server
- Fax Server
- Network Policy and Access Services
- Terminal Services
- Web Server (IIS)

The following features are supported:

- NET Framework 3.0 Features
- BitLocker Drive Encryption
- Desktop Experience
- Group Policy Management
- Internet Storage Naming Server
- Message Queuing
- Network Load Balancing
- Active Directory Domain Services (AD DS)
- Active Directory Lightweight Directory Services (AD LDS)
- Application Server
- DNS Server
- File Services
- Print Services
- Universal Description, Discovery, and Integration (UDDI) Services
- Windows Deployment Services
- BITS Server Extensions
- Connection Manager Administration Kit
- Failover Clustering
- Internet Printing Client
- LPR Port Monitor
- Multipath IO
- Peer Name Resolution Protocol

(Continued)
### Server Manager Elements

Because Server Manager operates like a portal, it relies on tying in multiple elements to make for a simplified user experience. The goal here is to reduce the number of clicks you need to make to get the job done. All but two of these elements are wizards that walk you through the addition or removal of a server role or feature. Table 3-1 lists each of these components and its function within Server Manager.

<table>
<thead>
<tr>
<th>Element</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality Windows Audio-Video Experience</td>
<td></td>
</tr>
<tr>
<td>Remote Differential Compression</td>
<td></td>
</tr>
<tr>
<td>Removable Storage Manager</td>
<td></td>
</tr>
<tr>
<td>Simple TCP/IP Services</td>
<td></td>
</tr>
<tr>
<td>SMTP Server</td>
<td></td>
</tr>
<tr>
<td>Subsystem for UNIX-based Applications</td>
<td></td>
</tr>
<tr>
<td>Telnet Server</td>
<td></td>
</tr>
<tr>
<td>Windows Internal Database</td>
<td></td>
</tr>
<tr>
<td>Windows PowerShell</td>
<td></td>
</tr>
<tr>
<td>Windows Server Backup Features</td>
<td></td>
</tr>
<tr>
<td>WINS Server</td>
<td></td>
</tr>
<tr>
<td>Remote Assistance</td>
<td></td>
</tr>
<tr>
<td>Remote Server Administration Tools</td>
<td></td>
</tr>
<tr>
<td>RPC over HTTP Proxy</td>
<td></td>
</tr>
<tr>
<td>SNMP Services</td>
<td></td>
</tr>
<tr>
<td>Storage Manager for SANs</td>
<td></td>
</tr>
<tr>
<td>Telnet Client</td>
<td></td>
</tr>
<tr>
<td>TFTP Client</td>
<td></td>
</tr>
<tr>
<td>Wireless LAN Services</td>
<td></td>
</tr>
<tr>
<td>Windows Process Activation Service</td>
<td></td>
</tr>
<tr>
<td>Windows System Resource Manager</td>
<td></td>
</tr>
<tr>
<td>Element</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Initial Configuration Tasks screen</td>
<td>I lied to you a little earlier when I said that Server Manager launches after the Initial Configuration Tasks screen has been completed. Technically, Server Manager is already running, since the Initial Configuration Tasks screen is actually an element of Server Manager. The Initial Configuration Tasks screen looks different and represents a small subset of specialized links and tasks that Server Manager can perform. This dual interface was created so that an administrator could focus on the critical key configuration tasks that need to be performed after the installation of Windows Server 2008.</td>
</tr>
<tr>
<td>Add Roles Wizard</td>
<td>This wizard helps you add roles to your server. If dependent roles or features exist for any of the roles you choose to add, this wizard will also inform you of those dependencies and help you install them.</td>
</tr>
<tr>
<td>Add Role Services Wizard</td>
<td>Some roles have subroles, called <em>role services</em>, which you can install after the primary role has been installed. For example, for a File Services role, you can also add File Replication Service (FRS). This wizard will walk you through the installation of those role services.</td>
</tr>
<tr>
<td>Add Features Wizard</td>
<td>This wizard helps you add features to your server, similar to how the Add Roles Wizard helps you add roles to your server.</td>
</tr>
<tr>
<td>Remove Role Wizard</td>
<td>This wizard guides you through the removal of roles from your system.</td>
</tr>
<tr>
<td>Remove Role Services Wizard</td>
<td>This wizard guides you through the removal of role services from your system.</td>
</tr>
</tbody>
</table>

Table 3-1. Server Manager Elements
SERVER MANAGER CONSOLE

So far, we’ve gone over some major highlights of Server Manager’s capabilities. Let’s explore Server Manager to see how effective it is at accomplishing its primary purpose, which is to simplify server administration. Upon starting up Server Manager, you’ll see a screen that summarizes your server’s general configuration. Four major sections are presented in this main page: Server Summary, Roles Summary, Features Summary, and Resources and Support.

Server Summary

The Server Summary presents some information you saw as part of the Initial Configuration Tasks screen. It shows the computer name, the domain, the network interfaces on the PC and how they are assigned an IP address, the status of Remote Desktop, and the product ID. It also shows the firewall and Windows Update status and whether Internet Explorer (IE) Enhanced Security Configuration (ESC) is enabled for administrators or users. IE ESC helps reduce the exposure of your server to attacks from web-based content. As is the philosophy behind Server Manager, for every piece of information it presents that is configurable, it should also provide you with a means of making those changes without leaving the comfort of this user interface.

Just to the right of the system information are links to change the system properties and the administrator account, as well as links to view network connections, so they can be configured, and to configure Remote Desktop. Clicking the Change System Properties link opens the same System Properties dialog box that appears when you click Start,
right-click My Computer, and choose Properties. Here you can update the computer description, change the computer name and domain membership, and access additional tabs for managing devices, adjusting system performance, and managing user profiles, startup and recovery settings, and remote control preferences. If you click the Change Administrator Account link, you can rename and set the password for the local Administrator account. The View Network Connections link opens the Network Connections Control Panel applet, where you can configure your network interfaces such as setting static IP addresses, protocol bindings, and bridging connections. The Configure Remote Desktop link is a shortcut to the Remote tab of the System Properties dialog box, where you can allow or disallow incoming Remote Desktop connections.

The Do Not Show Me This Console at Logon checkbox at the bottom of the Server Summary section prevents Server Manager from running automatically whenever you log on to the server.

**Security Information**

In the Security Information section, click the Go to Windows Firewall link to open the Windows Firewall configuration screen, where you can enable or disable the firewall, set exceptions, and configure advanced settings. The Configure Updates link lets you configure how Windows Update is handled by the server. The Run Security Configuration Wizard link launches a wizard that you can use to create or import a security policy for the server. This is useful if you use standard security templates across your server builds. The last link, Configure IE ESC, allows you to enable or disable IE ESC for administrators or users on the server.

**Roles Summary**

The Roles Summary lists all the roles currently installed on the server. If a problem exists with a particular role—for example, if a dependent service is stopped or critical errors are in the event log pertaining to that service—the status will be indicated next to the role name. You can think of the Roles Summary as a server roles health-check page. Clicking the role name will take you to the appropriate role management home page. Three links are to the right of the Roles Summary section. The Go to Manage Roles link (which you can see at the bottom of Figure 3-1) takes you to a more detailed overview format for all the roles installed on the server, as shown in Figure 3-3. The Add Roles link opens the Add Roles Wizard, which you can use to install one or more roles to the server. Conversely, the Remove Roles link takes you to the Remove Roles Wizard that you can use to remove one or more roles from the server.

**Features Summary**

The Features Summary section works similarly to the Roles Summary section, except that it pertains to features rather than roles. It displays the list of all installed features. Since features aren’t really managed in the same way as roles, there is no Manage Features link. Instead, this section has only two links—one to add features and another to remove features. These links launch the Add Features Wizard or Remove Features Wizard.
Resources and Support

This section allows you to configure your participation in both the Customer Experience Improvement Program (CEIP) and Windows Error Reporting. You have links to opt into these programs, or, if you have already opted in, you can change the status to opt out. You are also provided links to access the Windows Server TechCenter to browse technical resources such as documentation and webcasts.

SERVER MANAGER SNAP-INS

On the left side of the Server Manager screen is a tree view of Server Manager snap-ins. These are grouped together into five major categories: Roles, Features, Diagnostics, Configuration, and Storage. Since Server Manager also replaces the Computer Management snap-in from previous Windows versions, the tools previously under Computer Management have been incorporated into Server Manager. Following the idea that Server
Manager should be the one-stop shop for all server management tasks, almost everything you need to configure your server, from managing local users and groups to performance diagnostics and role management, can be accessed from this view.

**Roles Snap-In**

The Roles snap-in can be accessed by clicking Roles in the tree view on the left side of the Server Manager screen and by clicking the Go to Manage Roles link from the Server Manager main page (Figure 3-1). As you can see in Figure 3-3, this page provides a high-level summary of the status of each installed service along with added details regarding applicable role services. The Roles Summary section of this snap-in functions about the same way it functions on the main page. It provides a list of all installed roles as well as links to the Add Roles and Remove Roles wizards.

In addition, for every role installed, an appropriate summary of that role is displayed in the Roles Summary section. It provides a Role Status, where the service status is displayed, such as the number of services that are either stopped or started or whether any event log entries corresponding to that role might require attention. If an additional snap-in is available to manage that particular role, a link to access that snap-in is available to the right of the Role Status area. This is followed by a list of Role Services applicable to that role, followed by its installation status, indicating whether or not that particular role service is installed. The Roles Summary section also provides links to the Add Role Services and Remove Role Services wizards.

If you expand the Roles node in the tree view, you will see a list of child snap-ins that can be used to manage each installed role. For example, you can access settings for the File Services role by expanding the Roles entry (clicking the plus sign) and clicking File Services. This opens that role-specific snap-in page. Figure 3-4 shows the File Services snap-in page. As you may notice, the more you drill down into each snap-in, the more detailed the information presented. For example, rather than display only the number of services that are either stopped or started for this role, the File Services snap-in page shows specifically the name of the service, its status, and its startup type. You can then use the links to the right either to stop or start these specific services or just open the general Server Manager main window, which you may have to do if you want to make some configuration changes, such as changing the startup type or specifying an alternative credential for running the service.

The Role Services list and management shortcuts are available on this page as well. At the bottom of this page, you are also presented with Resources and Support links to help you access context-sensitive help or other related TechCenter links.

If you expand the Roles snap-in entry in the tree view, you can also access additional tools to configure that role. For example, for the File Services role, you can access the Shared Folders menu, where you can view shares, sessions, and open files as well as manage your file shares and perform disk management. For this particular role, its functionality combines everything related to File Services and any Role Services you installed that are part of this role, including features of the Shared Folders and Disk Management system tools that were available under the Computer Management snap-in in previous versions of Windows Server.
Features Snap-In

The Features snap-in provides a consolidated view into all the features installed on the server. Each installed feature and subfeature is listed here. You can either add or remove features by selecting the appropriate link in the upper-right corner to launch the Add Feature or Remove Feature Wizard.

Diagnostics Snap-In

The Diagnostics snap-in brings together the Event Viewer, Service Manager, Reliability and Performance tools, and Device Manager that were available in the old Computer Management console. The Event Viewer has been improved significantly. You can still access your typical Windows event logs, but application- and service-specific logs are newly added. In addition to being able to view your Windows and application event logs, you can also create custom views. This goes far beyond the filters that were available in previous versions to filter through event logs. Custom views allow you to consolidate queries
across one or more Windows logs and define criteria to refine what is being displayed. Each of these views can then be saved for future viewing. The snap-in also supplies a new subscription feature that lets you subscribe to event logs of other servers and then make them available via a local log, which by default is the ForwardedEvents Log. This is extremely useful if you want a centralized location for viewing events from multiple servers.

The Services node functions the same as the services MMC snap-in, so you can stop and start services as well as configure services from here. The Device Manager is exactly the same Device Manager that has always existed, except now it is neatly organized under the Diagnostics snap-in for easy access. Lastly, a number of performance diagnostic tools are available under the Reliability and Performance node. Reliability and performance monitoring is covered in greater detail in Chapter 7.

**Hands-On Exercise: Creating a Custom Event Log View**

In this exercise, we will create a custom view to display all Critical and Error event levels for the past seven days from the System and Security Windows logs.

1. Open Server Manager if you don’t already have it open.
2. Expand the Diagnostics node.
3. Expand the Event Viewer node.
4. Right-click Custom Views.
5. Select Create Custom View.
6. From the Logged drop-down list, select Last 7 Days.
7. Under Event Level, check both the Critical and Error checkboxes.
8. From the Event Logs drop-down list, expand the Windows Logs tree and check the Security and System checkboxes (Figure 3-5). Make sure you don’t just check the Windows Logs checkbox or all the checkboxes will be selected: Application, Security, Setup, System, and Forwarded Events.
9. Click OK.
10. In the Save Filter to Custom View dialog box (Figure 3-6), enter **System and Security Events** in the Name field.
11. Enter **Critical and Error messages from the System and Security Event Logs** in the Description field.
12. Click OK.
13. The custom view will be selected and the results displayed in the right pane, as shown in Figure 3-7.
Enabling Windows Remote Management

If you’re creating a subscription on a Windows Server 2008 instance that is not part of a domain and you are subscribing yourself, you will first need to enable Windows Remote Management (WRM) and then add your computer name to the list of trusted hosts using the following sequence of commands:

```
winrm quickconfig -q
winrm set wimrm/config/client @{TrustedHosts="%COMPUTERNAME%"}
```

Note that a reboot is required for this new configuration to take effect.
Figure 3-6. Save Filter to Custom View dialog box

Figure 3-7. Results of the newly created custom view
Hands-On Exercise: Creating an Event Log Subscription

Subscriptions are a good way to centralize event management. By subscribing to multiple server event logs, you need to look at only one central location to view all log entries in which you are interested, rather than having to connect to each server individually. For this exercise, we’re going to simplify things by subscribing to the computer we are working on rather than a different computer.

1. Open Server Manager if you don’t already have it open.
2. Expand the Diagnostics node.
3. Expand the Event Viewer node.
4. Right-click Subscriptions. You may see a warning that the Windows Event Collector service is not running, asking if you would like to start it. Click Yes.
5. Select Create Subscription.
6. Enter My Custom Subscription in the Subscription Name field.
7. Enter First attempt at creating a subscription in the Description field.
8. Click the Add button.
9. Enter the name of your server. In my case, my computer name is WIN2K8SRV1.
10. Click OK.
11. At this point, the screen should look like Figure 3-8.
12. Click the Test button to ensure that connectivity is successful. If it is successful, a confirmation is displayed on the screen (Figure 3-9).
13. Click OK to close the confirmation dialog box.
14. In the Events to Collect area, click the Select Events button.
15. Select Last 7 Days from the Logged drop-down menu.
16. Check all the events-level checkboxes.
17. Select the Security and System Windows Event Logs from the Event Log drop-down menu.
18. Click OK on the Query Filter dialog box to save the filter.
19. In the Subscription Properties dialog box, click the Advanced button.
21. Click the User and Password button. In my case, I entered WIN2K8SRV1\Administrator as the username and the password for the Administrator account. Click OK to close the Advanced Subscription Settings dialog box.
22. Click OK in the Subscription Properties dialog box to save the subscription. The subscription should now show a status of Active.

23. If you click the ForwardedEvents log, you should see all the events from the System and Security Windows logs, as shown in Figure 3-10.

**Configuration Snap-In**

The Configuration snap-in allows you to access the Local Users and Groups system tool. Three additional tools are available, namely the Task Scheduler, Windows Firewall with Advanced Security, and WMI Control. (I was happy to see that Microsoft finally moved the Task Scheduler into a more logical location rather than having you navigate to the Windows folder using Explorer to access it. It is also significantly improved in functionality.) The new Windows Firewall with Advanced Security snap-in provides an easy-to-use graphical interface for managing your inbound and outbound firewall rules as well as monitoring the overall firewall usage. Finally, WMI Control can be used to manage your Windows Management Interface (WMI) service.

![Subscription Properties dialog box](image)

**Figure 3-8.** Subscription Properties dialog box
Task Scheduler

Not only has Task Scheduler been added to Server Manager, but it has undergone a serious facelift and is much richer in functionality than previous iterations of this product. The first thing you see when you click the Task Scheduler is a Task Scheduler summary, as shown in Figure 3-11.

The two main sections of the Task Scheduler summary are the Task Status and the Active Tasks sections. By default, the Task Status section displays the tasks that were executed in the last 24 hours along with their status (running, succeeded, failed). The drop-down list allows you to change the time interval for the tasks to be displayed from what happened in the last hour all the way to the last 30 days. You can expand the name of each task on this list to get additional details, such as the task result status, start time, and end time. If the task ran multiple times over the period you selected, each run of the task is listed individually under the main task name, where only the last run status is displayed along with the time it completed.

The Active Tasks section lists every task that is currently active on the server. The summary includes the name of the task, the next time it will run, as well as any triggers
that initiate it. The triggers concept extends the simple concept of running tasks at scheduled intervals. You can still configure a task to run at a specific schedule, but now you have the added flexibility of executing a task during a multitude of other events, such as when the computer is idle or when a workstation is locked or unlocked.

Out of the box, Microsoft has included and configured a number of tasks. They are all neatly organized under several folders based on their purpose. If you expand the Task Scheduler node, you will see the task folders within the Task Scheduler Library folder, as shown in Figure 3-12. Within the Microsoft parent folder are groups of tasks in folders relating to specific services and functions within the server—such as Defrag, Multimedia, Tcpip, Windows Error Reporting, and so on. You can create new tasks in any of these folders. However, it’s probably best practice to create your own folder to organize your tasks so that you can easily identify your tasks from Microsoft’s built-in tasks.

Each task contains a number of properties that are organized into tabs: General, Triggers, Actions, Conditions, Settings, and History.
General Tab  The General tab contains the name, author, and description of the task. It is also where the security options of the task are configured. You can run the task under one of three different security contexts:

- **Run Only When a User Is Logged On**
  - The task will run under the logged-on user’s credentials.

- **Run Whether User Is Logged On or Not**
  - You must specify the user account under which this task will run. Optionally, you can tell Task Manager not to store the password, in which case it can have access only to local computer resources.

- **Run with Highest Privileges**
  - Lets the task do whatever it wants locally.
Chapter 3: Server Manager

You also have the option of running a task as hidden, and you can configure it to run for a specific OS compatibility.

**Triggers Tab** Each task on this tab is designed to run based on one or more triggers. In its simplest form, a task can be run based on a specific schedule. You can also use other events such as logon or startup events to trigger a task to run. The options to begin a task are described in the table that follows.
On a Schedule

The basic form of task trigger. You can specify the exact schedule for when to run this task, from one time only, to daily, weekly, or even monthly. You can also specify the start time and recurrence.

At login

Runs the task when a login occurs. You can select whether to run this at the logon of any user or just a specific user or members of a specific group.

At Startup

Runs when the server starts up.

On Idle

Runs when the system is idle. Use the Conditions tab to specify additional parameters for this option.

On an Event

This has a lot of potential. It triggers a task based on an event in the event log. You can select the log file to query, the source, and the event ID, or you can create your own custom event filter.

At Task Creation/Modification

Triggers an action whenever a task is created or modified.

On Connection to User Session

Runs the task when a connection to a user session is initiated. Can be configured to run when any user, a specific user, or members of a specific group connect. In addition, can be set to run whether the connection is remote or local.

On Disconnect from User Session

Similar to On Connection to User Session, except this trigger runs when the user disconnects.

On Workstation Lock

Runs the task when the workstation is locked. Can be configured to run when any user locks the workstation or when a specific user or members of a specific group of users lock the workstation.

On Workstation Unlock

Similar to On Workstation Lock except it occurs when the workstation is unlocked.

In addition to these initiating triggers, some advanced settings can be set that delay the start of the task by a period of time, repeat the task for a given time period, or stop a task after it has been running longer than a certain amount of time. You also have the option of setting the date and time when this task will automatically activate or expire.

**Actions Tab**

Every task can have one or more actions associated with it. The Task Scheduler can either start a program of your choice (including the ability to pass arguments), send an e-mail to a given SMTP server, or display a message on the server. If you have defined multiple tasks, you can also set the order in which these tasks are executed by using the up and down arrow buttons at the right of the Actions tab to move each task above or below another task. The tasks are executed from the top down.
Chapter 3: Server Manager

Conditions Tab In addition to the triggers, you can also apply certain conditions to control execution. If you select Run On Idle on the Triggers tab, you can set the condition as to how long the computer has to be idle before executing and also whether or not to stop the task if the computer ceases to be idle. You can also configure the task to run only if the computer is on AC power and to stop if it suddenly switches to battery power. The task can also wake the computer to run. It also offers a condition for checking whether any network connection or a specific network connection is available before continuing.

Settings Tab The Settings tab includes some additional options relating to the behavior of the task:

▼ Allow the task to be run on demand.
■ Run the task as soon as possible after a scheduled start is missed.
■ If the task fails, restart every X time period and set the number of retries.
■ Stop the task if it runs longer than X time period.
■ If the task does not end when requested, force it to stop.
■ If the task is not scheduled to run again, delete it after X time period.
▲ If the task is already running, the following rules can be selected to apply:
■ Do not start a new instance.
■ Run a new instance in parallel.
■ Queue a new instance.
■ Stop the existing instance.

History Tab The History tab looks at the Windows event logs and reports on the history of the tasks. It lists the log entries that indicate when the task has been triggered and when the task has stopped.

Hands-On Exercise: Creating a Task Using Task Scheduler

1. Open Server Manager if you don’t already have it open.
2. Expand the Configuration node.
3. Expand the Task Scheduler node.
4. Select the Task Scheduler Library folder.
5. Right-click Task Scheduler Library.
7. Enter My Custom Tasks as the name for the new folder, and then click OK.
8. Select the newly created My Custom Tasks folder.
9. Right-click the My Custom Tasks folder.
10. Select Create Task.
11. Enter **Display Message** in the Name field.
12. Enter **Displays a message on Windows unlock** in the Description field.
13. Under Security options, select Run Only When User Is Logged On. The General tab should now look like Figure 3-13.
14. Click the Triggers tab.
15. Click the New button to create a new task trigger.
16. Select On Workstation Unlock in the Begin the Task drop-down list, as shown in Figure 3-14.
17. Click OK to save the new trigger.

**Figure 3-13.** Completed General tab
18. Click the Actions tab.
19. Click the New button to create a new Action.
20. Select Display a Message under the Action drop-down list.
21. Enter Unlock Message in the Title field.
22. Enter You have unlocked your session! in the Message field.
23. Click OK to save the new action, as shown in Figure 3-15.
24. Click the Conditions tab and leave everything in its default state, as shown in Figure 3-16.
25. Click the Settings tab and leave everything in its default state, as shown in Figure 3-17.

26. Click OK to save this new task.

27. To test your new task, press CTRL-ALT-DEL and select Lock This Computer.

28. Press CTRL-ALT-DEL again and enter the password for your account; then press ENTER.

29. As expected, a message dialog box pops up with the message you specified in the task, as shown in Figure 3-18.
Chapter 3: Server Manager

Windows Firewall with Advanced Security

The Windows Firewall has evolved tremendously from the very basic inbound firewall that started with Windows XP Service Pack 2. The new Windows Firewall included in Windows Server 2008 is appropriately called Windows Firewall with Advanced Security because it is more than just a bidirectional stateful firewall. It is now also fully integrated with Internet Protocol Security (IPSec). Beyond regular IP traffic-filtering rules, the Windows Firewall is also responsible for Windows Service Hardening, is network location aware, has the ability to create authenticated bypasses, offers tight integration with Active Directory users, features computers and groups, and offers IPv6 support. The Windows Firewall is by far no replacement for a true dedicated firewall to segment your network, but a host-based firewall such as this can be used as an additional layer of security for your server.

Figure 3-16. Completed Conditions tab
Figure 3-17. Completed Settings tab

Figure 3-18. Message displayed after unlocking the session
The features of the Windows Firewall with Advanced Security are as follows:

**Windows Service Hardening**

These rules define what a service can or can’t do in relation to the local system. For example, you can restrict a service from writing to the file system or registry.

**Inbound/Outbound Filtering**

You can define very granular rules regarding both inbound and outbound connections. You have the option to block all inbound or outbound connections outright or define specifically what kind of traffic is allowed to come into or out of the server. This includes support for filtering by protocol and also by application.

**Location-Aware Profiles**

The firewall can define different rules based on where a network interface is connected. This is done through one of three firewall profiles:

- **Domain** Used when the server is connected to a network where the Active Directory domain to which the computer is a member can be accessed.
- **Private** Used when a computer is connected to a private network behind a private gateway or router. You must have administrative privileges to configure a network as Private.
- **Public** Used when the server is connected to an interface that is directly connected to the Internet or a network that is neither Private nor Domain.

**Authenticated Bypass**

Allows you to define bypass rules for authenticated computers. For example, you can block all inbound HTTP traffic but allow an authenticated computer to bypass this restriction.

**Active Directory user, computer, and group integration**

If the server is a member of an Active Directory domain, you can define rules around user and computer accounts as well as security groups. This requires authentication to be secured using IPSec with a protocol such as Kerberos version 5.

**IPv6 support**

Overall, Windows Server 2008 supports IPv6, so it makes sense to extend the Windows Firewall to support IPv6.
With all these options for defining rules around the Windows Firewall, some defined order must allow them to be evaluated so that it is clear which rules take precedence over other rules. Essentially six different types of rules can be defined for the Windows Firewall:

- **Windows Service Hardening** - Restrict specific services from establishing connections.
- **Connection Security Rules** - Define how and when a computer authenticates using IPSec.
- **Authenticated Bypass Rules** - Allow connections from particular computers that are authenticated via IPSec. These connections are allowed regardless of any block rule preventing access.
- **Block Rules** - Explicitly prevent a type of inbound or outbound traffic.
- **Allow Rules** - Explicitly allow a type of inbound or outbound traffic.
- **Default Rules** - The general catch-all rule if nothing else applies. By default, inbound connections are blocked and outbound connections are allowed.

These rule types are processed in the specific order shown in the table and in Figure 3-19. It’s important that you understand this sequence, since you will undoubtedly need it to troubleshoot connectivity problems. It’s tempting to disable the firewall, especially in a relatively enclosed and secure environment, but it really is a good idea to leave it on and create rules to allow exceptions rather than flat-out disable it and leave your server wide open. It might be more aggravating to set up Windows Firewall initially, but in the long run, the added layer of security can help mitigate certain risks.

You can manage the Windows Firewall using the Windows Firewall with Advanced Security MMC snap-in, which is incorporated into Server Manager, or you can use `netsh` as you did with the Server Core installation in Chapter 2. Certain rules can also be defined using Group Policy. For now, let’s focus on the MMC snap-in that is available in Server Manager.

When you click the Windows Firewall with Advanced Security snap-in, a summary pane is displayed in the middle of the Server Manager console, as shown in Figure 3-20. At the top of this pane is an Overview section listing the status of each of the three connection profiles. It indicates which profile is active along with the state of the firewall under each profile and whether inbound or outbound connections are allowed or blocked by default.

To make changes to these connection-based profiles, click the Windows Firewall Properties link at the bottom of the Overview section. You will see a tab for each of the connection profiles and an additional tab for IPSec Settings. Each of the tabs allows you
to change the state of the firewall for that profile. If you turn on the firewall, you then have
the option of setting the general inbound or outbound connection rules (Figure 3-21). For
Inbound Connections, you can select Block (Default), Block All Connections, or Allow
(see the following table). For Outbound Connections, you can select Allow (Default) or
Block. You can also change settings to control the Windows Firewall behavior. You can
notify the logged on user when inbound connections are blocked, and you can allow
unicast responses to multicast or broadcast requests sent out from the server. Lastly, you
can customize the logging option such as the name of the log file, the size of the log file,
and whether to log dropped packets and successful connections.

<table>
<thead>
<tr>
<th>State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block</td>
<td>Blocks connections that don’t match any active</td>
</tr>
<tr>
<td></td>
<td>firewall rules</td>
</tr>
<tr>
<td>Block All Connections</td>
<td>Blocks all inbound connections regardless of firewall</td>
</tr>
<tr>
<td></td>
<td>rules</td>
</tr>
<tr>
<td>Allow</td>
<td>Allows connections that don’t match any active</td>
</tr>
<tr>
<td></td>
<td>firewall rules</td>
</tr>
</tbody>
</table>

The IPSec Settings tab lets you configure Key Exchange, Data Protection, and Au-
thentication Method settings for IPSec. Usually, you will want to keep these at the de-
fault settings unless you have very specific requirements for your IPSec environment.
The Getting Started section duplicates the links already available if you expand the Windows Firewall with Advanced Security tree view node on the Server Manager navigational tree. These are links to define Inbound and Outbound Rules, to configure Connection Security Rules, and to monitor the existing policies and connections. The Resources section at the bottom of the entire screen is a set of handy links to various resources related to the Windows Firewall, such as best practices and troubleshooting guides.
Chapter 3: Server Manager

Inbound and Outbound Rules

The Inbound Rules define exactly what inbound connections are allowed or disallowed. Outbound Rules share the exact same set of properties as Inbound Rules, except the rules relate to outbound traffic. These rules allow for a granular definition of access, from simple port restrictions all the way to protocol- or application-based rules restricted by connection profiles. Out of the box, Microsoft provides a great deal of built-in rules, as shown in Figure 3-22; some are enabled while others aren’t. To enable or disable a rule, right-click the rule name and select Enable Rule or Disable Rule from the pop-up menu. Double-clicking a rule reveals that rule’s properties.

Figure 3-21. Windows Firewall with Advanced Security properties dialog box
Each rule’s property dialog box contains six tabs: General, Programs and Services, Users and Computers, Protocols and Ports, Scope, and Advanced, as shown in the following table. Each tab defines a number of properties that define each rule.

<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Allows you to define the name and description of the rule along with a checkbox to enable or disable it. You can specify whether this rule allows all connections, allows only secure connections, or blocks connections. For secure connections, you have the added options to require encryption and to have this override Block Rules.</td>
</tr>
<tr>
<td>Programs and Services</td>
<td>Lets you tie this rule to all programs or to a specific program. It can also be used to apply the rule to any process, only services, specific services, or a custom service short name. You would use this tab to be able to begin defining Service Hardening Rules.</td>
</tr>
<tr>
<td>Users and Computers</td>
<td>Lets you restrict access based on specified authorized computers, users, or security groups. The prerequisite for this is that the Allow Only Secure Connections checkbox must also be selected in the General tab. You can specify more than one computer, user, or group by clicking the Add button in each of the respective sections.</td>
</tr>
<tr>
<td>Protocols and Ports</td>
<td>Your bread-and-butter tab when it comes to firewalls allows you to control access based on protocol and port or the special Dynamic RPC, RPC Endpoint Mapper, or Edge Traversal keywords. This is typically the way you would control access on a traditional firewall. For most third-party applications, you will be given guidelines on what ports need to be open for the application to work correctly. This is one way to apply those guidelines on your server.</td>
</tr>
<tr>
<td>Scope</td>
<td>Lets you define to which local and remote IP addresses this rule applies. By default it applies to any local and any remote IP addresses. To specify a particular IP address or range of IP addresses, click the Custom radio button and add the IP addresses to the list. This works for both IPv4 and IPv6 addresses.</td>
</tr>
<tr>
<td>Advanced</td>
<td>Lets you define this rule under all profiles or explicitly define to which profile this rule will apply. You’re not restricted to applying this only to one connection profile. To apply this to multiple connection profiles, simply check the corresponding checkbox next to the profile name. You can also tie this rule to specific interface types. Your choices are either all interface types or specific interface types: Local Area Network, Remote Access, or Wireless.</td>
</tr>
</tbody>
</table>
Hands-On Exercise: Creating a New Inbound Rule  In this exercise, we will create an inbound rule to allow traffic to a Web server only when connected to the domain by allowing TCP connections to port 80 for the domain connection profile.

1. Open Server Manager if you don’t already have it open.
2. Expand the Configuration node.
3. Expand the Windows Firewall with Advanced Security snap-in.
4. Select Inbound Rules.
5. Right-click Inbound Rules.
7. Under Rule Type, select Port, and then click Next (Figure 3-23).
8. Select TCP and enter 80 in the Specific Local Ports field; then click Next (Figure 3-24).

9. In the Action screen, select Allow the Connection (selected by default); then click Next.

10. In the Profile screen, uncheck all checkboxes except Domain, and then click Next (Figure 3-25).
11. Enter **Allow Inbound HTTP Traffic** in the Name field and leave the Description field blank. Then click Finish (Figure 3-26).

12. This will now create the new inbound rule and immediately enable it. If you want to disable it, you can always right-click the rule and choose Disable.
Computer Connection Security  Connection Security settings are used by IPSec to negotiate secure connections between hosts. No connection security settings are defined by default, so you’ll have to create settings if you want to create rules that are restricted to secure connections only. IPSec then uses these rules to determine how to secure the host information between two computers. When you create a new Authentication Rule, it can be of one of five types: Isolation, Authentication Exemption, Server to Server, Tunnel, or Custom:

Isolation  Creates a virtually isolated environment regardless of physical connectivity. Select this rule type when implementing a domain isolation strategy. You need to specify when authentication will be requested.
or required as well as the method of authentication. Typically this is
done via Kerberos or some kind of computer certificate from a specific
certification authority (CA).

Authentication Exemption Specifies connections that do not require authentication. For example,
exempt all hosts from a particular subnet from requiring authentication.
You must provide a list of exempt computers.

Server to Server Protects communications between two specific servers. You must
specify the endpoints and the authentication method.

Tunnel Protects communications between two gateway computers and is
typically implemented in securing connections across the Internet
between security gateways. You must specify the tunnel endpoints and
authentication method.

Custom If none of the other authentication rules fulfill your requirements, use a
Custom rule to specify the rule parameters manually.
Firewall Monitoring  Windows Firewall with Advanced Security includes its own monitoring tools for Firewall and Connection Security. Clicking the Firewall Monitoring snap-in displays the currently active firewall profile. You can then drill down to lists of all active firewall rules, Connection Security rules, and Security Associations. This is based on your currently active connection profile, connection interface type, and any policies pushed down through Group Policy. This is a good place to start when trying to determine whether any firewall rules are preventing either inbound or outbound connections to and from your server. The Connection Security Monitoring snap-in lists all currently enabled connection security rules.

WMI Control
The WMI Control snap-in links you to the tool to configure and control the WMI service. To make changes, right-click the WMI Control snap-in and select Properties. You can then use the four-tab interface of the WMI Control Properties dialog box to back up or restore the WMI repository on the server and assign security to any WMI namespace you want. You can also set the default namespace used when a script connects to the WMI provider without specifying a namespace.

Local Users and Group/Device Manager
Nothing’s really changed with these two system tools other than they’ve been relocated from the System Tools menu in Computer Management and incorporated into Server Manager under the Configuration snap-in. Use the Local Users and Groups snap-in to manage your local users and groups. Device Manager can be used to check for any hardware issues, such as devices with missing drivers, and to manage devices overall, including disabling devices and updating drivers.

Storage Snap-In
Server Manager incorporates Windows Server Backup and Disk Management linked under the Storage snap-in. If you jumped the gun and clicked Windows Server Backup under the Storage snap-in, you will notice that it doesn’t work—that’s because it’s not installed. Windows Server Backup is actually a Windows Server feature that isn’t installed by default. This was done because many Windows Server administrators use third-party Windows backup tools to manage their backups; and remembering Microsoft’s minimal installation security strategy, it makes sense not to have this automatically installed if it will never be used by many of their clients in the first place.
Chapter 3: Server Manager

Adding the Windows Server Backup Feature
Before you can begin to use the Windows Server Backup snap-in in Server Manager, you will need to add the Windows Server Backup feature using the following procedure:

1. Open Server Manager if you haven’t already done so.
2. If it is not already selected, select the Server Manager (servername) entry at the top of the tree view.
3. Click the Add Features link under the Feature Summary section of the main page.
4. Check the Windows Server Backup checkbox. You will also need to select the Windows Recovery Disk feature as it is required by Windows Server Backup.
5. Click Next.
6. Click Install.
7. After the installation completes, click Close.

Windows Server Backup
If you have the Windows Server Backup feature installed, you can access it via the Windows Server Backup snap-in in Server Manager. When you click this snap-in, you will be presented with a summary of the latest backup messages. If you haven’t performed a backup of the server yet, it’s a good idea to back it up now or schedule a backup to take place in the near future.

Hands-On Exercise: Creating a Backup Schedule
In this exercise, we will create a new Backup Schedule to back up our primary Windows partition to a separate hard drive on the same system.

CAUTION
The backup destination disk will be reformatted and all data will be lost during this process, so make sure that the target disk you are going to use doesn’t contain any data you want to keep.
1. Open Server Manager if you don’t already have it open.
2. Expand the Storage node.
4. Right-click the Windows Server Backup snap-in.
5. Select Backup Schedule.
6. Click Next on the Getting Started screen.
7. Click Custom in the Select Backup Items screen, and then click Next.
8. Check only the volume where the operating system is installed, and uncheck everything else; then click Next (Figure 3-27).

![Backup Schedule Wizard](image)

**Figure 3-27.** Selecting the volume to back up
9. Select Once a Day and then select an appropriate time for the backups to occur. Then click Next.

10. The Specify Target Disk screen will show all backup devices the system can back up. You can also back up to a separate hard drive on the same system, which is what we will do in this example. Click the Show All button.

11. Check the box next to the volume to which you want to back up, as shown in Figure 3-28. Then click OK.

12. Check the box next to the volume under Available Disk, and then click Next.

13. You will be warned that the data on the selected target disk will be lost. Click Yes to proceed.

14. Click Next on the Label Target Disk screen.

Figure 3-28. Selecting the destination backup volume
15. Click Finish on the summary page. This will format the target disk and prepare it for the backup.

16. Once you receive the confirmation that the backup has been scheduled, click Close.

The backup should start at the time you specified.

**Disk Management**

The Disk Management snap-in can be used to create, format, and delete volumes. A new feature is the ability to shrink a volume in addition to extending a volume. All you need to do is right-click the partition you want to shrink and choose Shrink Volume from the pop-up menu. The amount you can shrink the volume is dependent on how much free space is available and whether or not snapshots or pagefiles are enabled on the volume. Disk Management can also be used to access the properties of each volume and configure security, sharing, shadow copies, and quotas. It also has your typical links to run Error Checking, Defrag, and Backup tools.

**CHAPTER SUMMARY**

This chapter went over all aspects of Server Manager, from its high-level summaries down to the nitty-gritty details of managing your Windows Firewall. You learned the different ways to access Server Manager and found out that it replaces and pulls together disparate tools and consoles into a unified view. Server Manager is a truly consolidated portal to all your server management needs. Not only does it give you a bird’s eye view of what’s installed on your server and its general health status, but you can use it to act quickly on issues or make changes to your system with very few mouse clicks. Server Manager gives you the ability to perform effective server administration without needing to launch a multitude of tools.

Windows Server 2008 is built around roles, features, and security. You need to understand the difference between roles and features before you can effectively deploy Windows Server 2008 in your environment. We went heavily into the workings of Windows Firewall with Advanced Security. The firewall is now a truly integral part of the Windows Server product, and you should learn it well and leverage it whenever possible. By combining a minimal installation strategy and following good Windows Update and firewall practices, you will be able to create a more stable and secure Windows Server environment for your organization.
CHAPTER 4

Active Directory
Domain Services
Unless you’ve been living on a deserted island for the past few years, you should understand at least the basics of Active Directory. But even if you don’t, you needn’t worry too much, since this chapter begins with a cursory review of Active Directory before it dives into the new stuff. If you’re a pro, you can gladly skip a few sections to get to the real meat.

Windows Server 2008 adds some new functionality to Active Directory as well as an introduction of a concept called a read-only domain controller. If you’ve administered a Windows NT 4.0 network, the first question that probably comes to your mind is “Isn’t that the same as a backup domain controller?” The answer is yes, and no—but we’ll get into that later. What you need to know for now is that Active Directory has evolved and matured significantly since its inception with Windows 2000 Server, and this iteration has a potential for higher availability and recoverability than ever before.

THE BIRTH AND EVOLUTION OF ACTIVE DIRECTORY

When Microsoft got serious about stepping into the backend enterprise computing market in the mid-1990s, the company needed a product that provided some kind of a centralized store to house user, group, and computer account information for Windows NT. It had to be easy to administer and fairly scalable and robust. From that idea came the Windows NT domain model—a predominantly NetBIOS-driven, simple (in terms of functionality), and centralized authentication store. As major enterprises began rolling out this product, they realized that the Windows NT domain model, although fairly scalable on paper, was a nightmare to manage in real life. Scalability demands flexibility, which is why the NT domain model’s lack of flexibility resulted in its subsequent lack of scalability.

In a Windows NT domain, users, groups, and computers are all stored in a flat structure. There is no way to split up users into a more logical hierarchy that follows your own organization’s structure other than through NT groups or, in some cases, multiple domains. Delegating authority over computers, users, and groups was difficult to achieve. Many large companies wound up managing several thousand, or even tens of thousands, of NT groups for each of their domains. In addition, large companies had to manage multiple domains, mostly due to political reasons rather than technical ones (for example, one part of an organization didn’t trust or allow the other part to manage its user accounts). Although I could talk volumes more about the shortcomings of Windows NT, I’ll fast-forward a few years to the development of Windows 2000 and Active Directory.

When Microsoft developers went back to the drawing board for Windows 2000, they realized that the NT domain model was nowhere near where it needed to be in terms of an enterprise directory service. What came out of this development process was Active Directory, which addressed many of the weaknesses in the NT domain model. It was hierarchical, extensible, more secure, and easier to administer. For many of us who had become very comfortable with the NT domain concept, Active Directory was a huge leap forward. Sure it required a bit of a learning curve, but the flexibility and increased reliability made it a natural choice. Active Directory required Windows administrators to change their entire mindset. Rather than just being fixated on simple domains and groups, you could now also organize your domain using organizational units (OUs).
Organizational units allow for a logical division of a directory to make it match your organizational structure and administrative boundaries more directly. The added benefit with this hierarchical structure is that you can granularly delegate authority so that not everyone needs to be a domain administrator to manage various aspects of the directory. For example, you can give human resources staff access to your user account information so they can update user attributes such as addresses and contact information, without giving them rights to modify your account’s Windows group membership.

Active Directory is a Domain Name System (DNS)–dependent service, unlike the NT domain model, which is mostly NetBIOS driven. The hierarchical structure of DNS and the widespread use of TCP/IP as the primary network protocol in many organizations makes Active Directory a natural choice for managing name resolution and service location. Active Directory cannot exist without a functioning DNS service. Although Microsoft does provide and recommend using its own DNS service, you can use another vendor’s service provided it supports SRV records (although you can’t take advantage of all the integrated features that Microsoft’s DNS service provides). With Active Directory, you are also not limited to creating trust relationships between domains. You can set up entire domain trees, and those trees can be combined into forests. This is especially advantageous when setting up relationships between separate organizations, such as between business partners or during mergers and acquisitions.

Since its debut, Active Directory has undergone tremendous changes, many of which were in response to feedback Microsoft received from user communities regarding Active Directory’s performance in real-world scenarios. For example, when Windows Server 2003 was released, it added a host of new functionality to the already existing Windows 2000 Active Directory. This included features that allowed you to rename domain controllers, add them using backup media, rename entire domains, and minimize network traffic by replicating changes only to groups rather than to the entire group membership list whenever a user is added to or removed from it. Windows Server 2008 goes a step further; we will explore all these new features in this chapter, as well as how Windows Server 2008 can fit into your organization.

ACTIVE DIRECTORY PRIMER

Now that we’ve gone over a brief history of Active Directory, I want to spend a moment going over some key Active Directory concepts so that we are all on the same page before focusing on Windows Server 2008–specific Active Directory enhancements. This section answers the following questions:

▼ What is Active Directory?
■ How is Active Directory organized?
■ What role does DNS play in all this?
▲ What are domain functional levels?
If you already know the answers to these questions, you can skip this particular section and jump straight to the Windows Server 2008–specific sections; however, you may want to read this part anyway as a refresher.

**What Is Active Directory?**

Active Directory is a directory service and hierarchical data store that holds information about objects on your network and makes it easy for administrators to manage and search for these objects. That’s a high-level, generic answer to what Active Directory is. In practice, Active Directory serves two purposes: It is the central repository for your account information such as users, groups, and computers. It is also a self-replicating application data store that is implemented through the use of application partitions.

The Active Directory itself is defined by a schema that indicates how each object is represented within the data store. For example, a user object has, among other things, a first name, last name, logon name, e-mail address, and password. If you’re familiar with databases, you should already be familiar with the term *schema* since a database schema refers to the structure of the database in the same way the Active Directory schema defines the Active Directory’s structure.

If you think of Active Directory as a database, then naturally you would expect it to have an index. This is called a *global catalog* (GC), and it stores a subset of the information regarding each object that you can use to search the directory. The information in the GC gets replicated to domain controllers in different sites and even different domains and forests, if that is how your Active Directory architecture is laid out (we’ll get into the Active Directory architecture later in this chapter). Replication is built into Active Directory so that if you’re working with multiple domain controllers, a change to any of them is automatically replicated to the others and is governed by a set of synchronization rules.

Active Directory is *extensible*—that is, the schema that defines how objects look in Active Directory can be modified. For example, if you install Exchange 2003 or a later version into an Active Directory domain, it will modify the schema so that a user object not only contains the standard user information but also information Exchange might want to use, such as the location of a mailbox and additional e-mail addresses tied to the user. Exchange is not the only product that modifies the Active Directory schema. Many products do, and this inherent ability to evolve is exactly what makes Active Directory so flexible and scalable.

One critical aspect of Active Directory is that it is exposed via Lightweight Directory Access Protocol (LDAP). As a functioning LDAP server, Active Directory can interact with any LDAP-compliant application and can be interfaced with other LDAP-compliant systems with relative ease. Although questions exist about Active Directory’s 100-percent compliance with the general LDAP specification, for many administrators and developers, the interfaces available today make it much easier to work with Active Directory as a directory service.
How Is Active Directory Organized?

When I talk about how Active Directory is organized, I am referring to its logical and physical structures. Physically, Active Directory is stored in each domain controller as a set of binary files that represent its underlying database. Logically, you can think of the internal objects of Active Directory as nodes on a tree. This tree analogy lends itself well since the smallest logical administrative boundary for Active Directory is the domain, and a domain tree is a hierarchical collection of one or more domains. It’s important to emphasize one or more, since a tree with only one domain is still a tree, albeit with only one node. An organization of related trees is, not surprisingly, called a forest. Some people get trees and forests confused: They think that two domains automatically equal a forest. What makes a group of two or more different domains a tree or a forest is their direct hierarchy.

Figure 4-1 shows a domain tree. The parent domain, Testlab.local, has a child domain called Engineering.Testlab.local, which has its own child domain called NY.Engineering.Testlab.local. This parent/child relationship forms a tree—you can clearly see by the namespace that Engineering is a branch of Testlab.local and NY is a branch of Engineering. And all these domains are actually part of the Testlab.local domain tree.

![Figure 4-1. A domain tree](image)
Figure 4-2 shows how a forest is formed. Testlab.local and UAT.local are separate, noncontiguous domains and are parents of their own respective domain trees. The existence of a trust relationship between these two otherwise unrelated domains forms a forest, and in doing so, the domains can share resources between them.

**Trusts**

By default, *two-way transitive trusts* are established between domains when you link them together either within a tree or when joining two or more trees to create a forest. When a trust is created, resources in one domain or tree can be assigned access to resources in a different domain or tree. A two-way trust that occurs by default means that resources in both domains participating in the trust can access resources in the other. A *one-way* trust can be established if resources in Domain A need access to resources in Domain B, but you don’t want resources in Domain B to have access to resources in Domain A. A transitive two-way trust means that if Domain A trusts Domain B and Domain B trusts Domain C, then Domain A automatically trusts Domain C (Figure 4-3). This was made the default configuration for Active Directory trusts since it simplifies much of the administration surrounding multidomain trusts.

**Organizational Units**

Using a domain as the smallest logical administrative boundary makes sense since Microsoft needed to provide a direct and easy migration path to allow customers to
transition from the old NT domain model to the new Active Directory model. However, unlike the old NT domain, the Active Directory domain also supports internal logical groupings—organizational units (OUs). In this sense, you can think of each Active Directory domain as its own tree of objects organized into containers such as OUs. If you envision Active Directory as a file system, you can think of containers such as OUs as folders within the file system. Objects that aren’t containers can be considered files that can be moved around into different folders depending on where you want them. How does this play out in real life? Depending on your organization, you may decide to create an OU for each major department in the organization, such as IT, HR, Sales, Engineering, and Finance. Each of these OUs can then contain all the users, workstations, and even security and distribution groups associated with that department. You can even create sub-OUs—for example, you can have separate containers for user accounts and for computer accounts.

The key factor here is that you decide. Many best practices around Active Directory are published on the Microsoft TechCenter Web site. No one can tell you that it has to be done a certain way. No one will know your organization better than you, so you need to take that into consideration when designing an OU structure. At the end of the day, many factors come into play when planning an OU structure: Some of them might have to do with your political boundaries, while others may be directly related to the group policies you would like to implement. For example, you could tie a restrictive set of policies for the Sales OU so that sales staff can perform only certain actions on their workstations, while providing a more lax policy on the IT OU so that the IT staff can perform necessary administrative functions without being locked down. Also, just because OUs exist doesn’t mean groups don’t exist anymore. Windows groups are still the primary way to group user and computer accounts. You will need to make a conscious decision whether or not grouping should be performed through the creation of OUs or through some form of Windows group. In some cases, using groups won’t be optional. For example, if you are grouping users or computers for the purpose of assigning access to resources such as a file share, you can accomplish that only via security groups.
OUs are chosen for two main reasons: to deploy group policies and to delegate security. Figure 4-4 shows how a domain can be split up into OUs.

**NOTE** If you haven’t noticed already, I’ve been using triangles to denote Windows Active Directory domains. This is standard Microsoft convention since the triangle symbolizes the hierarchical nature of Active Directory.

**Flexible Single Master Operations Roles**

Each server in an Active Directory domain can be either a *domain controller* or a *member server*. A domain controller hosts the Active Directory domain service and stores a physical copy of the Active Directory store, and it is also responsible for authenticating users and computers. Member servers simply participate in the domain and can perform any number of server roles. When you install a brand new instance of Windows Server 2008 (and even previous versions, all the way back to Windows 2000), it is installed first as a member server. To act as a domain controller, it can be promoted to a domain controller server. A domain controller can also be demoted back to a regular member server.
Other critical Active Directory roles are called *Flexible Single Master Operations* (FSMO) roles. Every domain controller that participates in an Active Directory domain can be written to (though changed in Windows Server 2008, this is generally the case). That means that if you have two domain controllers in the same domain, you can update the password of a user on either of the two domain controllers and the changes will be replicated to the other domain controller in the near future (or in the case of intrasite replication, almost immediately). This is called a *multi-master configuration* since multiple masters are authoritative at any given time. Certain roles, however, pertain to Active Directory and can be sensibly fulfilled only by a single server. These single master roles are referred to as FSMO roles and are listed in Table 4-1. You can and should diversify which servers hold each of these roles. If you have more than one domain controller at your disposal, it is best and sometimes required to split up these roles.

### Active Directory Sites

When planning for an Active Directory implementation, you will often be spending much of your time planning how many domains you will need, how they will be structured, how many domain controllers are required for each domain and where they will be located, and how your Active Directory will be organized internally using OUs. Much of

<table>
<thead>
<tr>
<th>FSMO Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema Master</td>
<td>Stores and manages changes to the Active Directory schema. The first domain controller in the domain is designated as the Schema Master by default.</td>
</tr>
<tr>
<td>Domain Naming Master</td>
<td>Manages domains that are created, added, or removed to and from the entire forest.</td>
</tr>
<tr>
<td>Relative ID (RID) Master</td>
<td>Ensures that all security principles issued by Active Directory (such as Security Identifiers [SIDs]) are unique.</td>
</tr>
<tr>
<td>Primary Domain Controller (PDC) Emulator</td>
<td>Synchronizes time within the domain, controls account lockout states, and manages password synchronization. When Group Policy objects (GPOs) are edited, it is performed on the server hosting the PDC Emulator role by default.</td>
</tr>
<tr>
<td>Infrastructure Master</td>
<td>Manages group membership and ensures that references to objects in this domain are updated for objects in other domains.</td>
</tr>
</tbody>
</table>

Table 4-1. Active Directory FSMO Roles
what determines the answers to these questions will be your administrative boundaries and sometimes your geographic locations. If your organization is geographically diverse and connected with expensive WAN links, it’s tempting to create a different domain for each geographic region. Although this may be the right thing to do in certain scenarios, you can also create multiple sites within an Active Directory domain if you simply need to manage replication traffic.

An Active Directory site effectively defines a collection of subnets that are connected by high-speed links. This is critical in managing replication traffic since Active Directory will try to minimize latency for *intra-site* (within a site) replication traffic while trying to minimize bandwidth utilization for *inter-site* (between sites) replication traffic. In the real world, that means that Active Directory will try to synchronize almost immediately for every change you make when computers are the same site, whereas if they are located in different sites, you can define replication parameters to control replication traffic based on known bandwidth utilization. In your organization, if users and computers are centrally managed, but satellite locations are connected to your main datacenter via expensive WAN links, instead of creating multiple domains to localize traffic, you can simply define one domain and then split it up into multiple sites, with each Active Directory site corresponding to your geographic sites.

Each site must have at least one domain controller assigned to it, and these domain controllers for a site are in charge of servicing clients that are local to the site as well as managing replication to other sites. In addition, at least one domain controller for each site should be configured as a GC server so that searches across the domain can be accomplished fairly quickly while still minimizing the amount of traffic going through the WAN link. Each site will have a domain controller that is automatically assigned the role of a *bridgehead* server—the domain controller that acts as the preferred replication partner with other sites. With this method, changes that are made within a site can be collected by the bridgehead server and sent once over the WAN link rather than having every domain controller talk to every other. Likewise, if data needs to be replicated to a site, the bridgehead server collects this new information and disperses it to the other domain controller in the site. I use the term *preferred replication partner* because replication between two sites is by no means limited to the communication between the two bridgehead servers. You can set up links between the other domain controllers in each site and other sites to act as secondary links. Figure 4-5 shows a simple scenario of a domain with two Active Directory sites: one in New York and another in Los Angeles.

**Application Data Partitions**

Since Active Directory automatically possesses the ability to replicate data across your enterprise, it seems logical that Microsoft would allow applications to take advantage of this feature. You can create application data partitions in your Active Directory to store data for your applications and have it synchronize automatically everywhere your Active Directory can reach—even halfway across the world, if you’re directory goes that far. This allows you to develop or use applications that store data in Active Directory and leverage its inherent replication abilities.
Active Directory and DNS

Considering Active Directory’s many moving parts and flexibility, your ability to locate resources across your network environment is a critical component to making it all work. With the proliferation of TCP/IP as the dominant network protocol in most organizations, DNS is a natural choice as the name resolution method on which Active Directory relies. Windows Server comes with its own DNS service with features that make it more Active Directory-friendly. For starters, it has the support for the required SRV record type, as described in RFC 2052. This DNS entry type is used to help locate a service. It also supports secure dynamic updates so that your computers can automatically register their IP addresses with the DNS Server without administrator intervention, as would be the case with traditional DNS systems. Active Directory uses DNS to locate domain controllers as well as specific FSMO roles, such as the global catalog server and the PDC emulator.

Domain and Forest Functional Levels

When Active Directory was first launched with Windows 2000, you could run it in either native mode or mixed mode. In mixed mode, Windows 2000 can interact with NT 4.0 domain controllers by pretending to be an NT 4.0 domain controller. This makes it easier to migrate to Windows 2000 from an NT 4.0 domain model. Windows 2003 introduced the concept of domain functional levels, which provide capabilities similar to those available with a mixed mode that allows a newer version of Active Directory to coexist with a previous version. (It is no longer called mixed mode because that name doesn’t indicate what it can mix with.)

have Windows 2000 domain controllers participating in your domain. Essentially, the functional level dictates the lowest common denominator supported by Active Directory. To get all the features available in the Windows Server 2008 functional level, you will need to make sure all your domain controllers are running Windows Server 2008.

Forests support four functional levels: Windows 2000, Windows Server 2003 interim, Windows Server 2003, and Windows Server 2008. Similar to domain functional levels, forest functional levels restrict features supported by the forest to those supported by that specific functional level. To get the most features out of your current setup, you will need to raise the forest functional level to the latest version.

**WINDOWS SERVER 2008 ACTIVE DIRECTORY DOMAIN SERVICES**

If you were patient enough to read through the entire “Active Directory Primer” section, congratulations! Entire books or even volumes of books could be written about Active Directory, but the purpose of the primer in this chapter is to make sure we’re all speaking the same language and understanding the same basic concepts. Now you’ve reached the juicy parts: The remainder of this chapter will focus on the new features of Active Directory in Windows Server 2008 as well as several migration scenarios.

**Active Directory Requirements**

Installing Active Directory, though quite painless, requires a bit of careful planning. Assuming you have planned how you are going to configure your Active Directory forest and domain, the following prerequisites should be in place before installing a Windows Server 2008 Active Directory:

- Your server must be running the Windows Server 2008 operating system.
- TCP/IP and DNS Server addresses should be configured. (If this is the first domain controller and DNS Server, the installation process will install the DNS service automatically and update the primary DNS entry for you.)
- If you are adding this server to an existing Windows 2000 or Windows Server 2003 forest, you must first update the schema on the schema operations master by running `adprep /forestprep`.
- If you are adding this server to an existing Windows 2000 or Windows Server 2003 domain, you must also update the infrastructure master by running `adprep /domainprep /gpprep`.
- If you are installing a read-only domain controller (RODC), you need to prepare the forest by running `adprep /rodcprep` (more on RODCs in the next section and later in the chapter).
- A working DNS infrastructure must be in place. If you don’t already have one, you can install the DNS service as part of the installation.
The New Active Directory Domain Services Installation Wizard

Since Windows Server 2008 introduces some new functionality to Active Directory, the Active Directory Domain Services Installation Wizard—otherwise known as dcpromo.exe—has also undergone some changes. DNS installation and configuration in Windows Server 2008 is automatic, if needed, unlike previous Windows versions in which it was optional. DNS also creates a new delegation or updates an existing delegation for the server automatically if it has to be installed. In addition to this, you can specify the site to which this new domain controller belongs or have it automatically determine the site to which it belongs based on its IP address. You can also configure a domain controller as an RODC. This option applies only to domain controllers other than the first one in the domain—which obviously makes sense, since you need to have at least one writable database before having a read-only copy. An RODC stores a read-only copy of the Active Directory database similar to the way backup domain controllers did in the old NT domain model. This new domain controller role is covered later in this chapter in the section called “Read-Only Domain Controller”.

The Active Directory Domain Services Installation Wizard can still be initiated by running dcpromo, but it can now also be accessed by using the Add Roles Wizard from either the Initial Configuration Tasks screen or Server Manager. You can also switch to advanced installation mode from the wizard’s interface rather than having to run dcpromo /adv from the command prompt. Microsoft has also moved the ability to create a new domain tree to the advanced mode screen. Since a Server Core installation runs without a GUI but has to support the installation of Active Directory, the unattended options for dcpromo have to support a completely silent installation. To address that, dcpromo can run now without any user interface prompts—not even to ask for a reboot. This makes the installation truly silent and unattended.

Installation Options for Active Directory Domain Services

When installing Active Directory Domain Services (AD DS) on Windows Server 2008, you can choose from a number of options for how it should be installed based on the role the new domain controller will play on your network:

- New Windows Server 2008 domain in a new Windows Server 2008 forest
- New Windows Server 2008 domain in an existing Windows 2000/2003 forest
- New Windows Server 2008 domain controller in an existing Windows 2000/2003 domain
- New Windows Server 2008 domain controller in an existing Windows 2000/2003/2008 domain from restored backup media

Ultimately, what drives these decisions will be based on your current network setup, what it is you’re trying to accomplish, and your migration plan if you are moving from a Windows 2000 or 2003 domain to a Windows Server 2008 domain. As a result, you must plan for the installation in advance of actually installing AD DS. First, you’ll read about the semantics for performing any of the preceding installation options. Then you’ll learn about planning your own migration strategy to Windows Server 2008.
New Windows Server 2008 Domain in a New Windows Server 2008 Forest

This is about the cleanest install you will ever get. It will typically be performed only if Active Directory never existed on the network or if you are creating a whole new Active Directory infrastructure (such as in a test lab environment). When performing this type of install, you must heavily consider whether Windows 2000 Server or Windows Server 2003 domain controllers will exist in this domain. That will drive the forest and domain functional level of your Active Directory. Windows Server 2008 also drops support of all Windows NT Server 4.0 domain controllers. The PDC Emulator domain controller role will still exist, except Microsoft will not support it talking to domain controllers running the legacy Windows NT 4.0 server operating system. Since this is the first domain controller in a new domain and new forest, it cannot be set up as an RODC.


In this exercise, we will install and configure Active Directory Domain Services on Windows Server 2008 as a new Windows Server 2008 domain in a new Windows Server 2008 forest.

1. Open Server Manager and click Add Roles to start the Add Roles Wizard.
2. On the Before You Begin screen, complete all the preliminary tasks. The most critical one is to set a static IP address, since domain controllers should not be configured to use DHCP. Click Next after you have completed these tasks.
3. On the Select Server Roles page, select Active Directory Domain Services, as shown in Figure 4-6. Then click Next.
4. You will see additional information about Active Directory Domain Services (Figure 4-7). Make sure you read and understand this information. Click Next when you’re ready to move on.
5. A summary page will display, showing your installation options. As no additional installation options are available at this time, click Install. This will begin to install the AD DS components but will not promote it to a domain controller yet.
6. Select the Active Directory Domain Services node in Server Manager. On the Summary page, click the Run the Active Directory Domain Services Installation Wizard link.
7. At the Welcome screen, click Next.
8. On the Choose a Deployment Configuration screen, select Create A New Domain in a New Forest, as shown in Figure 4-8; then click Next.
9. Enter the full DNS name for the new domain in the Name the Forest Root Domain screen, as shown in Figure 4-9. This will become the root domain for this forest. In this exercise, I am using WIN2K8TEST.LOCAL as my domain name. Click Next to continue.
Chapter 4: Active Directory Domain Services

Figure 4-6. Selecting Active Directory Domain Services in the Add Roles Wizard

Figure 4-7. Introduction to Active Directory Domain Services screen
Figure 4-8. Selecting to create a new forest

Figure 4-9. Name the Forest Root Domain screen
10. On the Set Forest Functional Level screen, select the Forest Functional Level based on the operating systems of the domain controllers you expect to participate in this forest. Since we don’t expect any non–Windows Server 2008 domain controllers to participate in this forest, select Windows Server 2008 as the forest functional level (Figure 4-10). Click Next.

11. The Active Directory Domain Services Installation Wizard will automatically detect that your DNS hasn’t been configured and will automatically check the DNS Server option in the Additional Domain Controller Options screen. The Global Catalog option is also selected by default since it is required for the first domain controller in a domain. You cannot select the Read-only Domain Controller option since the first domain controller cannot be made read-only (see Figure 4-11). Click Next to continue. If you see a warning that a delegation for this DNS Server will not be created, click Yes to continue.

12. In the Location for Database, Log Files, and SYSVOL screen, specify the location of the Active Directory database, log files, and SYSVOL folder. By default, it points to %WINDIR%\NTDS for the database and log files and %WINDIR%\SYSVOL for the SYSVOL. You can either enter new paths here or click Browse to select the folders. Since this is only an exercise, leave the default settings as shown in Figure 4-12. Click Next to continue.

![Figure 4-10. Selecting a forest functional level](image-url)
Figure 4-11. Specifying additional options for Active Directory Domain Services installation

Figure 4-12. Specifying the location of the Active Directory database, log files, and SYSVOL folder
TIP For efficiency, Microsoft recommends that you select a volume that does not contain application or non-directory files since Windows Server Backup backs up the directory service by volume rather than by folder.

13. In the Directory Services Restore Mode Administrator Password screen shown in Figure 4-13, enter the password you want to use as the restore mode password. This password is different from the password used for the domain administrator account. You should make sure you store this password in a safe place in case you forget it, since you will need it later if you have to recover your directory. Click Next to continue.

14. You are presented with a summary of selections made (Figure 4-14). Review this information to make sure the data is correct, and then click Next to begin the installation.

Figure 4-13. Entering the restore mode password
114 Microsoft Windows Server 2008 Administration

After several minutes, the installation is complete and you’re prompted to restart the server to complete the installation.

New Windows Server 2008 Domain in an Existing Windows 2000/2003 Forest

This is almost as clean and straightforward as creating a new Windows Server 2008 domain in a new Windows Server 2008 forest, except you must take into account the limitation of having a Windows 2000 Server or Windows Server 2003 forest. This is important since you are practically limiting your forest functionality to any of the Windows 2000 Server– and Windows Server 2003–compatible functional levels. Your most critical step before successfully installing a new Windows Server 2008 domain in an existing Windows 2000/2003 forest is to extend the schema to support Windows Server 2008 by running the following command on the schema master:

```
Adprep /forestprep
```
Like Windows Server 2003, Windows Server 2008 requires that the primary domain controller operations master run on Windows Server 2008 before any Windows Server 2008 security principles are created.

**Hands-On Exercise: Installing a New Windows Server 2008 Domain in an Existing Windows Server 2003 Forest**  In this exercise you will need an existing Windows Server 2003 forest. I set up a new Windows Server 2003 domain and forest called LABTEST.LOCAL, into which I will add my new Windows Server 2008 domain. You must have the username and password of an account on the existing Windows 2003 forest that is a member of Enterprise Admins, Schema Admins, and Domain Admins Windows global security groups.

1. Log on to the server in the existing Windows 2003 forest that currently acts as the schema master. You must log in with an account that is a member of Enterprise Admins, Schema Admins, and Domain Admins.

2. Copy the \Sources\adprep folder from the Windows Server 2008 media to the local drive on the schema master—for example, C:\adprep.

3. Open a command prompt, change to the directory where you copied the ADPREP folder, and run the following:

   Adprep /forestprep

4. ADPREP will display a warning that all your Windows 2000 Active Directory domain controllers must be upgraded to specific service-pack and patch levels to prevent Active Directory corruption, as shown in Figure 4-15. If you know your servers are compliant, press c, and then press enter to continue.

---

**Figure 4-15.** Running adprep /forestprep on Windows Server 2003
5. Allow the process to complete. If multiple domain controllers are in the forest, make sure replication has completed before continuing.
7. In Server Manager, click Add Roles to start the Add Roles Wizard.
8. Review the preliminary tasks and click Next to continue.
9. On the Select Server Roles page, select Active Directory Domain Services, and then click Next.
10. Read the Introduction to Active Directory Domain Services and click Next.
11. Verify the installation options and then click Install.
12. Click Close when the installation completes.
13. In Server Manager, expand the Roles node and select Active Directory Domain Services.
14. Click the Run the Active Directory Domain Services Installation Wizard link.
15. On the Active Directory Domain Services Installation Wizard Welcome screen, click Next.
16. On the Choose a Deployment Configuration screen, select Existing Forest and Create a New Domain in an Existing Forest, as shown in Figure 4-16. Then click Next.

Figure 4-16. Deployment configuration selection to add to an existing forest
17. Enter the name of any domain in the forest to which you want this domain to join. Click the Set button to specify credentials that have privileges to add a domain in the parent domain (Figure 4-17). Then click Next. In my test environment, I add this new Windows Server 2008 domain to my existing LABTEST.LOCAL Windows Server 2003 forest using the Administrator account.

18. Enter the full DNS name of the parent domain, the name of this new domain (just the domain name, not the FQDN), and verify that the complete DNS name for the child domain is displayed in the appropriate field, as shown in Figure 4-18. Then click Next.

19. Select the appropriate domain functional level, and then click Next. You aren’t restricted to using the same functional level as the forest. For example, you can use a Windows Server 2008 domain functional level even if your parent forest is running in a Windows Server 2003 forest functional level. Your selection here is based on what operating system is being used by the domain controllers in this new child domain. In this exercise, I want this domain to have only Windows Server 2008 domain controllers, so I select Windows Server 2008 as the domain functional level.

Figure 4-17. Providing network credentials for the parent domain
20. Choose the Active Directory site to which this domain controller will belong. Optionally, you can check the Use the Site that Corresponds to the IP Address of this Computer checkbox to have the Active Directory Domain Services Installation Wizard automatically configure it. Since we have only one site, select Default-First-Site-Name and click Next (Figure 4-19).

21. In the Additional Options screen, the DNS Server option is automatically selected, since no DNS Server is configured as authoritative for this child domain. Optionally, you can select this domain controller to also act as a global catalog server. Check this checkbox in this exercise since you want this domain controller to have a global catalog. Click Next to continue.

22. Specify the location of the Active Directory database, log files, and SYSVOL folder in the Location for Database, Log Files, and SYSVOL screen. By default, it points to %WINDIR%\NTDS for the database and log files and %WINDIR%\SYSVOL for the SYSVOL. You can either enter a new path here or click Browse to select a folder. For now, leave the default settings and click Next to continue.
23. Enter the Restore Mode Password, which can be different from the domain Administrator account password.

24. Verify the summary of your selection, and then click Next to begin the installation.

25. Once the installation completes, restart the server when prompted.

**New Windows Server 2008 Domain Controller in an Existing Windows 2000/2003 Domain**

This might be one of those typical migration scenarios in which you are slowly upgrading to Windows Server 2008 Active Directory without “going all out.” If your old domain controllers need to be refreshed, installing new domain controllers into your existing Windows 2000/2003 domain with Windows Server 2008 installed eases you in as you decommission older hardware running the older operating system. Once you’ve completely replaced all your domain controllers for Windows Server 2008, you can then raise its functional level to Windows Server 2008 and take advantage of the new functionality.
Similar to adding a new Windows Server 2008 domain to an existing Windows 2000/2003 forest, if this is the first Windows Server 2008 domain controller to be added to your existing domain, you will need to extend the schema by running `adprep /forestprep` on the schema master. If this is the first Windows Server 2008 domain controller in a Windows 2000 Server domain, you must also prepare the domain by running the following command on the infrastructure master:

```
adprep /domainprep /gpprep
```

If this is the first Windows Server 2008 domain controller in an existing Windows Server 2003 domain, you will need to run a similar command as above on the infrastructure master minus the `/gpprep` switch:

```
adprep /domainprep
```

**NOTE**  Technically, you could run the same command on a Windows Server 2003 domain and a Windows 2000 Server domain (with the `/gpprep` switch), except it will display an error message on a Windows Server 2003 domain that you can safely ignore.

Surprisingly enough, you can install a Windows Server 2008 domain controller as an RODC when it’s added to an existing Windows 2000/2003 domain, but only if the domain is running in the Windows Server 2003 forest and domain functional levels, and the PDC FSMO role is on a Windows Server 2008 server. The caveat is that if this is the first RODC in the forest, you must also run the following command to prepare the forest:

```
adprep /rodcprep
```

**Hands-On Exercise: Installing a New Windows Server 2008 Domain Controller in an Existing Windows Server 2003 Domain**  In this exercise we will install a new Windows Server 2008 domain controller in an existing Windows Server 2003 domain. We will work under the assumption that this is the first Windows Server 2008 domain controller to be added to the domain and the forest, so we will need to extend the schema and prepare the domain using `adprep.exe`.

1. Log on to the server in the existing Windows 2003 forest that currently acts as the schema master. You must log in with an account that is a member of Enterprise Admins, Schema Admins, and Domain Admins.
2. Copy the `\Sources\adprep` folder from the Windows Server 2008 media to the local drive on the schema master—for example, `C:\adprep`.
3. Open a command prompt, change the directory to the directory in which you copied the ADPREP folder, and run this command:

```
adprep /forestprep
```
4. ADPREP will warn you that all Windows 2000 Active Directory domain controllers must be upgraded to a specific service pack and patch levels to prevent Active Directory corruption. If you know your servers are compliant, press c, and then press ENTER to continue.

5. Log on to the server on the existing Windows Server 2003 domain that is acting as the infrastructure master with an account that is a member of the Domain Admins group.

6. If this is a different server from that used in step 1, copy the \Sources\adprep folder from the Windows Server 2008 media to this server.

7. Open a command prompt, change to the directory in which you copied the ADPREP folder, and run the following:

   Adprep /domainprep

8. Wait for this process to complete successfully and replicate the changes to the rest of the domain before proceeding.


10. In Server Manager, click Add Roles to launch the Add Roles Wizard.

11. Review the preliminary tasks and click Next to continue.


13. Read the Introduction to Active Directory Domain Services, and then click Next.

14. Verify the installation options, and then click Install.

15. Click Close when the installation completes.

16. Expand the Roles node in Server Manager and select Active Directory Domain Services.

17. Click the Run the Active Directory Domain Services Installation Wizard link.

18. Click Next on the Active Directory Domain Services Installation Wizard Welcome screen.

19. On the Deployment Configuration screen, select Existing Forest and Add a Domain Controller to an Existing Domain; then click Next.

20. Enter the name of any domain in the forest to which you want this domain to join. Click the Set button to specify credentials that have privileges to add a domain in the parent domain. Then click Next.

21. On the Select Domain screen, select the domain to which this domain controller will belong. You will see a warning that you will not be able to install a read-only DC because adprep /rodcprep has not yet run. Click Yes to continue.

22. On the Select Site screen, select the site to which you want this domain controller to belong; then click Next to continue.
23. On the Additional Options screen, you can select to install a DNS Server and Global Catalog, which are both selected by default. You will not be able to select the Read-only Domain Controller option unless the server running the PDC Emulator role is running Windows Server 2003. Click Next to continue.

24. Specify the location of the Active Directory database, log files, and SYSVOL folder in the Location for Database, Log Files, and SYSVOL screen. By default, it points to %WINDIR%\NTDS for the database and log files and %WINDIR%\SYSVOL for the SYSVOL. You can either enter a new path here or click Browse to select the folder. I leave the default settings and click Next to continue.

25. Verify the summary of your selection, and then click Next to begin the installation.

26. Once the installation completes, restart the server when prompted.


Windows 2003 introduced the ability to restore an Active Directory from backup media to reduce the replication traffic required to set up a new domain controller. This functionality continues with Windows Server 2008. You can use this method to install a new Windows Server 2008 domain controller only in an existing domain, and the domain must be prepared from the same server type as the new domain controller. To be specific, the following options must match:

- Domain controller option (Writable versus Read-Only)
- Operating system including Service Pack level
- Platform (x86, x64, or IA64)

If you are installing from restored backup media on a full installation of Windows Server 2008, the source of the media can be from a Server Core installation provided that the same server type conditions listed above match. You can use backup media from a read-only domain controller but only to another RODC.

**Hands-On Exercise: Installing Active Directory from a Restored Backup**

This exercise is quite lengthy because a number of tasks need to be performed to make this work. First, on the source domain controller, create a backup of the volume containing the Active Directory database (ntds.dit). To simplify things, back up the source domain controller to a share on the destination server that you want to turn into your new domain controller. Then perform a restore of this backup to extract the NTDS folder. The Windows Server Backup user interface doesn’t allow you simply to restore the system state; instead, use the wbadmin command to perform this specialized restore. Finally, run the Active Directory
Chapter 4: Active Directory Domain Services

Domain Services Installation Wizard in advanced mode to perform the installation of Active Directory Domain Services on your destination server using the backed up and restored system state. For simplicity sake, I call the source domain controller SERVER1 and the new server that will become a domain controller SERVER2.

**NOTE** You need to add the Windows Server Backup feature on both servers prior to continuing with this exercise. SERVER2 must already be a member server of the domain, and you may need to allow File Sharing to go through the Windows Firewall or disable the firewall completely so that the backups from SERVER1 can be copied over to SERVER2.

1. Log on to SERVER1.
2. Run Server Manager, expand the Storage node, and select Windows Server Backup.
3. Right-click Windows Server Backup and select Backup Once.
4. Click Next on the Backup Options screen.
5. Select Custom from the Select Items menu and click Next.
6. Select the volume (drive letter) that contains the ntds.dit file and click Next.
7. On the Specify Location Type screen, select Remote Shared Folder and click Next.
8. Type in the UNC path to the share on SERVER2 where you want to store the backup (that is, `\SERVER2\e$\backup`). Select Inherit from the Access Control options and click Next.
9. Review the summary page, and then click Backup.
10. Wait for the backup to complete.
11. Log onto SERVER2.
12. Open a command prompt.
13. Run the following:

   ```
   wbadmin get versions -BackupTarget:\\SERVER2\e$\backup
   ```

14. Take note of the Version Identifier; it will be a date and time in the format MM/DD/YYYY-HH:MM—for example, 12/10/2007-01:30.
15. Create a folder to which you want to restore—for example, E:\restore.
16. Restore the system state data by running the following:

   ```
   wbadmin start recovery -backupTarget: \\SERVER2\e$\backup
   -version:12/10/2007-01:30 -items:ADExtended -itemtype:app 
   -recoveryTarget:"E:\restore"
   ```
17. Press y when asked if you want to restore the application Active Directory Domain Services.
18. Wait for the restore to complete.
19. In Server Manager, click Add Roles to launch the Add Roles Wizard.
20. Review the preliminary tasks and click Next to continue.
22. Read the Introduction to Active Directory Domain Services and then click Next.
23. Verify the installation options; then click Install.
24. Click Close when the installation completes.
25. Expand the Roles node in Server Manager and select Active Directory Domain Services.
26. Click the Run the Active Directory Domain Services Installation Wizard link.
27. On the Active Directory Domain Services Installation Wizard Welcome screen (Figure 4-20), check the Use Advanced Mode Installation checkbox. Then click Next.

![Figure 4-20. Selecting the Advanced Mode installation](Image)
28. In the Deployment Configuration screen, select Existing Forest and Add a Domain Controller to an Existing Domain; then click Next.

29. Enter the name of any domain in the forest to which you want this domain to join. Click the Set button to specify credentials for privileges to add a domain in the parent domain, and then click Next.

30. On the Select Domain screen, select the domain to which this domain controller will belong. Then click Next.

31. On the Select Site screen, select the site to which you want this domain controller to belong. Then click Next to continue.

32. On the Additional Options screen, you can select to install a DNS Server and Global Catalog, which are both selected by default. You will not be able to select the Read-only Domain Controller option unless the server running the PDC Emulator role is running Windows Server 2003. Click Next to continue.

33. From the Install from Media screen, select Replicate Data from media at the Following Location and specify the path to the folder where you restored the backup from SERVER1, as shown in Figure 4-21. Then click Next.

Figure 4-21. Specifying the Install from Media source folder
34. Select Any Writable Domain Controller from the Source Domain Controller list, and then click Next. If you want to select a specific domain controller, you can select that instead.

35. Leave the defaults in the Location for Database, Log Files, and SYSVOL screen and then click Next.

36. Enter the Restore Mode Password and click Next.

37. Review the Summary and then click Next. This will install Active Directory Domain Services and initialize it with the restored data. It will then synchronize with a writable domain controller to get the latest updates.

38. Restart the computer when prompted.

Verifying Active Directory Installation

After you install AD DS and restart the computer, you should make sure that everything is working the way it should. The first place you should check is the Directory Service event log. If you see any warning or error messages, read through them and resolve any problems. Next you should verify that you can access the SYSVOL share. Go to any client in your domain and try to access `\WIN2K8DC\SYSVOL`, where `WIN2K8DC` is the name of your domain controller. If you have more than one domain controller, SYSVOL is replicated via the File Replication Service, so make sure no errors occur in the File Replication event log. Launch the Active Directory Users and Computers MMC snap-in, and make sure you can view all the objects in your domain.

If you installed DNS as part of your AD DS installation, make sure it’s working as well. You should verify that no errors appear in the DNS Server event log. Open the DNS Server MMC snap-in on your primary DNS Server and make sure it includes a zone called `_msdcs.DOMAIN`, where `DOMAIN` is your fully qualified domain name—such as `_msdcs.LABTEST.LOCAL`. This zone holds all the relevant SRV records for your domain. It will have entries for your domain controllers, domains, global catalog servers, and PDC emulators, as shown in Figure 4-22.

Removing Active Directory Domain Services

What goes up must come down, and a clean removal strategy makes it easy. Removing the AD DS role using the Active Directory Domain Services Installation Wizard can be accomplished using the full user interface or completely unattended. You can also initiate the removal of AD DS if you choose to uninstall it using the Remove Roles Wizard in Server Manager. You can use three different scenarios to remove the AD DS role from a server. It can be one of many domain controllers in a domain that you want to demote back to member server status. It can be the last domain controller to be removed from a domain or even the last domain controller to be removed from a forest.
To remove the Active Directory Domain Services role using a GUI, you will first need to demote the domain controller to a regular member server using the Active Directory Domain Services Installation Wizard (dcpromo). Once it has been demoted, you can use the Remove Roles Wizard and select Active Directory Domain Services for removal. You will then need to follow the on-screen directions and restart; upon restart, your server will no longer be a domain controller. The only caveat with removing the last domain controller for the domain or the forest is that you will be asked to perform a series of security tasks that remove the cryptographic keys and then decrypt the Encrypted File System (EFS) before proceeding. This is necessary only if you want to keep any of the data that has been encrypted using these methods.

Hands-On Exercise: Removing Active Directory Domain Service from the Last Domain Controller in a Domain and a Forest

In this exercise, we will remove AD DS from the last domain controller in a domain and a forest. When this process completes, the Active Directory forest you are removing will cease to exist. If you follow along, make sure you do this exercise in a test lab first, since the only way back would be a complete restore of Active Directory.

1. Launch the Active Directory Domain Services Installation Wizard by running dcpromo from the command prompt.
2. Click Next at the Welcome screen. Click OK if you are prompted about the server being a global catalog server.
3. Since this is the last domain controller in the domain, check the Delete The Domain Because this Server Is the Last Domain Controller in the Domain checkbox and click Next, as shown in Figure 4-23.

**NOTE** Prior to deleting the domain, it is a good idea to export all cryptographic keys and decrypt any EFS-encrypted files or e-mails, because once this process completes, you will be unable to access them.

4. The application partitions that are available in your Active Directory database are shown and will be marked for deletion. By default, if you have an Active Directory integrated DNS Server, the DNS directory will be displayed here as an application partition. Click Next to delete these partitions (Figure 4-24).

5. Confirm that you want to delete all the application partitions by checking the checkbox on the Confirm Deletion screen, and then click Next (Figure 4-25).

6. Enter the password for the domain’s administrator account and click Next.

7. Review the selections you’ve made and click Next to begin the removal process.

8. Restart the server when prompted after the process has completed.

---

Figure 4-23. Deleting the domain
Figure 4-24. Deleting application directory partitions

Figure 4-25. Confirming the deletion of application directory partitions
9. Run the Remove Roles Wizard by clicking the Remove Roles link in Server Manager or by using the Initial Configuration Wizard.

10. Review the preliminary tasks and click Next to continue.

11. Uncheck the Active Directory Domain Services checkbox and click Next (Figure 4-26).

12. Confirm the Removal Selections, and then click Remove.

**Unattended Installation**

The graphical installation of AD DS using the Active Directory Domain Services Installation Wizard makes it easy to install this role onto a Windows Server 2008 installation. In some cases, you will want to install AD DS using an unattended installation. For example, as you saw in Chapter 2, the unattended installation method is the only method you can
use to install AD DS on a Server Core. You might also want to use this method of installation if you are scripting an install of multiple domain controllers to make sure they are set up consistently.

To install and configured AD DS using an unattended installation, you will need to create an answer file for dcpromo.exe (which is really just the Active Directory Domain Services Installation Wizard). Optionally, if you don’t want to use an answer file, you can pass all these parameters at the command line. The first step to making this successful is to understand what parameters you will need to specify, since this choice depends on the type of options you will be selecting for installation. For example, setting up an answer file for the first domain controller of a new domain and a new forest has different parameter requirements than adding a new domain to an existing forest.

**NOTE** The unattended option replaces the steps in the AD DS installation only as it relates to selections in the Active Directory Domain Services Installation Wizard. If you are adding a new domain to an existing Windows 2000/2003 forest or a new domain controller to an existing Windows 2000/2003 domain, you will still need to perform the schema and other domain updates using adprep as described in the previous hands-on exercises.

**Hands-On Exercise: Unattended Installation of a New Domain Controller to an Existing Windows Server 2008 Domain**

In this exercise, we perform an unattended installation of a new domain controller to an existing Windows Server 2008 domain. It is assumed that the server is already a member server of the domain in which it will become a domain controller.

1. Open the Notepad application.
2. Type `[DCINSTALL]` and press **enter**.
3. Enter the following lines (each on its own line, as shown here):
   
   ```
   UserName=Administrator
   UserDomain=LABTEST.LOCAL
   Password=P@ssword
   ReplicaOrNewDomain=replica
   ReplicaDomainDNSName=LABTEST.LOCAL
   DNSOnNetwork=yes
   SafeModeAdminPassword=P@ssword
   RebootOnCompletion=yes
   ```

   **NOTE** You will need to replace the username, passwords, and domain names with whatever is appropriate for your network configuration.

4. Save the file to C:\Unattend.txt.
5. Open a command prompt and run `dcpromo /unattend:c:\unattend.txt`. 
Restartable Active Directory Domain Services

Anyone who’s ever had to support Active Directory Domain Controllers knows how cumbersome it is to perform maintenance on the Active Directory database. For example, if you want to perform an offline defrag of the database, you have to restart the computer and boot into Directory Services Restore mode. Not only does that increase the amount of downtime for the domain controller, but other services that are not related to Active Directory, such as DNS and Dynamic Host Configuration Protocol (DHCP), will also be unavailable while in Directory Services Restore Mode (DSRM) if the domain controller also performs those two roles. If the domain controller is not local to you and you don’t have some form of hardware-based, lights-out, remote-control solution, you would need to visit the server physically to perform this task as well.

Microsoft has made managing Active Directory much easier by giving administrators the option to stop and start Active Directory at will. DSRM is still available, but if you want to perform simple maintenance tasks such as offline defrag, you can simply stop the Domain Controller service on the domain controller. The server will effectively stop acting as a domain controller, and the Active Directory database will be offline and available for maintenance.

**TIP** Stopping the Domain Controller service also stops Active Directory–dependent services such as Distributed File System (DFS) replication, inter-site messaging, and Kerberos Key Distribution Center. When you restart the service, the other services aren’t automatically started, so you will need to start them manually.

The advantage to this functionality is that unrelated services, such as DHCP, remain up and running. If you run a small or medium-size network in which your domain controller also acts as a DHCP Server, you can take your time performing Active Directory maintenance while the server happily goes on issuing IP addresses. This feature is available regardless of functional level, as non–Windows Server 2008 domain controllers will simply treat this domain controller just as they would any domain controller that is in Directory Services Restore Mode.

With this newly added feature, Active Directory can now be in one of three different states at any given time: Started, Stopped, and Directory Services Restore Mode. When a domain controller is in a Started state, it functions as any regular domain controller. In a Stopped state, the Active Directory database (ntds.dit) goes offline, just as in DSRM, but it also acts like a member server in that if other domain controllers are still available, you can log on to the server with domain credentials. As a functional member server, you can perform maintenance using software-based, remote-control solutions such as Terminal Services. You should leave your server in a Stopped state only while you are performing maintenance. Replication with domain controllers and authenticating domain users cannot be performed by the server until it is returned to the Started state. In the last state, Directory Services Restore Mode, the Active Directory database is offline and the server goes into safe mode, in which other nonessential services are also not started. DSRM in Windows Server 2008 is functionally equivalent to DSRM in Windows Server 2003.
Auditing Active Directory Domain Services

Auditing allows you to track access and changes to your Active Directory. This is nothing new. Auditing has always been a part of Windows Server, but with Windows Server 2008, auditing has been enhanced. For example, you can now log changes to attributes, which means you can log old values and new values. Auditing shouldn’t be taken lightly, as many changes are made to Active Directory over the course of a day or even a few hours, and too much auditing can adversely impact performance and drastically increase your storage requirements. This can also create a lot of event log “clutter” that requires filtering to locate events in which you are truly interested. Careful planning of what events to log and how frequently to purge or save the log to offline storage can either make or break an audit policy.

Auditing is enabled by modifying the default domain controller policy. When defining your audit policy, you should specify whether to audit success or failure, or not audit at all. Remember that just because you’ve enabled auditing by modifying the default domain controller policy, you will still need to modify the system access control list (SACL) of an object you want to audit. This allows you to be very granular while defining in which audit events you are actually interested. As you can see in Figure 4-27, I have enabled success and failure auditing of directory service access for my Windows Server 2008 domain.

![Group Policy Management Editor showing default domain controller audit policy](image)

Figure 4-27. Group Policy Management Editor showing default domain controller audit policy
TIP  You must install the Group Policy Management feature if you want to manage group policies from your Windows Server 2008 server. It is no longer available by default through Active Directory Users and Computers. In addition, to access the Security tab of an object to specify what actions you would like to audit, you must check the Advanced Features option in the View menu of Active Directory Users and Computers.

Windows Server 2008 has four subcategories relating to the Audit Directory Service Access policy:

- Directory Service Access
- Directory Service Changes
- Directory Service Replication
- Detailed Directory Service Replication

All audit events are sent to the Windows Security Event Log. What’s exciting about these new subcategories is that when an object’s attribute is changed, both the old and new values of the modified attribute are logged (Table 4-2). Likewise, when a new object is created, attribute values that are set during the object’s creation are also logged. When an object is moved, the old and new locations of the object are logged; when an object is undeleted, the location where it is restored is logged as well. This detailed logging capability is useful when you want to track down the history of an object’s changes. Object deletion is logged only if you have enabled the Audit Directory Service Access policy.

**Global Audit Policy**

As shown in Figure 4-27, the Audit Directory Service Access policy can be enabled by modifying the default domain controller policy. Doing so globally enables all directory service policy subcategories.

If you are looking at the default domain policy and can’t figure out how to enable or disable policy subcategories selectively, you’re not alone. Microsoft didn’t provide an intuitive interface where you can set or unset audit subcategories. Instead, you will need

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Type of Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5136</td>
<td>Modify</td>
<td>Event logged when an object’s attribute is modified</td>
</tr>
<tr>
<td>5137</td>
<td>Create</td>
<td>Event logged when a new object is created</td>
</tr>
<tr>
<td>5138</td>
<td>Undelete</td>
<td>Event logged when an object is undeleted</td>
</tr>
<tr>
<td>5139</td>
<td>Move</td>
<td>Event logged when an object is moved</td>
</tr>
</tbody>
</table>

Table 4-2. Event IDs Associated with Audit Directory Service Access Policies
to use a command-line tool called auditpol.exe to perform these changes. For example, to see the current policy for Directory Service Changes, you can run this:

Auditpol /get /subcategory:"Directory Service Changes"

To disable failure event logs for the Directory Service Changes subcategory, you can run the following command:

Auditpol /set /subcategory:"Directory Service Changes" /failure:disable

System Access Control List

Each object contains a security descriptor that defines not only who or what can access it but also a SACL that ultimately determines whether access to this object will be audited. Setting the global option to audit directory service access or changes is only half the story. You still need to create access control entries (ACEs) explicitly in the SACLs of an object before any auditing will be performed. This is done by design to ensure that logging is enabled only for those objects in which you are actually interested.

TIP Some SACL ACEs are created by default. When enabling auditing, you may want to remove some of these ACEs if you think you’re logging too much.

Schema

This advanced method of controlling auditing allows you to exclude an attribute from being audited at the schema level. This is done by setting bit 8 in the searchFlags property of an attribute. When this is done, this attribute will not be audited for all objects that contain this attribute.

TIP The searchFlags property of an attribute also controls whether it is indexed, replicated to the GC, marked as confidential, or, in this case, not logged in the event log.

Read-Only Domain Controller

When Active Directory was introduced in Windows 2000 Server, it completely changed the way we thought about domain controller deployment. In the NT domain model, we had a single writable instance of the domain database that was stored in the PDC. To provide load balancing and a relative amount of redundancy, you could deploy additional backup domain controllers (BDCs), but these had only a read-only copy of the domain database.

Active Directory domain controllers follow a multi-master model, where all domain controllers are writable and changes can be made to any domain controller. These changes are then replicated to all the other domain controllers. Now, with Windows Server 2008, the concept of a backup domain controller has returned, this time in the form of an RODC. Active Directory domain controllers are still multi-master, but now you have the option of deploying RODCs throughout your network.
Why would you ever want to deploy RODCs? This domain controller mode is highly useful if you want to provide Active Directory authentication services in a location that is not adequately secure for a writable copy of your Active Directory database. Also, in read-only mode, the domain controller can respond to requests more quickly since it doesn’t have to worry about processing changes that need to be replicated up to other domain controllers. It is also a good option if you have an application that performs best when installed on a domain controller. By running that application on an RODC rather than a regular domain controller, you don’t run the risk that the DC will be inadvertently used by the application to make changes to your directory. You can also deploy RODCs to provide localized authentication services to locations that have slower network connections to your main datacenter and that do not have knowledgeable IT staff onsite.

Before you can install an RODC in your domain, the domain controller hosting the PDC Emulator role must reside on a server running Windows Server 2008. An RODC also has the added restriction that it cannot act as a Global Catalog server, but it does support caching of universal groups. In addition to this, the functional level of the forest must at minimum be Windows Server 2003 before an RODC can be installed.

**TIP** Universal groups are groups that are available and can be used throughout an entire forest. They can contain other groups and users and can be assigned to resources. Universal group membership is stored in the global catalog (or cached on an RODC) and affects replication.

The RODC Active Directory database stores all the same objects and attributes that any regular domain controller would store—except it doesn’t store account passwords. Read-only Active Directory queries to domain controllers using LDAP are processed normally, whereas any requests to write to the database using LDAP will be returned with a referral to a writable DC. Only downstream replication occurs on an RODC. This includes replication data related to both the Active Directory database and to DFS replication traffic. This simplifies the replication process and optimizes any work that needs to be done by the bridgehead servers in the same site.

Passwords are not stored on RODCs by design, since it is assumed that the RODC will reside in a potentially less secure environment than the rest of your domain controllers. If a user or computer attempts to authenticate to an RODC and it determines that the account exists, the password is then forwarded to a writable DC for authentication. Needless to say, this is not very efficient since it would only increase the traffic going between domain controllers. You can enable credential caching on an RODC. In this case, if an authentication request arrives, it can check whether the user’s password has already been cached on the RODC’s Active Directory database: If so, it can process the authentication on its own; otherwise, it will forward the request to a writable DC and then store the password for future authentication requests by the same account.

You can control how often this replication occurs with an RODC. You want it frequent enough so that password changes are propagated effectively while minimizing replication traffic. This default behavior of caching credentials only of accounts that are
already authenticated limits the potential exposure of your domain database. If someone were to gain access to this read-only data store, it would contain the cached passwords of those accounts that have authenticated and not passwords of every account in your domain. Since you will typically deploy RODCs at remote branch offices, this default behavior is ideal, since only a very small subset of your users would be authenticating from that site anyway.

To address maintenance concerns, Microsoft designed the RODC so that you can delegate a regular user account with administrative rights specifically on your RODC server. That designated user account can then log on to the server and perform any maintenance task necessary, such as installing Microsoft Critical Updates or defragmenting the hard drive. Users would not be able to log on to any other domain controller in your domain or perform any other tasks on the domain and are completely restricted to local changes that require administrative privileges.

An RODC can also host DNS to provide name resolution services. However, unlike other Active Directory integrated DNS zones, computers will not be able to update their DNS entry on an RODC. Instead, they will get a referral to a writable DNS Server that can take the update and then replicate this back down to the RODC. (This read-only DNS mode is new with Windows Server 2008 and is discussed thoroughly in Chapter 10 along with all the other DNS changes in Windows Server 2008.)

Backup and Recovery
The ability to back up and recover Active Directory properly is an absolutely necessary skill every Windows administrator must master. If you’ve never had to perform an Active Directory recovery in a production environment, consider yourself very lucky. Although you can mitigate the risk of having to restore Active Directory from scratch by setting up enough domain controllers and physically dispersing them to prevent single points of failure, as with all things, you should carefully plan, and more important, test your backup and restore procedures. You need to worry about only two general scenarios when it comes to your domain controllers: complete domain controller failure either due to hardware failure or software corruption, and intentional or accidental deletion or modification of objects within your directory.

If you lose a domain controller and you are fortunate enough to have more than one domain controller in your domain, you can simply set up a new domain controller, and replication will automatically commence to bring that new server up to the current state of your Active Directory. If objects in your directory are deleted or modified either intentionally or accidentally, these changes may have already been replicated to all your domain controllers before you can stop it from propagating. In this case, you can perform an authoritative restore of those objects to restore them. If you are restoring a domain controller and want to minimize the amount of replication traffic with your other domain controllers, you can optionally install a domain controller using data from a previous backup. Once this controller goes online, it can then replicate the remaining updates from other domain controllers.
The System State in Windows Server 2008 contains much more data than the System State of previous Windows versions. The System State now minimally contains the following:

- Registry
- COM+ Class Registration database
- Boot files and system files
- Certificate Services database
- Active Directory Domain Services
- SYSVOL folder
- Cluster service information
- Microsoft IIS metadirectory
- System files protected by Windows File Protection (WFP)

In addition to this, you can no longer simply back up the System State as you could using NTBACKUP. In fact, NTBACKUP has been deprecated in Windows Server 2008 and replaced by Windows Server Backup. Windows Server Backup is not installed by default on Windows Server 2008, so it must be installed by running the Add Features Wizard from Server Manager. To back up the System State, you must back up the entire volume on which the system files reside. As a result, Microsoft’s best practices guidelines recommend that operating system files (%WINDIR%), Active Directory database (ntds.dit) and log files, and SYSVOL all be stored on a volume that does not contain user data or application data.

**IMPORTANT** Before you can perform any kind of backup or restore of AD DS, you must install Windows Server Backup.

Depending on how often your Active Directory database changes, you will want to back up your Active Directory database at least once a day. You can back up the full server or only critical volumes that will be needed to restore your AD DS on either the same or a new physical server. Critical volumes contain the following:

- Operating system files (the entire %WINDIR% directory)
- Registry
- Ntds.dit database file and associated log files
- SYSVOL folders

I won’t go over how to use Window Server Backup again in this chapter. Chapter 3 discusses it fully. The important thing to remember is that you must minimally back up all the volumes containing the information listed here as part of the critical volumes.
Hands-On Exercise: Performing a Non-authoritative Restore of Active Directory

In this exercise we will perform a non-authoritative restore of Active Directory. Unlike the procedure for installing Active Directory from a restored backup, this assumes that the server on which you will be performing this is already a domain controller with AD DS installed. In this example, I assume that you have made a critical volume backup to a separate drive on your server (the E: drive).

1. Log on to the server to which you want to restore, and open a command prompt.
2. Run the following command sequence to restart the server in Directory Services Restore Mode (DSRM):
   ```
   bcdedit /set safeboot dsrepair
   shutdown -t 0 -r
   ```
3. Click Switch User at the logon screen.
4. Click Other User.
5. Enter `.\Administrator` as the username and the DSRM password; then log in.
6. Open a command prompt.
7. Enter `Diskpart` then press enter.
8. Enter `list vol` and press enter.
9. Note the drive letter assigned to the volume where you created your critical volume backup based on the disk label.
10. Enter `exit` and press enter.
11. Run the following command (replace E: with the appropriate driver letter of your backup volume):
   ```
   Wbadmin get versions -backuptarget:E:
   ```
12. Note the version identifier for your backup. It should be in the format MM/DD/YYYY-HH:MM—for example, 09/26/2007-22:30.
13. Enter the following command to initiate the restore process. Make sure you enter the version identifier that you got in the previous step and the drive letter of the backup drive from step 9:
   ```
   Wbadmin start sysstaterecovery -version:09/26/2007-22:30 -backuptarget:E:
   ```
14. Press y and then press enter to proceed.
15. After the restore has completed, run the sequence of commands shown next to reset the server in normal (non-DSRM) mode.
Hands-On Exercise: Performing an Authoritative Restore of Active Directory

With regards to restoring Active Directory, the most common of all the possible problems is that someone deletes an object either on purpose or accidentally. To recover the deleted object, you will need to perform an authoritative restore of the object, and then mark whatever objects you want restored as authoritative. In this example, we restore a deleted user account with the distinguished name of CN=TestUser,CN=Users,DC=TESTDOM2,DC=LOCAL.

1. Perform a non-authoritative restore of Active Directory, but don’t restart the server into normal mode (in other words, stop at step 14 from the preceding procedure if you followed those instructions on performing a non-authoritative restore).

   IMPORTANT  It is absolutely critical that you do not start the server in normal mode, or Active Directory Domain Services will immediately begin synchronization. If you are unsure whether you can properly boot into DSRM, unplug the network cable prior to restart to ensure that there is no way synchronization can occur if the server is accidentally booted up normally.

2. Choose Start | Run to open the command prompt, type ntdsutil, and then press ENTER.

3. Type activate instance NTDS and then press ENTER.

4. Type authoritative restore and then press ENTER.

5. Type restore object “CN=TestUser,CN=Users,DC=TESTDOM2,DC=LOCAL”, and then press ENTER.

6. Click Yes to confirm the command.

7. Take a note of where the text file and LDIF files for the operation are stored. You may need this information to restore backlinks in this domain or a different domain.

8. Type quit and then press ENTER.

9. After the restore has completed, run the following sequence of commands to reset the server in normal (non-DSRM) mode:

   Bcdedit /deletevalue safeboot
   shutdown -t 0 -r
10. Log on to the server.

11. Open a command prompt, run the following command, and make sure that no errors are returned (this assumes that DNS name of this server is SERVER3.TESTDOM2.LOCAL):

```
Repadmin /syncall SERVER3.TESTDOM2.LOCAL /e d /A /P /q
```

12. Recover any backlinks to the object you just restored using the LDIF file that was created by the authoritative restore. Run the following command (this assumes that the output of the authoritative restore created an LDIF file called c:\restored_object.ldf):

```
Ldifde -i -k -f c:\restored_object.ldf
```

13. If this is the only domain in the tree, you are done; otherwise, continue on to the next step.

14. If you are restoring an object in a forest that has more than one domain, you will need to create LDIF files for each of your recovered objects by booting a domain controller in each of the domains in the tree, going into DSRM, and running the following sequence of commands. You will need to copy over the text file created by the original authoritative restore to each of the DCs on which you will be running this. (This example assumes that you copied the text file to C:\restored_object.txt.)

```
Ntdsutil
ntdsutil: authoritative restore
ntdsutil authoritative restore: create ldif files from c:\restored_object.txt
```

15. Note the location of the newly created LDIF file; then quit ntdsutil and restart the domain controller into normal mode.

16. Log on to the domain controller and run the same command used in step 12 but making sure to replace c:\restored_object.ldf with the path to the LDIF file created by the commands you ran in step 14.

17. You will need to repeat steps 14 to 16 on one domain controller for each of the domains in your tree.

**MIGRATION STRATEGIES**

If you want to take advantage of the new features in Windows Server 2008 AD DS, you will need to start replacing your existing Windows 2000 Server or Windows Server 2003 domain controllers with Windows Server 2008 domain controllers. You can do this in two basic ways: You can perform in-place upgrades of your domain controllers, or bring
in new Windows Server 2008 domain controllers as you retire the older domain controllers. The latter is ideal if done in conjunction with a server refresh, since you can ensure that your new servers are up to current specifications as you decommission the older domain controllers.

The two new features in Windows Server 2008 that you should consider in planning your new domain controller architecture are the Server Core installation option and RODCs. Typically, you want your servers acting as domain controllers to perform that function and nothing else. Domain controllers are perfect candidates for a Server Core installation since you want your domain controllers to have absolutely the bare minimum number of components installed. Not only does this increase the stability and performance of your domain controllers, but it decreases the possibility for unrelated vulnerabilities from affecting your Active Directory infrastructure.

You should also revisit your strategy around any domain controllers you have deployed at remote offices. They may be better served by RODCs if no local IT staff is available to secure and maintain the domain controllers properly. This will increase the overall performance of your remote domain controller by limiting replication traffic to one direction and reducing any potential security risk that a remote domain controller possesses.

Although it’s not a hard-and-fast rule that every domain controller in your domain must be running the same operating system, it’s the ideal scenario to reduce the possibility for replication or compatibility issues. The operating systems that your domain controllers can run when participating in the domain are limited by your domain functional level. Once all your domain controllers have been upgraded to Windows Server 2008 and you are sure that no legacy domain controllers will be participating in your domain, you should raise the domain functional level to Windows Server 2008 to be able to use all the new features. If you want to deploy RODCs in your domain, you will minimally need to ensure that you have at least one Windows Server 2008 domain controller and that it is running the PDC Emulator role.

**NOTE** I wish I could include a more step-by-step checklist on the right way to perform a migration. In the real world, each company and each environment presents different and unique challenges that ultimately drive these design decisions. If you already have Active Directory in place and have planned out your Active Directory infrastructure, all that remains for you is to determine if and where RODCs belong and whether or not a Server Core installation is right for your organization.

**CHAPTER SUMMARY**

This chapter offers a lot of important content. Active Directory is one of the most critical pieces—if not the most critical piece—of infrastructure in a Windows-based network. You need to consider carefully not only your physical network infrastructure but also your overall organizational structure when planning, designing, and implementing Active Directory. In general, you want to leverage domains and OUs to organize your
directory based on your business organization, lines of authority, and areas of responsibility. Once those logical pieces are in place, you will need to create sites effectively around well-connected subnets to optimize replication traffic.

The new features of Active Directory Domain Services such as RODCs can help further increase security across your network while providing load-balancing services to improve your overall user experience. Just as important is the need for a backup and recovery solution that is not only well documented but tested on a regular basis to prepare for disaster recovery and business continuity. Windows Server 2008 doesn’t completely revamp Active Directory. Instead, it expands its functionality and gives you even more flexibility with regards to Active Directory infrastructure planning, design, and security.
This page intentionally left blank
CHAPTER 5

Windows Deployment Services
In Windows Server 2008, Windows Deployment Services (WDS) replaces Remote Installation Services (RIS) offered in previous versions of Windows Server. You can use WDS to perform “bare metal” installs (installations on computers without an operating system installed already) of base Windows operating systems without your being physically present or having access to the physical installation media. Instead, the system uses a combination of a pre-boot execution environment (PXE) and a Trivial File Transfer Protocol (TFTP) Server to boot the system from the network and load the operating system. This service provides an in-box solution that makes it easier for you to deploy Windows Server and Workstation operating systems throughout your organization.

WDS uses images created in Windows Imaging Format (WIM), a file-based imaging format unlike traditional disk imaging solutions that are sector-based. The advantage of WIM is that it is not hardware-dependent since the smallest unit within a WIM image is a file. In the WIM format, files are stored only once, even if they are referenced multiple times in the file tree. In other words, it leverages a single instance store. This makes the image smaller, and it is made even smaller since higher compression can be achieved on the files themselves. This image format is used by Windows PE (Preinstallation Environment) 2.0. Windows PE can be considered the replacement for MS-DOS as the boot environment for testing, installing, and deploying Microsoft Windows operating systems. It’s a minimal install of a Windows system that is based on the kernel of the Windows operating system in addition to some necessary services.

**BENEFITS OF USING WINDOWS DEPLOYMENT SERVICES**

The fact that WDS is available for free as part of Windows Server 2008 is a huge benefit. Although it may not be the most feature-rich of all the different OS deployment solutions on the market, it’s a good solution for quickly deploying Windows Server and Workstation operating systems with a great deal of automation at no additional cost. WDS offers the following benefits:

- Can be used to deploy Windows XP, Windows Vista, Windows Server 2003, and Windows Server 2008 to bare-metal computers
- Designed and built on top of the core Windows setup technologies (Windows PE, WIM, and image-based setup)
- Can be used to reprovision workstations and servers with a previous operating system to Windows Vista and Windows Server 2008
- Offers improved management capabilities: WDS can be managed both from an MMC snap-in and through the command line with WDSUTIL.EXE
- Integrates with Active Directory
Scalable Windows PE environment supports plug-ins using an open API for standards-based support
- Can transmit data and images via multicast

SCENARIOS FOR WINDOWS DEPLOYMENT SERVICES

If you don’t already have a system “imaging” or operating system deployment solution in place, WDS can drastically reduce the time it takes to deploy Windows-based operating systems in your environment. In addition, when used in conjunction with effective user group policies, WDS can reduce the amount of maintenance required. Since WDS is initiated by booting the computer into a PXE and then loading it over the network, even a regular user could fairly easily be walked through reloading the operating system if it had somehow been corrupted. This assumes that you have restricted access to your workstations so that they don’t have any local files to begin with. In the server space, you could have a datacenter operator simply “rack and stack” a group of servers, start them up, and initiate a server image load without the operator needing to know any intricate details about what options to select during setup. Simply put, WDS is a good option for rapidly deploying Windows operating systems throughout an environment.

WDS is part of what Microsoft likes to call a “zero-touch deployment strategy.” Loading the OS is one thing, but loading applications is a completely different story. This is why WDS is only part of the greater puzzle of Windows deployment. WDS can be used not only to load the base operating system, but when used in conjunction with distribution shares, it can also be used to load additional third-party drivers, patches, and even applications at the time of the install. This layered approach makes it easy to mix and match base images with various driver sets and applications to tailor your images with your needs.

Even if you have to load applications manually either on your workstations or your servers after the OS is installed, automating the bare installation can still save you significant time and resources—which, of course, equates to saving money. For the purpose of WDS, you will need to learn how to install and configure WDS and set up your clients PXE boot, create images, and create unattended setup files. This is significant up-front engineering work, but in an organization with hundreds of servers, it is well worth the effort.

The general procedure for WDS is to install the server, configure the server, add your images, deploy the images, and lastly, maintain your images. This last task can be a nightmare: Typically, an image is made for a particular system build with all the appropriate base applications and utilities preloaded. This is then marked as the baseline image. The problem is that as changes are made in the environment—such as application setting changes and system updates—you will eventually need to go back and update your baseline image to create a new one. Updating the baseline image typically means dumping the current baseline image, then making all the necessary updates, running sysprep to reseal it, and finally recapturing this new baseline image. Without this level of maintenance, you run the risk several months later of loading images on your network
that are not appropriately patched or are incorrectly configured. Unfortunately, this is time-consuming and in most cases error-prone. The new WIM format helps out considerably in this area since WIM images can be mounted onto the file system like regular drives and then manipulated, so you can copy down new drivers, language packs, or hotfixes and easily reseal it.

COMPONENTS

WDS comprises a number of components that interact to get the job done. For starters, the WDS Server itself hosts the core PXE server and manages communications between client and server. A TFTP Server is used to dish out images to PXE clients. Significant enhancements have been made to the TFTP Server that allow for faster communications by controlling the communication windows. Finally, a file share called REMINST points to the folder on the server where the WDS images are kept. This is used by the WDS client when uploading Install images created from Capture images (more on these images later in this chapter).

WDS INSTALLATION

I’m sure you’ve gotten the gist of what WDS is all about, but there’s nothing quite like getting your hands into it to help you understand its intricacies. Installing WDS requires that a bit of infrastructure be in place before it will function correctly. The basic requirements for installing WDS are shown in Table 5-1.

Hands-On Exercise: Installing and Configuring Windows Deployment Services

In this example, we install and configure WDS to take on WIM Images and prepare it for deployment. For simplicity, we configure WDS to respond to all clients. In production, you can opt to select more stringent security options that best fit your environment.

1. Open Server Manager.
2. Click the Add Roles link to open the Add Roles Wizard.
3. Review the preliminary tasks on the Before You Begin screen, and then click Next.
4. On the Select Server Roles screen, select Windows Deployment Services, and then click Next.
5. Review the Introduction to Windows Deployment Services and make sure you have all the prerequisites, and then click Next.
<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Directory</td>
<td>You shouldn’t install WDS on your domain controller for security reasons, but your WDS server must be either a domain controller or a member server of an Active Directory domain. It doesn’t matter what the domain or forest functional levels are.</td>
</tr>
<tr>
<td>Domain Services</td>
<td>For PXE boot to work, you must have a service to issue IP addresses to your clients. This is the role of your DHCP server. Unlike RIS, you don’t have to authorize WDS in DHCP. This is beneficial not only from a technical aspect but also from a political one, especially if the team managing WDS is different from the team managing DHCP. Microsoft refers to “a fragmentation of the PXE environment,” when multiple teams within an organization are responsible for the same environment, which can sometimes lead to ownership issues.</td>
</tr>
<tr>
<td>Domain Name System</td>
<td>Your WDS clients will need DNS to locate your WDS Server. Either way, you’re going to need this since AD DS requires it as well.</td>
</tr>
<tr>
<td>Installation media</td>
<td>You can’t install anything without the installation media; make sure your OS source media is available locally or via an accessible share on the network. The installation media for Windows Vista and Windows Server 2008 comprises multiple WIM files that are loaded based on your selection during the boot process.</td>
</tr>
<tr>
<td>An NTFS partition on the WDS server</td>
<td>A NTFS partition is required for WDS to store and secure the OS images that will be used by the WDS Server. The folder where the images are stored is shared by WDS, and NTFS is required to set up security on the share and folder to prevent unauthorized access.</td>
</tr>
<tr>
<td>Windows Server 2008</td>
<td>The WDS role can be hosted on a Windows Server 2008 server only, so a Windows Server 2008 server must be available. Technically, WDS was introduced as a hotfix for Windows Server 2003 SP1, but is only introduced as a server role in Windows Server 2008. Also, in Windows Server 2008, WDS can operate only in native mode and cannot coexist with RIS (mixed mode).</td>
</tr>
</tbody>
</table>

Table 5-1. WDS Installation Requirements
6. On the Role Services screen, verify that Deployment Server and Transport Server are checked; then click Next.
7. Confirm the installation options, and then click Install.
8. Click Close when the installation completes.
10. Expand Windows Deployment Services under Console Root.
11. Right-click the server name and select Configure Server from the context menu.
12. Review the information on the Welcome screen; then click Next.
13. Enter the path or click Browse to select where the operating system images will be stored, as shown in Figure 5-1. This must be an NTFS partition. In practice, you should specify a path to a non-system partition since you should keep your OS images separate from your main OS system files to help optimize performance and backups. Click Next.

![Figure 5-1. Specifying the location of the remote installation folder](image-url)
14. On the PXE Server Initial Settings screen, select Respond to All (Known and Unknown) Client Computers, as shown in Figure 5-2.

15. Click Finish.

16. If prompted, uncheck Add Images to Windows Deployment Services Now, and then click Finish.

**WDS PROPERTIES**

Once you’ve completed the installation and performed your initial configuration, you need to start creating and loading your OS images. Before moving onto that, let’s explore the various WDS properties that can be queried or set depending on your desired configuration.
To access the server properties, right-click the server name in the WDS console and click Properties. The server Properties dialog box consists of eight tabs:

- **General**: Displays the server name, path to the remote installation folder, and the server mode.
- **PXE Response Settings**: Controls whether to respond to all clients or just known clients. You can also set how to respond to unknown clients and the PXE response delay.
- **Directory Services**: Controls the default name given to new clients and the location in Active Directory where their accounts will be created. The default location is the Computers container. You can create and select a separate organizational unit (OU) for clients added using WDS to help you keep track of them.
- **Boot**: Specify the default boot program and boot image for x86, ia64, and x64 architectures.
- **Client**: Specify unattend files to enable full unattended mode for your DS clients. Use this to automate image selection and disk management.
- **DHCP**: Since both WDS and DHCP listen for DHCP requests on port 67, if the WDS server is also a DHCP server, you need to configure WDS not to listen on port 67 and to set DHCP option tag 60 to PXEClient.
- **Network Settings**: Configure the multicast IP address range to use, UDP port ranges, and network speed of the WDS server.
- **Advanced**: Configure WDS to use a specific domain controller or global catalog server or allow it to use any available domain controller. You can also specify whether a WDS server needs to be authorized in DHCP before it is allowed to service clients. By default, authorization is not required in DHCP.

**CREATING AN OPERATING SYSTEM IMAGE FOR WDS**

Now that the server is ready and prepped to accept new images and PXE clients, your first major task will be to create an image of an OS for use with WDS. Three different image types can be created using WDS: Capture, Discover, and Install images. Capture boot images launch the Image Capture Wizard, which is used to create an Install image.
of a volume on a reference system and, if desired, upload it to WDS server. Discover images are used to boot non-PXE-capable systems into the WDS client. You can think of the Discover image as a bootable image to get you into the WDS.

Technically, plain boot images aren’t created but are supplied with your Windows Vista and Windows Server 2008 installation media. These WIM files are used to launch the setup environment for the OS. When you install WDS from scratch, no boot images are available; you need to load each one before continuing with any of the following exercises in this chapter.

**Hands-On Exercise: Adding a Boot Image**

Although the most trivial of all the actions, this is the most crucial, since without having any boot images on your WDS server, you cannot create any other type of image.

1. Open the Windows Deployment Services console.
2. Expand your server on the left pane and right-click the Boot Images folder.
3. Choose Add Boot Image.
4. Enter or browse to the sources folder path on the Windows Server 2008 installation media where boot.wim is located (Figure 5-3), and then click Next.

![Figure 5-3. Specifying the location of boot.wim](image)
5. Enter the Image Name and Image Description in the Image Metadata screen; then click Next (Figure 5-4).

6. Review the Summary page, and click Next.

7. Click Finish when completed.

Hands-On Exercise: Creating a Capture Image

Once you have boot images in place, your first step to capturing an image is to create a Capture boot image. The Capture boot image creates a boot environment that will allow you to create an Install image. In this example, you create a Capture image for Windows Server 2008.

1. Open the Windows Deployment Services console.
2. Expand your server name on the left pane and click the Boot Images folder.
3. Right-click the name of the boot image for which you want to create a Capture boot image. In this case, select Microsoft Windows Server 2008 Setup (x86).
4. Select Create Capture Boot Image.
5. Enter the Image Name, Image Description, and full pathname where you want to save the new Capture image; then click Next (Figure 5-5). I recommend you save this image in the images folder of your Remote Install folder.
6. Click Finish when complete.
7. Right-click the Boot Images folder and select Add Boot Image.
8. Browse to and select your newly created Capture boot image; then click Next.
9. Enter a name and description for the Capture image, and then click Next.
10. Review the summary and click Next.
11. Click Finish.

Figure 5-5. Capture Image Metadata screen
Hands-On Exercise: Creating an Install Image from a Windows Server 2008 Reference System

Now that you have a Capture boot image, you can start creating Install images. This is done by capturing an image of a sysprepped reference system, and it can be done for both Windows Vista and Windows Server 2008. The reference system is then started up in PXE boot mode so that it can begin uploading the image.

NOTE This exercise creates an Install image for WDS—it does not prepare a reference system using sysprep. Review Microsoft’s documentation to learn how to use sysprep on each respective operating system for which you want to create an Install image.

In this exercise, we create an Install image for Windows Server 2008.

1. Prepare your reference system by installing Windows Server 2008 and any additional application you want to load.
2. Open the command prompt, change the directory to %systemroot%\system32\sysprep, and run the following:
   
   Sysprep /OOBE /Generalize /Reboot

3. When the computer restarts, bootup using the network interface card (NIC) into PXE mode. Usually this is done by pressing F12 during bootup or changing your boot priority in BIOS to boot using the NIC first.
4. If multiple boot images are available, a menu will present all the possible boot options. Select the name of your Windows Server 2008 Capture image.
5. Open the Windows Deployment Services Management console.
6. Expand your WDS server and right-click Install Images.
7. Select Add Image Group and enter LABTEST IMAGES in the dialog box. Then click OK (Figure 5-6).
8. After the Capture boot image has finished loading using TFTP, you will see a Windows Deployment Services Image Capture Wizard (Figure 5-7). Click Next on the Welcome screen.

9. In the Image Capture Source screen, select the volume you want to capture (this should be the volume where your system files are located) and enter the Image Name and Image Description, as shown in Figure 5-8. Click Next.

10. On the Image Capture Destination screen, click Browse and type in the path and filename where the image will be stored.

11. Check the Upload Image to WDS Server checkbox.

12. Enter the server name of your WDS server, and click Connect. If prompted, enter domain credentials of an account with permissions to upload images (typically an administrator account). Select the Image Group Name from the drop-down list and click Finish (Figure 5-9).

Figure 5-7. Windows Deployment Services Image Capture Wizard
13. After the image has been created, go back to the WDS Management console and expand your server on the left pane.

14. Expand the Install Images folder.

15. Right-click the Image Group folder to which you want this image added (in this case, LABTEST IMAGES), and select Add Install Image.

16. Browse to select the newly uploaded Install image. This should be in your Remote Install folder under the \Images\LABTEST IMAGES folder (where LABTEST IMAGES is the name of your Install image group). Then click Next (Figure 5-10).

17. Click Next on the List of Available Images screen.

18. Review the summary; then click Next.

19. Click Finish.
Hands-On Exercise: Creating a Discover Image

Although PXE boot is a very useful tool for a seamless, over-the-network installation of Windows Server 2008, it does have limitations. For example, if the computer doesn’t have PXE boot capabilities or is on the other side of a very slow WAN link, loading an OS image over the network may not be the ideal solution. In this example, we create a Discover image that is created to a file and then burned onto CDs or DVDs to be distributed and loaded at a later time.

1. Open the Windows Deployment Services console.
2. Expand your server in the left pane.
3. Click the Boot Images folder.
4. Right-click the boot image from which you want to create a Discover image and choose Create Discover Boot Image (Figure 5-11). (This assumes you’ve already added a boot image to your Boot Images folder.)

5. Enter the Image Name and Image Description.
6. Enter the full pathname where the image will be stored (Figure 5-12).

7. Enter or browse to select the Windows deployment server that will respond to the request, and then click Next. Enter the credentials for an account with administrator privileges if prompted.

8. Click Finish.

The result of the preceding steps is a Discover boot image. That solves only half of our problem. Since Discover boot images are designed to be burned onto CD or DVD media, we still have to create a bootable ISO image that we can burn onto removable media. Before you begin with this part of the exercise, download the Windows Automated Installation Kit (AIK) from Microsoft (www.microsoft.com/downloads/details.aspx?familyid=c7d4bc6d-15f3-4284-9123-679830d629f2&displaylang=en) and install it. This is a 992MB download, so it will take a while to pull it down. The AIK is set up to be burned to a DVD, so you will need to download this to a workstation or server that supports DVD burning. This doesn’t have to be done on the WDS server; it can be installed on Windows XP SP2, Windows Server 2003 SP1 and later, Windows Vista, and Windows Server 2008.
**TIP** Because the Windows AIK is constantly being enhanced, check Microsoft’s Web site to see which operating systems support the current Windows AIK.

1. Open a command prompt.
2. Change the current directory to `C:\Program Files\Windows AIK\Tools\PETools`.
3. Run the following command to create a WinPE directory structure:
   ```bash
copytpe x86 c:\WinPEx86
```
4. Delete `C:\WinPEx86\ISO\sources\boot.wim`.
5. Copy the Discover boot image you created earlier to `C:\WinPEx86\ISO\sources` and rename it `boot.wim`.
6. You should now have only one file in `C:\WinPEx86\ISO\sources`, and the filename has to be `boot.wim`.
7. Change the current directory to `C:\Program Files\Windows AIK\Tools\PETools`.
8. Run the following command to create the ISO file:
   ```bash
oscdimg -n -bC:\WinPEx86\ISO\boot\etfsboot.com c:\WinPEx86\ISO c:\WinPEx86.iso
```
9. You will now have an ISO file (`C:\WinPEx86.iso`) that you can burn onto a CD or DVD and use to bootup a system.

## LOADING YOUR INSTALL IMAGE TO YOUR CLIENTS USING WDS

So far, we’ve loaded boot images to WDS; created a Capture image from the boot image, which is nothing more than a boot image that automatically goes into the Capture Image Wizard; used the Capture image to create an Install image; and created a Discover image to boot non-PXE-enabled devices. The end goal of all this engineering effort is to load Windows operating system images to bare metal or existing PCs with minimal effort. If you try PXE booting one of your test machines to load your newly created Install image, you will probably be wondering why the boot menu still shows only your boot and Capture boot images in the PXE boot menu, even though you’ve clearly created and added an Install image to WDS.
If you read through this chapter, you should know that there are only three different types of boot images:

- **The regular boot image that is taken from OS installation media (boot.wim)**
- **Capture boot images created from regular boot images**
- **Discover boot images that are also created from regular boot images**

Notice how I never said to create an Install **boot** image. There is no such thing as an Install boot image, since they are actually called Install images. The lack of the word **boot** in *Install images* is intentional, since Install images are not bootable. You will actually need to boot using either a regular boot image using PXE or a Discover boot image, and then selecting the appropriate Install image to load when prompted.

**Hands-On Exercise: Installing Windows Server 2008 Using WDS and PXE Boot**

1. Boot the server onto which you’d like to load Windows Server 2008 using PXE (press F12 when prompted).
2. When the boot menu appears, select Microsoft Windows Server 2008 Setup (x86). Note that this menu will not appear if you have only one boot image in WDS. Also, your selection will be based on the boot image you have available, and the actual text may be different. Simply select the option that loads the boot image for Windows Server 2008 that you added from the Windows Server 2008 installation media.
3. After it boots, select the locale and keyboard or input method; then click Next.
4. When prompted, enter the credentials for a user account that is minimally a member of the Domain Users group. This is important since this clearly allows you to delegate the ability to load images without having to give a user admin privileges. In the background, this credential is used to access the REMINST share on the WDS server.
5. Select the OS you want to install, as shown in Figure 5-13, and then click Next. If applicable, you can also select the Language to install.
6. Select the partition to which you want to install, and then click Next. This will initiate the installation of Windows Server 2008.
UNATTENDED INSTALL USING WDS

So far, you’ve been able to leverage WDS to help you load your captured Install image onto new systems over the network. Although that in itself can be quite useful, you can really extract the power of WDS by utilizing its unattended install capabilities. You can think of the Windows installation process as having two phases: The first phase has the preinstallation options such as OS and language selection as well as drive partitioning. In the second phase, installation of the core operating system has completed, but you still have a number of outstanding initial configuration tasks to accomplish.

WDS allows you to automate this process by specifying unattend (answer) files to help answer the selection for you. If implemented correctly, you can automate your server installation from soup to nuts so that a junior member of the team or a less-skilled resource can complete the tasks of setting up the servers for you. They can simply bootup the server using PXE, select the appropriate OS boot menu, select the OS they want to install, and sit back while the WDS does all the dirty work.
To create a fully unattended installation, you need to create the appropriate unattend files based on what you are trying to automate—for example, WDS client, Windows setup, or legacy setup. You then need to associate these unattend files with a specific image or architecture type (by globally defining it on the server as the default for a specific architecture). You can use the Windows System Image Manager (SIM) that is part of the Windows AIK to help create your unattend.xml file.

**Windows System Image Manager**

The Windows SIM is part of the Windows AIK. Although not part of WDS, SIM is a critical tool in developing unattended installations. You can still create an unattend.xml file using nothing but Notepad, but that process is error-prone. Using SIM makes it a lot easier not only to configure the options you want to set but also to explore other available options—for example, customizing Internet Explorer as part of your Windows Server 2008 or Windows Vista installation. The Windows SIM also has the added advantage that it can verify the validity of your unattend.xml file as it’s created. Just like a compiler, it will display error messages and warnings to indicate if you have entered invalid data or warn you if you have included options but not specified any values. Windows SIM is also context-sensitive, so the options available in the answer file vary depending on the type of Windows image you have loaded.

Open the Windows SIM, and you will see that it is divided into five distinct panes (Figure 5-14): Distribution Share, Windows Image, Answer File, Properties, and Messages, as follows:

- **Distribution Share**  
  Create or select a distribution share; each share contains additional software and third-party drivers you may want to load as part of a Windows installation. A distribution share contains three folders: $OEM$ Folders, Out-of-Box Drivers, and Packages. The $OEM$ folders can contain software you want to install automatically as part of the installation. Out-of-Box Drivers can contain third-party drivers you want to make available during install. Packages are files provided by Microsoft such as hotfixes, security updates, service packs, language packs, and modifications to Windows features.

- **Windows Image**  
  Add WIM files, which not only allows your images to be organized but also works as a context switch for answer files you will create.

- **Answer File**  
  Adds entries for your answer (unattend) file.

- **Properties**  
  Displays additional properties for any option you select in your answer file.

- **Messages**  
  Displays success, error, or warning messages when compiling your answer file. Double-click an error or warning and you are directed to the property that triggered the message.

When you create an answer file, you will have to add one or more components from the Windows Image pane to the Answer File pane. The components can be added only to very specific configuration passes. In total, seven configuration passes occur during the installation and configuration of Windows Server 2008, as shown in Table 5-2.

In this exercise, you will create an unattend.xml file for a Windows Server 2008 installation. To complete this exercise, you will need to install the Windows AIK and have the Windows Server 2008 installation media available. The unattend.xml file you create will provide the bare minimum options for providing an unattended installation. You won’t create a distribution share for this exercise, but if you are going to create an image that will require the loading of third-party device drivers or you want to perform unattended installs of other software during installation, you need to create and use a distribution share. Refer to the Windows System Image Manager documentation for more advanced options.

1. Log on to the computer where Windows AIK is installed.
2. Insert the Windows Server 2008 installation media into the CD/DVD-ROM drive.
<table>
<thead>
<tr>
<th>Pass</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pass 1: windowsPE</td>
<td>Windows setup settings, including basic setup options and the creation and formatting of partitions and setting product keys. All the information you specify during the initial installation phase of a Windows Server 2008 installation can be configured in this component pass.</td>
</tr>
<tr>
<td>Pass 2: offlineServicing</td>
<td>Updates are applied to a Windows Image, including applying packages.</td>
</tr>
<tr>
<td>Pass 3: specialize</td>
<td>Applies system-specific information such as domain information and network settings.</td>
</tr>
<tr>
<td>Pass 4: generalize</td>
<td>Sets options that must persist even after running <code>sysprep /generalize</code>. Runs only if you run <code>sysprep /generalize</code>, so for the purpose of WDS, this isn’t used. The steps here are used to remove unique identifiers such as SIDs.</td>
</tr>
<tr>
<td>Pass 5: auditSystem</td>
<td>This phase is executed when the system is booted in audit mode. Audit mode is a bootup mode used by OEMs and corporations to make changes to a Windows image without going through Windows Welcome, which is the full out-of-box experience (that is, Welcome screen and other options that are configured the first time a user logs on). In this mode, you get to the Windows desktop to perform the customizations faster. Any configuration you want done when the system boots and before a user logs in to audit mode can be added to this component configuration pass.</td>
</tr>
<tr>
<td>Phase 6: auditUser</td>
<td>Similar to auditSystem in that it runs only when a Windows installation is started in audit mode. The difference is that the components you run in this mode run after a user logs on to a computer in audit mode.</td>
</tr>
<tr>
<td>Phase 7: oobeSystem</td>
<td>OOBE stands for <em>Out-of-Box Experience</em>. In this pass, you can customize any setting you want on Windows before Windows Welcome starts. For example, customizations to the Internet Explorer interface, such as adding your corporate branding, can be specified.</td>
</tr>
</tbody>
</table>
4. Right-click the Windows Image pane and choose Select Windows Image.
5. Select Catalog files (*.clg) from the Files of Type drop-down list and select \Sources\install_Windows Longhorn SERVERENTERPRISE.clg on the Windows Server 2008 installation media; then click Open, as shown in Figure 5-15.
6. Right-click the Answer File pane and select New Answer File.
7. In the Windows Image pane, expand Windows Longhorn SERVERENTERPRISE and then expand Components.
8. Expand x86_Microsoft-Windows-Setup_6.0.6001.16510neutral and highlight DiskConfiguration, as shown in Figure 5-16. (The version number might be different on your computer based on the version of the Windows Server 2008 installation media you provided.)
9. Right-click Disk and select Add Setting to Pass 1 windowsPE.
10. In the Answer File pane, expand Components\1 windowsPE\x86_Microsoft-Windows-Setup_neutral\DiskConfiguration\Disk (Figure 5-17).
11. Select Disk, and in the Disk Properties pane, set the following values under Settings:
   DiskID: 0
   WillWipeDisk: true

12. Right-click CreatePartitions, and select Insert New CreatePartition.

13. In the CreatePartition Properties, set the following values to create a 10GB primary partition:
   Order: 1
   Size: 10000
   Type: Primary

14. Right-click ModifyPartitions, and then select Insert New ModifyPartition.
15. In the ModifyPartition Properties, set the following values to format and configure the partition you will create as a result of step 14 (Figure 5-18):
   Active: true
   Extend: false
   Format: NTFS
   Label: Local Disk
   Letter: C
   Order: 1
   PartitionID: 1


17. Right-click InstallTo and choose Add Setting to Pass 1 windowsPE.
18. Under the Answer File pane, make sure the Components\1 windowsPE\x86_Microsoft-Windows-Setup_neutral\ImageInstall\OSImage\InstallTo component is selected, and then in the InstallTo Properties, set the following values:
   - DiskID: 0
   - PartitionID: 1


20. Right-click UserData and choose Add Setting to Pass 1 windowsPE.

21. In the Answer File pane, select Components\1 windowsPE\x86_Microsoft-Windows-Setup_neutral\UserData. Then in the UserData Properties screen, set the following values:
   - AcceptEULA: true
   - FullName: John Smith
   - Organization: MyCorporation Inc
22. Expand the UserData component in the Answer File pane and then select ProductKey.

23. In the ProductKey Properties, enter the following:
   Key: Your Windows Server 2008 product key in the format 12345-12345-12345-12345-12345
   WillShowUI: OnError

24. Finally, your last configuration step is to assign a password for the default Administrator account. In the Windows Image pane, expand Components\x86_Microsoft-Windows-Shell-Setup_6.0.6001.16510_neutral\UserAccounts.

25. Right-click AdministratorPassword and choose Add Setting to Pass 7 oobeSystem. Since you added it to the oobeSystem pass, this will set the Administrator password when the system boots into Welcome Screen mode but before the user first logs on.

26. In the Answer File pane, select Components\oobeSystem\x86_Microsoft-Windows-Shell-Setup_neutral\UserAccounts\AdministratorPassword.

27. In the AdministratorPassword Properties, set the following:
   Value: P@ssword123
   Obviously, you can change P@ssword123 to whatever your default Administrator password should be. This is encrypted in the unattend.xml file, so you don’t have to worry about anyone retrieving this password later.

28. On the Windows SIM menu bar, choose Tools | Validate Answer File. Verify that no error messages are displayed in the Message pane. If any errors or warnings appear, you may have forgotten to fill out one of the fields. Go back and make any necessary changes.

29. Choose File | Save Answer File.

30. Save this answer file to the WdsClientUnattend folder in your RemoteInstall folder on your WDS server. You can give this any filename, but to make it descriptive, call it server2008en.xml.

---


Now that you have created an answer file, you need to tell WDS to use that answer file whenever installing Windows Server 2008 Enterprise Edition.

1. Log in to your WDS server.
2. Expand your server in the WDS Management console.
3. Expand the Install Images folder in the WDS console.
4. Click the image group containing your Windows Server 2008 Base Image (Figure 5-19).
6. On the General tab, check the Allow Image to Install in Unattend Mode checkbox, as shown in Figure 5-20.

![Figure 5-19. Selecting an image group](image1)

![Figure 5-20. Allowing Windows Server 2008 to install in unattend mode](image2)
7. Click the Select File button.
8. Enter the path or click Browse to select the unattend file you created in the preceding exercise (Figure 5-21); then click OK.
9. Click OK in the Image Properties dialog box to save the changes.

You can now install Windows Server 2008 in unattend mode on a server by booting it up using PXE, selecting your Windows Server 2008 Setup (x86) boot image, and then selecting Windows Server 2008 Base Install from the list of available Install images.

**NOTE** Since you specified the creation of a 10GB partition in your unattend file, the server must have at least a 10GB hard drive for the partition creation to be successful.

**CHAPTER SUMMARY**

The ability to automate the installation and configuration of Windows Server 2008 can make or break a Windows Server 2008 rollout strategy for many organizations. Windows Deployment Services is an excellent in-box solution that can ease deployment of Windows Server 2008 in your organization by providing a mechanism for managing various Windows Server configurations and loading them over the network. Although significant improvements have been made in both functionality and performance of WDS when compared to RIS, the end-to-end process of loading, installing, and then configuring your server through WDS is still fairly lengthy when compared to some other third-party imaging solutions. That said, although the entire process can be lengthy, it can also be fully automated, so while the duration may be long, the actual effort required by you, or whoever will be loading the
images onto your servers, is minimal. This will work well in many organizations: You can start loading images onto your servers and do other things while the images load. When you return, all your servers will be completely set up and ready for any post-installation tasks. The key to remember is that a boot image is used to PXE boot into a Windows PE. Capture boot images boot into the Windows Capture Wizard to capture an Install image from a syspreped reference system. You can use an Install image to install Windows Server 2008 on a system. Discover boot images allow you to capture images from a system that doesn’t support PXE boot. Once loaded, you can create answer files using the Windows SIM to automate as little or as much of the installation process as you want. This ultimate level of flexibility is what makes WDS so powerful and what makes the new unattend XML format so much more powerful the collection of answer and configuration files you used with previous Windows Server builds. However, it is possible to deploy previous Windows Server and Workstation builds using WDS, but the process won’t be nearly as seamless.

Creating the ultimate rollout and deployment strategy for your organization will bear heavily on how well you can take advantage of these new tools. The underlying setup process for Windows Server 2008 is a huge leap from what it was even with Windows Server 2003. In fact, it may benefit you to unlearn some of your previous Windows deployment techniques and absorb some of the new terminology and concepts. The Windows Setup that is part of Windows Server 2008 is much more component driven than ever before and allows for a very granular definition of each option you want to enable. A well-defined answer file combined with an intelligent set of group policies can make for a potent tool for simplifying the deployment and management of your Windows Server 2008 environment.
This page intentionally left blank
Although Internet Information Services (IIS) security has gotten better throughout the years, for the most part, even IIS 6.0 was simply a more functional form of IIS 4.0. When you installed previous versions of IIS, almost everything was installed by default, and you were provided only a handful of components that could be selectively installed. Version 7.0 is far more granular and follows the entire Windows Server 2008 mantra, “What doesn’t get installed won’t need to get patched.”

This new version was built from the ground up to be more modular. This reduced attack surface not only increases security, but it can potentially minimize downtime related to system maintenance. Fewer patches means fewer server restarts as a result of a patch install. Version 7.0 also includes a new management interface that is more task oriented. Until now, IIS has kept to its bland interface, which dates back to IIS 4.0. The IIS 7.0 administrative interface is far better organized and more functional, using a dashboard-style design with task-oriented panes and easy-to-filter selections.

The granular component design of IIS 7.0 is made easier to manage by allowing administrators to delegate a significant number of IIS management tasks back to the developer or Web site owner. If you host a large number of Web sites but you don’t own the content, this delegation model gives you the flexibility to hand off common administrative tasks in a secure manner, removing yourself as the administrative bottleneck for simple configuration changes that don’t affect the stability of your server.

Version 7.0 also contains additional performance and troubleshooting capabilities that didn’t exist in previous versions. For example, you can see every request coming into your server so you can track down the cause of problems or decipher which application or request is using server resources. IIS 7.0 is not just another pretty coating on top of an aging Web server architecture. It is a complete redesign of IIS and addresses many of the complaints that both IIS administrators and developers have been voicing for years. In this chapter, we explore the various aspects of IIS and how you can take full advantage of them.

**IIS 7.0 FEATURES**

IIS 7.0 is designed around a set of key objectives and provides the following features:

- **Ability to delegate administration**
- **Flexible extensibility model to allow for customizations that have the added effect of reducing the attack surface and increasing security**
- **Integrated application and health management**
- **Increased ability for diagnostic and troubleshooting (more insight into what IIS is doing)**
- **Much more intuitive administration tools**
- **True application Xcopy deployment**
With IIS 7.0 you can granularly delegate administrative control to Web site developers or site owners. The installation is fully customizable and allows you to select only those specific components you want to install and enable. Developers also have much richer application programming interfaces (APIs) to extend IIS functionality. As an administrator and not a developer, you could probably care less whether new APIs are included, but the reality is that these new APIs allow you as the administrator to write code to manage all aspects of IIS through the Microsoft.Web.Administration namespace or through the WebAdministrator Windows Management Instrumentation (WMI). IIS 7.0 can also check application health through Windows Communication Foundation (WCF) services such as Windows Activation Service (WAS), which provides intelligent resource management, process tracing, and automatic failure detection. For example, if a request times out, IIS can automatically log a traceback through the code that generated the exception to help you track server issues.

The internal workings of IIS 7.0 have also been exposed. You can now get in-depth information about IIS activity at any time, such as the types of requests that are coming in, which resources are being accessed, and what they’re doing. This makes it a bit easier to troubleshoot and diagnose server or application issues. Not only does IIS have a new task-focused user interface, it also includes a new command-line tool called APPCMD .EXE that can be used to query or configure any of the many options and configuration settings available in IIS. If you have an inclination toward scripting, you’ve probably dropped this book just to read more about it (but this tool is covered later in this chapter). With an easy-to-use command-line interface and the ability to write managed code to interface directly with the Web server administration components, you’ll find it much easier to reach whatever level of automation you want from IIS than ever before.

Finally, one of IIS 7.0’s best new strengths is the ability to deploy applications by doing nothing more than running Xcopy. Site- or application-specific configuration settings can be stored in web.config files along with the application, so that as soon as you copy the folder to a new server, the configuration is instantly enabled. This functionality has existed in competing Web server products for years, and it’s good to see this finally implemented in IIS 7.0.

**Hands-On Exercise: Installing IIS 7.0**

Although I love knowing the technical details of how things work, what I like most is working with the product in front of me. Before we move on to the rest of the exciting features in IIS 7.0, let’s install IIS 7.0 on a Windows Server 2008 server.

1. Open Server Manager if you don’t already have it open.
2. Click the Add Roles link under the Roles Summary screen to initiate the Add Roles Wizard.
3. Verify that you have completed the tasks listed in the Before You Begin screen, and then click Next to continue.
4. Select Web Server (IIS) from the Select Server Roles screen.
5. You will be asked to add features required for Web Server (IIS), as shown in Figure 6-1. Click Add Required Features, and then click Next.
6. Read through the Introduction to Web Server (IIS) to make sure you won’t have any issues on your server; then click Next.
7. In the Select Role Services screen (Figure 6-2), the major components of IIS have been preselected. You can select any other service you want here, such as ASP.NET, ASP, CGI, FTP, and more. For now, keep the default settings and click Next. The details of each of these role services are listed in Table 6-1.
8. Review the installation options, and then click Install to begin the installation.
9. Click Close when the installation completes.
Chapter 6: Internet Information Services 7.0

UNATTENDED INSTALLATION

If you have to set up many IIS servers and intend to install the same general options on all of them, you can automate the installation process by performing an unattended installation. This is done using a command-line tool called pkgmgr.exe. This tool is used to install any Windows optional features in Windows Server 2008. Using pkgmgr.exe, you can perform an unattended installation in two ways: You can specify the packages you want installed at the command line using the /iu switch or create an unattend XML file that contains the list of options you want installed.

Each IIS component listed in Table 6-1 is provided with an abbreviated name version (specified in parentheses after the full descriptive name) that is used by pkgmgr.exe. You will need to compile a list of all the components you want to install and then either specify them on the command line along with pkgmgr.exe or put them all in an unattend XML file that you pass to pkgmgr.exe to use as its input.
Service:  
**Web Server (IIS-WebServer)** Installs the IIS 7.0 Web server, the parent component for all optional Web site components such as HTML, ASP, and ASP.NET (default selected).

Feature:  
**Common HTTP Features (IIS-CommonHttpFeatures)** When installed, allows the server to serve static Web content such as HTML files, images, custom errors, and redirection (default selected).

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static Content (IIS-StaticContent)</td>
<td>Allows the server to serve static content (default selected).</td>
</tr>
<tr>
<td>Default Document (IIS-DefaultDocument)</td>
<td>Allows you to specify a default file to serve when none is specified (default selected).</td>
</tr>
<tr>
<td>HTTP Errors (IIS-HttpErrors)</td>
<td>Makes HTTP error files available (default selected).</td>
</tr>
<tr>
<td>HTTP Redirection (IIS-HttpRedirect)</td>
<td>Allows you to redirect requests to an alternate location (default selected).</td>
</tr>
</tbody>
</table>

Feature:  
**Application Development (IIS-ApplicationDevelopment)** When installed, allows Web application support such as classic ASP, ASP.NET, CGI, and ISAPI (default selected).

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASP.NET (IIS-ASPNET)</td>
<td>Allows ASP.NET applications to be hosted.</td>
</tr>
<tr>
<td>.NET Extensibility (IIS-NetFxExtensibility)</td>
<td>Allows .NET Framework managed module extensions (default selected).</td>
</tr>
<tr>
<td>ASP (IIS-ASP)</td>
<td>Allows classic ASP pages to be hosted.</td>
</tr>
</tbody>
</table>

Table 6-1. IIS 7.0 Role Services Matrix
CGI (IIS-CGI) | Allows CGI scripts to be hosted.
---|---
ISAPI Extensions (IIS-ISAPIExtensions) | Allows ISAPI extensions to be hosted.
ISAPI Filters (IIS-ISAPIFilter) | Allows ISAPI filters to modify Web server behavior.
Server Side Includes (IIS-ServerSideIncludes) | Allows .stm, .shtm, and .shtml include files.

**Feature:**
**Health and Diagnostics (IIS-HealthAndDiagnostics)**
When installed, allows you to monitor and manage your server and application health (default selected).

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP Logging (IIS-HttpLogging)</td>
<td>Enables logging of Web site activity (default selected).</td>
</tr>
<tr>
<td>Logging Tools (IIS-LoggingLibraries)</td>
<td>Installs logging tools and scripts (default selected).</td>
</tr>
<tr>
<td>Request Monitor (IIS-RequestMonitor)</td>
<td>Allows you to monitor server, site, and application health (default selected).</td>
</tr>
<tr>
<td>Tracing (IIS-HttpTracing)</td>
<td>Allows tracing of ASP.NET applications (default selected).</td>
</tr>
<tr>
<td>Custom Logging (IIS-CustomLogging)</td>
<td>Allows custom logging of servers, sites, and applications</td>
</tr>
<tr>
<td>ODBC Logging (IIS-ODBCLogging)</td>
<td>Allows logging to an ODBC-compliant data source.</td>
</tr>
</tbody>
</table>

**Feature:**
**Security (IIS-Security)**
When enabled, allows additional security layers for the Web server (default selected).

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Authentication (IIS-BasicAuthentication)</td>
<td>Allows regular clear text usernames and passwords.</td>
</tr>
</tbody>
</table>

### Table 6-1. IIS 7.0 Role Services Matrix (Continued)
### Component Description

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digest Authentication (IIS-DigestAuthentication)</td>
<td>Allows authentication by password hashes sent to domain controllers.</td>
</tr>
<tr>
<td>IIS Client Certificate Mapping Authentication (IIS-IISCertificateMappingAuthentication)</td>
<td>Allows mapping of client certificates using one-to-one or many-to-one Windows account mappings.</td>
</tr>
<tr>
<td>URL Authorization (IIS-URLAuthorization)</td>
<td>Allows authorization of client access to URLs containing Web applications.</td>
</tr>
<tr>
<td>Request Filtering (IIS-RequestFiltering)</td>
<td>Allows rules to be configured to block specific client requests (default selected).</td>
</tr>
<tr>
<td>IP and Domain Restrictions (IIS-IPSecurity)</td>
<td>Allows access to be granted based on IP address or domain name.</td>
</tr>
</tbody>
</table>

**Feature:**

**Performance (IIS-Performance)**  When installed, enables performance optimizations to be performed (default selected).

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static Content Compression (IIS-HttpCompressionStatic)</td>
<td>Allows compression of static content when served (default selected).</td>
</tr>
<tr>
<td>Dynamic Content Compression (IIS-HttpCompressionDynamic)</td>
<td>Allows compression of dynamic content when served.</td>
</tr>
</tbody>
</table>

**Table 6-1.**  IIS 7.0 Role Services Matrix *(Continued)*
Service:

Features:
IIS Management Console (IIS-ManagementConsole) Installs the IIS Management Console (default selected).
IIS Management Scripts and Tools (IIS-ManagementScriptingTools) Installs scripts and tools used for local IIS management.
Management Service (IIS-ManagementService) Allows remote management of IIS.
IIS 6 Management Compatibility (IIS-IIS6ManagementCompatibility) Allows IIS 7.0 to be managed using existing IIS 6.0 APIs.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IIS 6 Metabase Compatibility (IIS-Metabase)</td>
<td>Installs IIS metabase to allow metabase calls.</td>
</tr>
<tr>
<td>IIS 6 WMI Compatibility (IIS-WMICompatibility)</td>
<td>Installs IIS 6.0 WMI scripting interfaces.</td>
</tr>
<tr>
<td>IIS 6 Scripting Tools (IIS-LegacyScripts)</td>
<td>Installs IIS 6.0 scripting tools.</td>
</tr>
<tr>
<td>IIS 6 Management Console (IIS-LegacySnapIn)</td>
<td>Installs IIS 6.0 Management Console; can be used to manage existing IIS 6.0 servers but not IIS 7.0 servers.</td>
</tr>
</tbody>
</table>

Service:
FTP Publishing Service (IIS-FTPPublishingService) Installs FTP support.

Features:
FTP Management Snap-in (IIS-FTPManagement) Installs FTP Server Management Console.

Table 6-1. IIS 7.0 Role Services Matrix (Continued)
NOTE Since IIS 7.0 is dependent on WAS, you will need to install the following components in addition to any of the IIS components: WAS-WindowsActivationService, WAS-ProcessModel, and WAS-ConfigurationAPI.

Hands-On Exercise: Unattended Installation of IIS Using pkgmgr.exe

In this example, we install IIS with all the default features, first using the command-line parameter method and then the unattend XML file method.

To install all the default features of IIS using the command line, run the following:

```
Start /w pkgmgr.exe /iu:IIS-WebServerRole;WAS-WindowsActivationService;
WAS-ProcessModel;WAS-NetFxEnvironment;WAS-ConfigurationAPI
```

NOTE If you open Task Manager as the various components are being installed, you will see that a process called TrustedInstall.exe is running. This is the installer. After the installation is completed, the new Web Server role may not be instantly visible in Server Manager. Re-open Server Manager and Roles and it will refresh and display the IIS that has been installed.

To perform a default install of IIS with an unattend XML file, create a file called unattend.xml with the following contents:

```
<?xml version="1.0" ?>
<unattend xmlns="urn:schemas-microsoft-com:unattend"
<servicing>
<!-- Install a selectable update in a package that is in the Windows Foundation namespace -->
<package action="configure">
<assemblyIdentity
name="Microsoft-Windows-Foundation-Package"
version="6.0.6001.17051"
language="neutral"
processorArchitecture="x86"
publicKeyToken="31bf3856ad364e35"
versionScope="nonSxs"
/>
Replace the version value with the exact version of your Windows Server 2008 installation. To find out this information, follow these steps:

1. Open Windows Explorer.
2. Navigate to %WINDIR%.
3. Right-click Explorer.exe and select Properties.
4. Click the Details tab. You’ll see the Windows version information there.

You will also need to replace the processorArchitecture value in the preceding code with whatever architecture your server uses. Your options are x86, x64, and amd64.

Finally, to perform the unattended install, assuming that this unattend.xml file is saved as C:\unattend.xml, run the following:

```
Start /w pkgmgr /n:c:\unattend.xml
```

## IIS Management Console

One of the most significant changes to IIS is its overall management user interface: IIS Manager. If you select the default role services when you install IIS, the IIS Management Console is automatically installed for you. The IIS Manager can be launched by choosing Start | Internet Information Services (IIS) Manager | Administrative Tools. You will see the IIS Start Page, with several panes: Recent Connections to IIS servers you have managed, a collection of quick links to Connection Tasks, Online Resources, and IIS News (Figure 6-3). In the left-most Connections pane are connections to various IIS 7.0 servers. By default, the local server automatically appears here. Additional servers can be added by clicking the Create New Connection button (the globe with a plug in it). To view your server’s configuration, click your server name in the Connections pane (Figure 6-4).
If you expand your server in the Connections pane, you will see two containers: Application Pools and Web Sites. Use application pools to group applications, typically to isolate different applications. Since each application pool is associated with its own worker thread, you don’t have to worry that errors that occur in one application will affect another application as long as the applications are in different application pools. By default, a DefaultAppPool is created during installation. Unless you create new application pools, all your applications will be run under this single application pool. The Web Sites folder contains all the Web sites on your server. The type of content your Web sites can contain is determined by the role services you installed. By default, IIS 7.0 installs
and supports serving only static Web pages (regular HTML files). If you want support for dynamic content such as Active Server Pages (ASP), ASP.NET, or Common Gateway Interface (CGI) script, you will need to install those role services as well. This can be done at any time using Server Manager’s Add Role Services Wizard (more on that later in the chapter). As with previous IIS versions, a default Web site is created when IIS is installed and still points to C:\Inetpub\wwwroot.

The Home pane at the center can be used to view the features or the content of the selected object in the Connections pane. This can be toggled by clicking either the Features View or Content View button at the bottom of the Home pane. If you click Features View,
the various configurable properties are displayed in the Home pane. They can be grouped either by Category or Area, or you can choose No Grouping. The grouping can be changed by selecting the appropriate grouping level in the Group By drop-down list. Furthermore, just as in a Windows Explorer view, you can click the Window icon to the right of the Group By drop-down list and set the view to Details (default), Icons, Tiles, or List.

The right-hand Actions pane contains links to context-sensitive actions based on the selections in either the Connections pane or the Home pane. For example, if you click the server name in the Connections pane, the options to stop and start the server are made available in the Actions pane. If you select a Web site, the Actions pane contains links to let you stop and start the site, view applications and virtual directories, and edit its various properties. The familiar and useful Browse and Explore icons are also available.

Now that you’ve received a quick tour of the new IIS Manager, let’s examine the Home pane a bit more closely, since this is where you will do a majority of your configuration work. With IIS and ASP.NET properties now fully intertwined in IIS 7.0, the old-fashioned tabbed interface is too cumbersome. Instead, what would have been a separate tab in the previous interface is represented by an icon in the Home pane. In the default Details view, the link to each of these features is presented by an icon, its name, and a description. The ability to filter the features you want displayed makes this user interface much more user friendly than previous interfaces. Double-click the feature name to access the various properties that can be configured for that feature. Depending on the data that the feature is presenting, it can take on the appearance of one of three different page layouts: List, Property Grid, and Dialog.

A List page layout is typically used when a list of items needs to be presented. For example, if you double-click the MIME Types feature icon, a list of all the MIME types defined on your Web site are presented in a list (Figure 6-5). A Property Grid layout is used to display various properties that can be configured. Property displays can use Friendly Names and Configuration Names as well as values. You can select whether you want the Property Grid to display the Friendly Names or the Configuration Names, or even both, by selecting the display type from the Display drop-down list at the top of the Property Grid (Figure 6-6). Friendly Names are user-friendly names that make it easier for the user of the interface to understand what properties are being set. Configuration Names are the internal configuration property names that the server actually uses in its configuration.

Dialog pages behave more like dialog boxes and have elements such as drop-down menus, checkboxes, and other typical dialog box–type elements. When changes are made to a Dialog page, you can apply or cancel the changes by clicking the appropriate button in the Actions pane. If there are any important alerts that the IIS Manager would like to bring to your attention, they are displayed in the Alerts pane on top of the Actions pane (see Figure 6-7). This modular approach makes the user interface much easier to extend. If additional features are developed in the future, you can create an appropriate page layout to represent the properties of that feature and attach it to the IIS Manager interface. This is easier than modifying the numerous tabs to integrate the feature into the old IIS interface.

**TIP** You may have noticed the address bar at the top of the IIS Manager Console (see Figure 6-4). This functions similar to the Windows Explorer address bar, in that as you drill down through the hierarchy of servers, Web sites, and properties, you can use the address bar to go back to previous screens or quickly go up to a higher level within the hierarchy.
Figure 6-5. List page

Figure 6-6. Property Grid page
By default, remote management of IIS is disabled in IIS 7.0. Before you can begin remote administration, you must install the Web Management Service WMSVC, which is set to manual startup after installed; enable remote administration; configure any additional settings such as certificates or IP/domain restrictions; and then start the WMSVC. Optionally, you can set the startup type of the WMSVC to automatic so that the server starts up automatically whenever the server is rebooted.

**Hands-On Exercise: Installing and Enabling IIS Remote Management**

In this exercise we install and enable IIS WMSVC as well as configure it to start automatically whenever the server reboots.

1. Open Server Manager.
2. In the Roles Summary section of the main page, click Web Server (IIS). Optionally, you can expand Manage Roles in the left pane and select Web Server (IIS) from there.
3. On the Web Server (IIS) screen, click Add Role Services to start the Add Role Services Wizard.

4. On the Select Role Services screen, scroll down to the Management Tools (Installed) section and check the Management Service checkbox (Figure 6-8). Then click Next.

5. Click Install to install the IIS Web Management Service.

6. When the installation completes, click Close.

7. Close Server Manager.

8. Open the IIS Manager.

9. Click your server name in the Connections pane.

10. Double-click Management Service in the feature name list.

11. In the Management Service Dialog page (Figure 6-9), check the Enable Remote Connections checkbox, and then click Apply in the Actions pane to save the changes. Optionally, you can click the Allow and Deny buttons in the IP and Domain Restrictions section to create restrictions on which computers can remotely manage your server.

**Figure 6-8. Selecting the Management Service for installation**
12. Click Start in the Actions pane to start the Management Service.


14. Double-click Web Management Service, and then change the startup type from Manual to Automatic. Click OK to save the changes.

15. Close the Services Management Console.

ADMINISTRATION USING APPCMD.EXE

As exciting as the new interface is, the true test of its power is its ability to automate configuration changes. The more you can automate, the more repeatable and predictable a process becomes and the more time you save in administering your site. IIS 7.0 includes a command-line administration tool called APPCMD.EXE. Almost everything that can be done via the IIS Manager GUI can be done from the command line. Of course, as with any
commands that can be executed at the command line, they can also be incorporated into various scripts to automate repetitive tasks. APPCMD.EXE is located in `%WINDIR%\system32\inetsrv`. This path doesn’t exist in the default system or user path, so you’ll have to add it to your path environment or simply navigate to `%WINDIR%\system32\inetsrv` from the command prompt to run it.

Not only can APPCMD.EXE be used to configure your server, it can also be used to query information about the objects on your server and to query requests coming into your server. For example, run the following to list all the sites on your server:

```
Appcmd.exe list SITE
```

This command shows the names of all the sites on your server, its internal identifier, bindings (protocol and port), and state. The results of this command are shown in Figure 6-10. As you can see, Default Web Site is starting and is listening to HTTP requests on port 80. If you like to write shell scripts, you can see that the output of this command is very script-friendly by having the output delimited by unique delimiters. This is useful, since elements such as the site’s ID are a required parameter in other commands.

APPCMD.EXE commands always follow this syntax:

```
APPCMD <verb> <object-type> [identifier] [-argument1:value1 …]
```

The only two required parameters are `<verb>` and `<object-type>`. In the preceding site listing example, the object-type is `SITE` and the verb is `list`. The verbs (actions) available depend on the object-type against which APPCMD.EXE is being run. Identifiers and arguments are generally optional but may be required for certain command combinations. The supported object types are listed in Table 6-2.
Returning to the APPCMD.EXE example, you can see how parameters are used by looking at a more specific version of that command. The preceding command lists all the virtual sites on your server; if you want to see the virtual site information about a specific site—for example, the default Web site—you can run the following command:

```appcmd.exe list site "Default Web Site"
```

In the output of the APPCMD.EXE list SITE command, you will notice a number of comma-delimited properties displayed in name/value pairs enclosed in parentheses. You can refine the output to any of these properties by specifying it as a parameter—for example, to show all started virtual sites, you can run this:

```appcmd.exe list site /state:started```

You can also use APPCMD.EXE to create a virtual site. To find out what parameters are required to create a site, run the following command:

```appcmd.exe add site /?```

From that command’s output, you’ll see that this command has four required parameters: name, id, bindings, and physicalPath. The name is the name of your site. If a space appears in the name of the site, simply enclose the site name in double quotation marks ("Test Website"). The id is the unique numeric ID used internally to identify the site within the server. It can be any number as long as it’s unique. The bindings specify the

<table>
<thead>
<tr>
<th>Object Types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SITE</td>
<td>Administration of virtual sites</td>
</tr>
<tr>
<td>APP</td>
<td>Administration of applications</td>
</tr>
<tr>
<td>VDIR</td>
<td>Administration of virtual directories</td>
</tr>
<tr>
<td>APPPOOL</td>
<td>Administration of application pools</td>
</tr>
<tr>
<td>CONFIG</td>
<td>Administration of general configuration sections</td>
</tr>
<tr>
<td>WP</td>
<td>Administration of worker processes</td>
</tr>
<tr>
<td>REQUEST</td>
<td>Administration of HTTP requests</td>
</tr>
<tr>
<td>MODULE</td>
<td>Administration of server modules</td>
</tr>
<tr>
<td>BACKUP</td>
<td>Administration of server configuration backups</td>
</tr>
<tr>
<td>TRACE</td>
<td>Working with failed request of trace logs</td>
</tr>
</tbody>
</table>

**Table 6-2. APPCMD.EXE Supported Object Types**
protocol, address, and port to which this virtual site will listen. The **physical path** is the full pathname to the root of the Web site.

The following command creates a site called Test Website that will listen to HTTP traffic on port 8010 and will point to C:\inetpub\TestWebsite as the physical path. Since the ID can be any number, but must be unique, I use 8010 as the ID for simplicity:

```
APPCMD.EXE add site /name:"Test Website" /id:8010 /bindings:"http://*:8010:" /physicalPath:"C:\inetpub\TestWebsite"
```

When you create a site using APPCMD.EXE, you’ll also create and associate an application and a virtual directory object to that site. You can find out which applications are associated with a site and which virtual directories are associated using APPCMD.EXE. For example, given the Test Website just created, you can run the following commands first to determine what applications are associated to it and then, based on the applications, you can find what virtual directories are associated with those applications.

```
APPCMD.EXE list APPS /site.name:"Test Website"
APPCMD.EXE list VDIR /app.name:"Test Website/"
```

The output of the `list APPS` command against the site name “Test Website” returns an application called “Test Website/” that is associated with the application pool called DefaultAppPool. The `list VDIR` command requires an application name as its parameter, so we use the information retrieved from `list APPS`, which in this case is the application name “Test Website/”; we can then determine that the virtual directory for “Test Website/” is C:\inetpub\TestWebsite, just as we had specified when we created it.

APPCMD.EXE can also be used to back up and restore the IIS global configuration. Whenever you make major configuration changes to your IIS server, it’s a good idea to back up this global configuration just in case it is modified inadvertently and you need to restore it to a good configuration. To create a configuration backup and call it IIS_Backup, you can run this:

```
APPCMD.EXE add backup IIS_Backup
```

**TIP** You can omit the identifier, which in this case is IIS_Backup, if you don’t care what the backup is called. If you do that, APPCMD.EXE will create a backup and give it a name based on the date and time the backup was executed. This name is then displayed on the screen when the backup completes.

If a backup with that name already exists, an error will be displayed saying that it cannot create the file since it already exists. If you want to reuse that name, you will need to delete the backup and run the backup again. Here’s an example:

```
APPCMD.EXE delete backup IIS_Backup
APPCMD.EXE add backup IIS_Backup
```
Backups are great to have, but knowing how to restore from backup is just as important. Luckily, APPCMD.EXE makes this an easy task. You can list all available backups and restore a specific backup by running the following commands:

APPCMD.EXE list backup
APPCMD.EXE restore backup IIS_Backup

One of the greatest features of IIS 7.0 is the increased visibility into the server's state, such as its worker processes and requests. You can determine the state of your application pools or even which applications have been started or are currently stopped using the following commands:

APPCMD.EXE list apppools
APPCMD.EXE list apppools /state:started
APPCMD.EXE list apppools /state:stopped

You can see a list of all your currently running worker processes, the status of a specific worker process, and even all the worker processes associated with a specific application pool using the following commands:

APPCMD.EXE list wps
Appcmd list wp "2994"
Appcmd list wps /apppool.name:MyApplicationPool

You can also find out in realtime all the requests that are coming into your server. This can be further filtered by application pool, worker process, and site ID using the following commands:

APPCMD.EXE list requests
APPCMD.EXE list requests /wp.name:2994
APPCMD.EXE list requests /apppool.name:MyApplicationPool
APPCMD.EXE list requests /site.name:"Test Website"

The IIS 7.0 configuration is controlled by a set of hierarchical configuration files. You can use APPCMD.EXE to view and even update your configuration. This is not limited to your system configuration file but can be viewed at any level to see the net effect of the various configuration files to a particular path or URL within your server. The following examples show how to display your entire configuration file, how to filter it based on a specific section, and how to show the configuration of a specific path or URL.

APPCMD.EXE list config
APPCMD.EXE list config /section:defaultDocument
APPCMD.EXE list config "http://localhost/testWeb site/www" /section:asp

Setting the configuration is almost the same as viewing it. Instead of using the list verb, you use the set verb. You then need to add the parameter and value to set once you’ve specified the path or URL and the section to which you want this
parameter added. The following examples show how to set the enabled parameter of the defaultDocument section to true for the entire server and how to do it for a specific URL:

```
APPCMD.EXE set config /section:defaultDocument /enabled:true
APPCMD.EXE set config "Default Web Site/main/www" /section:defaultDocument /enabled:true
```

Note that the path or URL can be specified either as a fully qualified URL or as a path relative to a site name. For example, the URL http://localhost/TestWebsite/www can be specified to note the configuration of the www folder of the TestWebsite virtual directory. If you specify the path relative to a site name, such as “Default Web Site/main/www”, this would denote the main\www folder of the site called Default Web Site.

---

### IIS 7.0 Configuration Files

One of the biggest changes with IIS 7.0 is the use of configuration files instead of the IIS metabase for managing the server configuration. In fact, ASP.NET and IIS configuration settings are now combined into a single unified format. The configuration is physically divided into four different configuration files that are set up in a specific hierarchy. This hierarchical approach allows settings to be globally defined in XML-encoded text files and then allows subsequent virtual directories and folders within them to change those default settings using a localized configuration file (if the configuration is unlocked). The configuration is split between four different configuration files:

- **Machine.config** These settings apply to the whole server and are inherited by all .NET and IIS configuration files.
- **ApplicationHost.config** These settings are specific to IIS and inherit any settings from Machine.config. This file is stored by default at %systemroot%\system32\inetsrv.
- **Web.config (root-level)** These settings are shared by all ASP.NET applications on the server and inherit from both Machine.config and Application.config. This file is stored by default at %systemroot%\Microsoft.NET\Framework\versionNumber\CONFIG.
- **Web.config (application-level)** These settings are used to control configuration settings for a specific ASP.NET application. They inherit from all three configuration files above as well as any other web.config file that is above its hierarchy. This file is stored in the same folder as the ASP.NET application.
DELEGATED ADMINISTRATION

When it comes to IIS, one of the largest administrative overheads is managing the various configuration changes required by each of the applications you host on your server. It would be great if you could just permit developers to make some of their own customizations while still retaining control over the stability of your server. If that’s what you’ve been wishing for all these years, your wish has been granted in IIS 7.0. You can selectively lock or unlock different sections of the global configuration so that they can be overridden by a local web.config file. Furthermore, since administration is all configuration file–based, all the developer or Web site owner would need is access to upload the configuration files to their application space and the changes would be made available. You don’t have to worry about developers writing code and mucking around in your IIS metabase. Each site can have its own settings, while you still have overall control as to which options others can configure. You need to remember, however, that locking and unlocking the configuration of various features changes values only in the related configuration files. You will still need to make sure that the permissions (ACLs) of all your configuration files are set appropriately so that the root configuration files can’t be modified or replaced.

Delegation can be performed in two ways: You can lock or unlock features you want to delegate graphically through the IIS Manager or through the command prompt using APPCMD.EXE. Technically, there is a third way, which is to modify the configuration files—which are simply XML text files—in any text editor such as Notepad; but for the most part, you should stick to one of the two standard methods. Configuring feature delegation using the graphical method is more straightforward than mucking around with APPCMD.EXE. However, if you have large IIS server farms and need to make changes to multiple servers and multiple sites, it might make sense to script the changes using APPCMD.EXE. The following example shows how to lock and unlock the defaultDocument feature in the default configuration using APPCMD.EXE:

APPcmd.exe lock config /section:defaultDocument
APPcmd.exe unlock config /section:defaultDocument

You can also lock or unlock a feature for a specific Web site. For example, to lock and unlock the defaultDocument feature on a site called IT Homepage, you can run this:

APPcmd.exe lock config "IT Homepage/" /section:defaultDocument
APPcmd.exe unlock config "IT Homepage/" /section:defaultDocument

The biggest question you must have now is “How do I find out the section names?” The section names used as parameters to APPCMD.EXE are the names of the various features as they are used within the application.config XML file. It would be nice if you could use the more user-friendly version of these names, as displayed in the IIS Manager, but you can’t. The easiest way to get the names is to look them up in the “IIS Manager Feature to Configuration Mapping” article on IIS.NET (www.iis.net/default.aspx?tabid =2&subtabid=25&i=1032). You can also open the application.config XML file and look up the section name there, although you may have to do a bit of searching and scrolling to identify the right section name.
Hands-On Exercise: Delegating Features Using IIS Manager

One of the easiest ways to configure delegation is through the graphical interface of the IIS Manager. You can configure delegation at the server level, wherein the delegation is inherited by all sites on the server, or you can create custom site delegation rules specific to an individual Web site. To configure site-wide delegation, follow these steps:

1. Open IIS Manager.
2. Click your IIS server in the Connections pane.
3. Double-click Feature Delegation in the Home pane (Figure 6-11).
4. You will see a list of features you can configure for delegation (Figure 6-12).
5. To change the delegation, double-click the feature you want to change, and then click the desired delegation state in the Actions pane. Alternatively, you can right-click the feature and select the delegation state from the pop-up menu.

Figure 6-11. Selecting Feature Delegation in IIS Manager
If you want to create site-specific delegation rules, follow these steps:

1. Open IIS Manager.
2. Click your IIS server in the Connections pane.
3. Double-click Feature Delegation in the server’s Home pane.
4. Click Custom Web Site Delegation in the Actions pane, which will open the Custom Web Site Delegation page (Figure 6-13).
5. In the Sites drop-down list, select the site you want to configure.
6. Optionally, you can make changes to multiple Web sites at the same time. You can do this by clicking the Copy Delegation button and selecting the additional sites you want to modify in addition to the site you selected, as shown in Figure 6-14.
7. Configure your feature delegation states by selecting the feature and selecting the feature state.
Figure 6-13. Custom Web Site Delegation page

Figure 6-14. Selecting multiple sites for custom site delegation
SERVER AND APPLICATION HEALTH AND PERFORMANCE

If you’ve had to troubleshoot a Web application, you know how tricky it can get. IIS 7.0 provides more out-of-the-box tools and functionality to make the server more transparent for troubleshooting. It now includes a Runtime Status & Control API (RSCA), which allows tools and even WMI scripts to be developed to get into the deep inner workings of your server. This makes it possible to query the status of your sites, application pools, worker processes, and even currently executing requests. In addition, automatic failed request trace logging is available. Have you ever had sporadic application performance or availability issues? If so, automatic failed request trace logging may be a lifesaver. You can configure IIS 7.0 to look for certain error or performance degradation issues and then automatically begin tracing when those conditions occur. That way, even if the issue occurs off hours when you’re not directly monitoring the server, you can determine the cause.

Runtime Status & Control API

The purpose of the RSCA is to expose both the runtime and configuration data of the various IIS 7.0 objects to assist in its troubleshooting and monitoring. Not only can you interface with the API directly, Microsoft also includes a WMI provider so that you can write scripts to access RSCA. The IIS RSCA WMI provider is implemented in the new WebAdministration namespace. The following objects are exposed:

- ApplicationPool
- WorkerProcess
- AppDomain
- HttpRequest
- Site

These objects are also associated through associator classes, which establish relationships among these objects. For example, you can use associator classes to query the worker processes that are associated with an ApplicationPool object. The following is an example of a script that uses the WebAdministration WMI namespace to query information about any running worker processes:

'--- Connect to the WebAdministration provider
Set oWebAdmin = GetObject("winmgmts:root\WebAdministration")

Set oW3Processes = oWebAdmin.InstancesOf("WorkerProcess")

'--- Display all running worker processes
For each oProcess in oW3Processes
'--- Display the information for each process
WScript.Echo "Process ID: " & oProcess.PID
Next

**NOTE**  To take advantage of this WMI provider, you will need to install the IIS Management Scripts and Tools role service using Server Manager’s Add Role Services Wizard. This is not installed by default.

You can also interact with the RSCA through the IIS Manager. You can view a list of running worker processes by double-clicking Worker Processes on your server’s Home pane in IIS Manager. Not only can you see key information about each worker process, such as its current state and its CPU and memory utilization, you can also click the View Current Requests link in the Actions pane to view any requests going into that worker process. If your Web application has any performance or timeout issues, you can use RSCA to see what type of requests are coming into your server and affecting your application.

**Automatic Failed Request Tracing**

One of the most frustrating things with any application is troubleshooting errors that aren’t easily reproduced or those that occur at odd times during the day when you aren’t physically monitoring the server. This sometimes makes it difficult to get to the root cause of an issue, such as application unavailability or sudden loss of performance. What you need is a way to trace the error as it occurs without tracing everything that goes on with your server (which in and of itself would degrade server performance and consume a lot of disk space). You need to complete four steps to get automatic failed request tracing to work on your server:

1. Tracing must be installed as role services for the Web Server (IIS) role.
2. Verify that the FailedRequestTracingModule is defined in IIS Manager.
3. Enable Failed Request Tracing.
4. Configure failure definitions.

**Hands-On Exercise: Setting Up Automatic Failed Request Tracing**

In this exercise, we install IIS Tracing. We then enable failed request tracing and create failure definitions that the server will use to decide whether to create a trace due to a specific condition.

1. Open Server Manager.
2. In the Roles Summary section, click Web Server (IIS).
3. Click the Add Role Service link.
4. Select Tracing from the Role Services list, as shown in Figure 6-15. Then click Next and then Install to complete the installation.

5. Open IIS Manager.

6. Select your server in the Connections pane.

7. Double-click Modules from the Feature list and verify that FailedRequestTracing Module is listed (Figure 6-16).

8. In the Web Sites folder under your server in the Connections pane, select the Web site on which you want to enable failed request tracing.
9. In the Actions pane’s Configure section, click the Failed Request Tracing link.

10. In the Edit Web Site Failed Request Tracing Settings dialog box, check the Enable checkbox, specify the directory to use to store the log files and the maximum number of trace files to store, and then click OK (Figure 6-17).

11. In your Web Site Home pane, double-click the Failed Request Tracing Rules icon (Figure 6-18).

12. To create a new Failed Request Tracing Rule, click Add in the Actions pane.
Figure 6-17. Enabling failed request tracing

Figure 6-18. Selecting the Failed Request Tracing Rules on a site’s Home pane
13. Specify the content you want to trace (Figure 6-19), and then click Next.

14. In the next dialog box (Figure 6-20), specify the event severity, status (error) codes you want to monitor, and/or if you want to check for a timeout condition, enter the maximum number of seconds a request can take before it should be traced. Then click Next.

15. In the next dialog box, select the trace providers you want to use and the verbosity (Figure 6-21). Then click Finish. Refer to Table 6-3 for a description of available trace providers and Table 6-4 for verbosity levels.
Figure 6-20. Defining trace conditions

Figure 6-21. Selecting trace providers
In IIS 7.0, IIS and ASP.NET configurations have been unified into the web.config file. It is now possible to create a web.config file that can reside in your application directory and then be copied along with the application to new servers without any other configuration changes. This has the caveat that whatever configuration or application settings are being specified in the application or Web site’s web.config file are not locked through another configuration file higher up in the hierarchy. It is also completely possible to store the web.config file in a centralized location and then have Web sites and applications reference it. This way, changes can be made globally without your having to modify multiple files individually.

**XCOPY DEPLOYMENT**

<table>
<thead>
<tr>
<th>Trace Provider</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASP</td>
<td>Used for tracing start and completion of ASP requests</td>
</tr>
<tr>
<td>ASP.NET</td>
<td>Used for tracing transition into and out of managed code including .ASPX files</td>
</tr>
<tr>
<td>ISAPI Extension</td>
<td>Used for tracing transitions into and out of ISAPI processes</td>
</tr>
<tr>
<td>WWW Server</td>
<td>Used for tracing processes to IIS worker processes</td>
</tr>
</tbody>
</table>

**Table 6-3. Trace Providers**

<table>
<thead>
<tr>
<th>Verbosity Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>Information that provides context for the request activity</td>
</tr>
<tr>
<td>Critical Errors</td>
<td>Information about actions that cause a process to exit abruptly</td>
</tr>
<tr>
<td>Errors</td>
<td>Information about components that experience errors that prevent it from proceeding</td>
</tr>
<tr>
<td>Warnings</td>
<td>Information about components that experience an error but can still proceed</td>
</tr>
<tr>
<td>Information</td>
<td>General information about requests</td>
</tr>
</tbody>
</table>

**Table 6-4. Verbosity Levels**
CHAPTER SUMMARY

IIS 7.0 has been written from the ground up to become a more secure and feature-rich Web service platform. The added customization options provide added flexibility but also require more thought and planning prior to deployment. You will need to weigh functionality heavily over security and should limit the components to be installed only to those absolutely required for your applications. From a diagnostic and monitoring perspective, take advantage of automatic failed request tracing to assist in identifying the root cause of Web application issues without necessarily having to re-create the entire problem.

Although the new IIS Manager interface is more cleanly organized and easier to use than ever before, don’t discount the power of APPCMD.EXE and the command line. APPCMD.EXE can perform most of the tasks you can deploy in the full GUI and lends itself easily to scripting and automation. This, combined with the ability to perform Xcopy deployments of your applications, can significantly reduce your administrative burdens.

Use feature delegation whenever possible as this can shift many administrative configuration tasks back to the developer or site owner. However, exercise caution when unlocking certain configuration settings since you don’t want to open everything up and have unwanted configuration settings applied to Web sites that could adversely affect the performance and stability of your server.
CHAPTER 7

Resource Management and Performance Monitoring
The purpose of nearly every server is to provide some form of centralized service for its users. Servers provide a cost-effective means of sharing resources, and, as such, they are critical pieces of infrastructure in almost every organization that uses them. Although they sit away from view and quietly perform their services, you can appreciate their importance simply by looking at users’ reactions when one or more servers suddenly becomes unavailable or responds very slowly. As Windows administrators, it is our job not only to ensure that the servers reliable, but also to extract the most performance out of our systems, in addition to tracking capacity and predicting growth. The only way to accomplish these objectives successfully is to perform reliability and performance monitoring on servers on a regular basis. This is especially true for application servers that increase in use over time.

Baseline performance metrics must be assessed at regular intervals so that when performance issues do arise, you can quickly and easily compare the current performance profile with previously recorded profiles to determine what, if anything, is happening out of the ordinary. Being able to monitor your system performance is one thing, but being able to manage that performance effectively is something else. Windows Server 2008 includes the Windows System Resource Manager (WSRM), which had its start in Windows Server 2003 Enterprise and Datacenter editions. This tool can be used to tune your server’s performance by allowing you to specify exactly where CPU and memory resources are allocated. To monitor system performance, Windows Server 2008 includes a Reliability and Performance Monitor, which is an enhanced version of the Performance Monitor available in previous Windows versions.

DATA IS GOOD!

If you work with any reliability and performance metrics-gathering tools, you will find that it is far too easy to gather what seems to be too much data. Sometimes the data you gather may not be what you want. For example, if you work in an environment that has experienced immense and sudden growth, you may find that you no longer have time to follow your own best practices, so that over time and when you finally collect performance metrics, you might determine that your servers are just as overworked as you are. Many small to mid-sized organizations (and, unfortunately, even some large ones) use metrics gathering as an afterthought, and these tools are brought out only when an administrator is reacting to a serious issue—such as when an application server performs at an unacceptable level.

For the most part, if you gather and trend your data proactively, you can find possible trouble spots well before they become issues or outages. In fact, reliability and performance-metrics gathering is the cornerstone of any proactive systems management strategy. If you’ve inherited a poorly managed server infrastructure, you may not like what the data is showing you, but at least you have the information you need to make some good decisions about where to focus your attention. In fact, such information is the most effective way to justify to upper management why you need to spend money on equipment and resources. Rather than simply providing a dollar amount, you can
supplement your proposal with data showing how your server capacity is shrinking and the potential risks involved if this issue isn’t resolved. In addition to all this, as we move more toward virtualization, it is important that we have good performance metrics to guide us in which servers and how many servers can be hosted on a single host node.

**WINDOWS SYSTEM RESOURCE MANAGER**

Windows System Resource Manager was part of Windows Server 2003 Enterprise and Datacenter editions and was available on a separate CD. In Windows Server 2008, WSRM is part of Windows Server 2008 Enterprise and Datacenter editions. The major difference between the old and new versions is that WSRM can now be installed directly through the Server Manager interface rather than having to run it from a separate disc.

WSRM acts as a kind of “resource police” among the various processes on your system. It allows administrators to specify constraints for each process, such as how much CPU and memory each process is allowed to use, and then enforces the constraints so that one application or process can use only the amount of resources that have been allocated to it. Setting constraints may cause that particular process to run slower once it reaches its constraints, but at least it won’t allow that process to overwhelm the server and cause problems with other processes.

**WSRM Architecture**

WSRM is composed of or interacts with nine distinct primary components, including a management interface, information stores, schedulers, and managers. The following table lists those components and a description of each.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSRM console</td>
<td>Graphical interface used to manage and monitor WSRM.</td>
</tr>
<tr>
<td>Distributed Component</td>
<td>Remote APIs used to communicate between the client and the WSRM service.</td>
</tr>
<tr>
<td>Object Model (DCOM) interface</td>
<td></td>
</tr>
<tr>
<td>WSRM service</td>
<td>Main service that performs resource management. Its job is to track processes and compare them against currently defined matching rules and policies. If a process exceeds its current resource allocation, it will attempt to control the process to comply with the resource allocation.</td>
</tr>
<tr>
<td>Accounting database</td>
<td>Stores information about managed processes on a per-process basis.</td>
</tr>
<tr>
<td>Policy store</td>
<td>Stores all the policies and resource matching criteria defined in WSRM.</td>
</tr>
<tr>
<td>WSRM settings</td>
<td>Stores the current management settings.</td>
</tr>
</tbody>
</table>
Managed vs. Unmanaged Processes

WSRM categorizes every process as either *managed* or *unmanaged*. Managed processes are all processes except those explicitly not controlled by WSRM. This includes any process not matching process-filtering criteria or excluded processes. Trying to restrict system processes using WSRM could result in very detrimental effects on your server. Due to this, WSRM includes a set of system-defined exclusions. This list cannot be modified and contains processes deemed by Microsoft as being critical to the running of the core operating system; these files should not be tampered with. These processes include, but are not limited to, the following:

- Csrss.exe
- Dumprep.exe
- Lsass.exe
- Msdtc.exe
- Services.exe
- Smss.exe
- Spoolsv.exe
- Taskmgr.exe
- Winlogon.exe
- Wmiprvse.exe

WSRM Service

The core of WSRM is the WSRM service. As the workhorse of WSRM, it continually polls your server for processes and compares them against existing matching rules and policies. It is also in charge of monitoring the consumption of CPU and memory resources so that it can control processes that exceed set thresholds. Whenever a new process is discovered, the WSRM service compares the process against its list of included or excluded processes using a priority-matching algorithm that follows the priority-matching criteria you specified in your policy. If no match is found, the process is automatically placed into the default group and is controlled by the default policy. The default group is allocated resources unaccounted for by the managing policy that are shared equally among all default group processes. If the WSRM service does find a match in the process list, the new process is grouped with other processes matching the same criteria, and

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calendar</td>
<td>Stores all calendar-related events.</td>
</tr>
<tr>
<td>Memory Manager</td>
<td>Manages memory allocated to managed processes.</td>
</tr>
<tr>
<td>Kernel Scheduler</td>
<td>Controls how processes are scheduled to run on the processor based on WSRM policies.</td>
</tr>
</tbody>
</table>

**Component Description**
- Calendar: Stores all calendar-related events.
- Memory Manager: Manages memory allocated to managed processes.
- Kernel Scheduler: Controls how processes are scheduled to run on the processor based on WSRM policies.
is subject to the utilization rules defined in the policy. The list of running processes is also re-examined if changes are made either to the process matching criteria or to the active allocation policy. Whenever a process exceeds its target resource allocation, it is subjected to a dynamic process priority-management algorithm that shuffles resources between processes, again based on your defined process priorities.

**Hands-On Exercise: Installing WSRM**

WSRM is not a server role. Instead, it is simply an optional feature that can be installed on your Windows Server 2008 server. To install WSRM, follow these steps:

1. Run Server Manager.
2. Click the Add Features link to open the Add Features Wizard.
3. Select Windows System Resource Manager (Figure 7-1). Click Add Required Features (Figure 7-2), if prompted, to install Windows Internal Database, which is required for WSRM to run. Click Next.

![Figure 7-1. Selecting Windows System Resource Manager](image-url)
4. Confirm the Installation Options, and then click Install.

5. Click Close when the installation completes.

6. Reboot the server or start the Windows System Resource Manager service, if it hasn’t been started yet. The WSRM console can be access by choosing Start | Administrative Tools | Windows System Resource Manager.

The WSRM Management Interface

Windows Server 2008 is all about task-oriented interfaces, so it’s not surprising that when you open the WSRM management interface, you see a three-pane interface similar to that of the IIS 7.0 management interface, as shown in Figure 7-3. The left pane is the Navigation pane used to access the various components of WSRM. The center pane, otherwise known as the Home pane, is the primary interface where configuration information is displayed and can be manipulated. The right pane is the context-sensitive Actions pane.

When you first connect to a WSRM-enabled server, you see a summary page that shows the state of WSRM (running or stopped) as well quick links to the various components that make up WSRM. For example, it will show whether the Calendar, Notification, or Accounting components are enabled, and provide links to make changes to any of these settings.

This chapter covers each of the major configuration categories in WSRM: Resource Allocation Policies, Process Matching Criteria, Conditions, Calendar, Resource Monitor, and Accounting. If you haven’t noticed already, WSRM cannot only be used to set resource allocation policies, it can also be used to monitor resource utilization. If you click
Resource Monitor in the navigation pane, you will see a familiar interface in the home pane. In fact, Resource Monitor is for the most part nothing more than an integrated version of the old Performance Monitor used in previous version of Windows—with a few enhancements.

**Process Matching Criteria**

Process matching criteria are important because they ultimately define to which processes a particular policy will apply. Two process matching criteria are defined automatically by the system: Residual and IISAppPool. Residual matches all processes and should be used if you want your policy to apply to every process running on your system. IISAppPool matches all IIS application pool worker processes. This is useful if you have IIS installed and want to create policies around how many resources IIS worker processes should be able to have. The criteria can be based on the path to the file or command line, or on users and groups, and you can control what to include or exclude.
For example, you can create a rule to apply to any account belonging to the Users local group but exclude the local Administrator account. The process matching criteria are useless by themselves. You can think of them as process filters. You can create as many of them as you want, but until you actually apply them to a policy, they won’t do anything.

For processes that are matched using included files or command lines, WSRM first attempts to match based on the process name. If that fails, it compares against the fully qualified path and filenames. Lastly, it compares against the full process command line. If a match is found in the included files, WSRM then checks the excluded files list. This is necessary since the process may have matched due to a wildcard filter but may have been explicitly excluded by the administrator. Then, against the excluded files, it follows the same general matching procedure entries used for the included files entries.

For processes that are matched using users and groups, WSRM compares the account used to create the process against the list of users and groups. This is first done using an exact user account match; if that is not successful, WSRM compares the user account against the membership of all the groups specified. If a match is found, a comparison against the excluded user and groups list is performed to filter out any process that should be excluded based on those rules.

Any filter criteria that includes both file and command-line matching criteria and user and groups matching criteria must evaluate to true in both cases to be included. For example, if you create a process matching criteria to look for the process MyService.exe and also specify that the user must be BUILTIN\Administrator, only MyService.exe processes initiated by BUILTIN\Administrator will be included. If any other user launches MyService.exe, it is not included and is placed in the default group.

---

**NOTE**  Criteria names cannot start with a hyphen (–), and cannot contain spaces or any of the following characters: \\ / * ? | : < > " , ;

**Hands-On Exercise: Creating a Process Matching Criterion**

In this exercise, we will create a process matching criterion to match Notepad.exe when executed by an account belonging to the local Users group.


2. You will be prompted to select the server to administer—either the local or a remote node. Select This Computer, and then click Connect.


4. In the Description field, enter **AllNotepad** as the Criteria Name and **This will match any notepad process initiated by any member of the local Users group.** (Figure 7-4).
5. Click the Add button to add a new rule.
6. In the Included Files Or Command Lines section, choose Application from the drop-down menu and click the Select button.
7. Browse to C:\Windows\System32 and select notepad.exe. Then click Open. This will add C:\Windows\System32\notepad.exe to your list of included files, as shown in Figure 7-5.
8. Click the Users Or Groups tab.
9. Click the Add button next to the Included Users And Groups list box.
10. Type in Users in the Select Users Or Groups dialog box and then click OK. This will add BUILTIN\Users to the list of Included Users And Groups, as shown in Figure 7-6.
11. Click OK on the Add Rule dialog box to save the settings.
12. Click OK on the New Process Matching Criteria dialog box to save this criterion (Figure 7-7).

**Resource Allocation Policies**

Once you’ve created your process matching criteria, you can create resource allocation policies. These policies dictate how processes get their share of resources. Each policy contains a list of one or more process matching criteria that in turn have a set of resource constraints such as CPU allocation, affinity, and memory limits. Four resource allocation policies are defined out of the box. By default, the Equal_Per_Process policy is set, which gives each process an equal share of CPU time. Only one resource allocation policy can be active at any given time. This active policy is called the *managing* policy. In the WSRM
console, this policy is clearly identified by the string *Manage* that appears next to the policy name. Following are the resource allocation policies:

- **Equal_Per_Process**  
  Each running process gets its equal share of CPU cycles (default).

- **Equal_Per_User**  
  Each user’s processes get an equal share of CPU cycles.

- **Equal_Per_IISAppPool**  
  Each IIS application pool’s worker process gets an equal share of CPU cycles.

- **Equal_Per_Session**  
  Each user-session’s processes get an equal share of CPU cycles (relates to Terminal Sessions).
You can control two types of resource allocation using WSRM: CPU and memory. The general procedure for creating a policy is that you assign it a process matching criteria that will tell the policy to which processes it applies, and then allocate a percentage of your overall resources, whether CPU or memory, to those processes. Each policy can have multiple associated resource matching criteria and each criterion can then have its own set of resource allocation parameters. You can allocate up to 99 percent of the CPU to each process matching criterion, but the total cannot exceed 100 percent. The remaining minimal 1 percent is reserved for use by processes placed in the default group.

NOTE WSRM will not enforce CPU allocation rules until resources begin to get used up. Until then, all processes get as much CPU as they want. For example, if your server is running only 50 percent CPU utilization, none of your CPU allocation policies will take effect even if you’ve specified only that a process should have 20 percent of your CPU resources. When your server starts to get closer to its maximum utilization, the processes are constrained based on whatever policy you’ve defined.
Chapter 7: Resource Management and Performance Monitoring

The amount of processing time each process within a particular matching group gets is defined by your selected management rule:

**Standard**  Default setting. The operating system, not WSRM, is in charge of distributing CPU processing time to each process.

**Equal per process**  WSRM will make every process within a group use up the same amount of CPU cycles. All processes within a particular group are, however, constrained by the overall total percentage you specified in your policy. For example, if you specify 25 percent CPU processing for a particular process matching criterion, all processes that fall within that criterion must equally share the 25 percent CPU. In other words, if you had five processes matching that criterion, each process would get only 5 percent of the CPU cycles.

**Equal per user**  Similar to Equal per process except it groups processes by user who initiated them. This is useful in a terminal server environment.

For multiprocessor systems, it gets even more complicated. The percentage reflects the percentage compared to your overall CPU bandwidth constraints. For example, if you have four processors and you specify 25 percent to divide between your managed processes, the 25 percent of your total CPU bandwidth that you then specified means that instead of having 100 percent of one CPU for your process, you have only 50 percent of one CPU (25 percent of two CPUs = 50 percent of one CPU). What this all really means at the end of the day is that you have to minimize the amount of unmanaged processes to make WSRM’s resource management effective. Outside of the system-defined exclusion list, you should avoid excluding processes as much as possible. The amount of memory you allocate to each of your matching criterion is limited only by the amount of memory you have on your system. You can create soft or hard limits. A soft limit is implemented in the form of an event log entry that’s generated when a process matching your criteria exceeds the maximum memory allocated to it. A hard limit stops the application completely when it has exceeded its memory allocation. This is useful in preventing a runaway process from completely using up all your server resources. For example, you can apply this to your IIS worker processes. This way, if a poorly written Web application suddenly wants to hog all your memory, WSRM will automatically stop it for you, keeping all your other processes up and running. Unlike the CPU resource limits, the memory limit you specify applies to each process that falls under that group and isn’t shared between them. For example, if you specify 20MB as the maximum limit for a given process matching criterion, then each process that matches that criterion gets 20MB.
Limits can be set on *working set* or *committed memory*. Working set memory refers to the amount of memory used by the process during its runtime. Once the upper limit for working set memory has been reached, the memory manager begins swapping out the memory pages. This can reduce the performance of the application somewhat but will not induce any out-of-memory errors. Committed memory, on the other hand, is used to watch for errant processes or memory leaks. By setting, a reasonable threshold on your processes, you can force those processes to stop when they reach the limit or log an entry in your event log for later troubleshooting. WSRM isn’t actually involved in the memory allocation. This is the job of the memory manager. WSRM’s job is to monitor utilization, and if it exceeds that utilization, inform the memory manager so it can adjust or even deny additional memory from being allocated to that process.

You can also specify additional advanced options. On a multiprocessor system, you can specify exactly which processor or processors each process is allowed to use. You can also optionally suballocate processor resources. For example, if you had four processors on your server and you allocate two processors for a specific process matching criterion, you can then use suballocation to specify how those two processors should be split up between the processes. This creates a parent/child relationship between an allocation and its suballocation. You’re not limited to one level of parent/child relationship. A child can have its own children, so you can have multi-level allocation relationships. The allocation begins at the lowest level, and any available resources are then made available to the parent, until finally any remaining resources are made available to the default group. This is referred to as a *priority-order chain*.

**Hands-On Exercise: Creating a CPU Allocation Policy**

In this simple example, we will use the AllNotepad process matching criterion to limit any notepad.exe processes to use only 10 percent CPU and up to 10MB of memory at any given time.

1. Open Windows System Resource Manager and connect to your server.
2. Right-click the Resource Allocation Policies in the navigation pane, and then select New Resource Allocation Policy from the pop-up menu.
3. Enter **Limit_Notepad** in the Policy Name field and **Limit notepad processes to 10% CPU and 10 MB of memory.** in the Description field, as shown in Figure 7-8.
4. Click the Add button.

**NOTE** Policy names, like criteria names, cannot start with a hyphen (–); nor can they contain spaces or any of the following characters: \ / ? * : | < > " , ;

4. Click the Add button.
5. Select AllNotepad from the Process Matching Criteria drop-down menu and enter the value 10 for the Percentage of Processor Allocated for This Resource, as shown in Figure 7-9.

6. Click the Memory tab and check the Use Maximum Committed Memory for Each Process checkbox.

7. Enter the value 10 in the Maximum Committed Memory Limit Per Process field.

8. If memory is surpassed, select Log an Event Log Message from the drop-down list (Figure 7-10). Click OK to save the allocation settings.

9. You have created a completed resource allocation policy, as shown in Figure 7-11. Click OK to save this new policy.
The Calendar is the WSRM scheduling module. It allows you to specify when a specific resource allocation policy becomes active. The Calendar is made up of calendar events and schedules. Schedules are periods of time within a 24-hour clock when a policy is active. For example, you may want a specific policy to be active during business hours and a different one to be active off hours. Calendars specify a start date and time as well as an end date and time where a particular policy will become active. They can also be used in conjunction with schedules to set a date range when a schedule will take effect. For example, you may want the business hours and off hours schedule to happen during the month of January. Following are the types of calendar events:

- **One Time**  Create a one-time event when a policy is active. This requires a start date and time and an end date and time.
- **Recurring Event**  Like a recurring meeting in Outlook, you can use this to schedule recurring calendar events for your policies.
- **Schedule**  Use to activate different resource allocation policies over the course of a 24-hour period.
Hands-On Exercise: Creating a Calendar Event

Continuing on from our previous exercises, we will create a one-time calendar event to make our Limit_Notepad resource allocation policy active for a specified two-week period from June 15, 2007, at 6:00 A.M. to June 30, 2007, at 6:00 P.M.

1. Open Windows System Resource Manager.
2. Verify that the Calendar is enabled (it should say Calendar {Enabled} in the WSRM navigation pane).
3. In the navigation pane, expand the Calendar node.
4. Right-click Calendar Events and select New One Time Event.
5. Enter or select the following information, as shown in Figure 7-12:
   - Event Name: NotepadSchedule
   - Description: Enable Limit_Notepad
   - Policy Name: Limit_Notepad
   - Start date and time: 6/15/2007 6:00AM
   - End date and time: 6/30/2007 6:00PM
6. Click OK to save the new calendar event.
Figure 7-11. Completed new resource allocation policy

Figure 7-12. Creating a new calendar event
Hands-On Exercise: Creating a New Schedule

In this exercise, we will schedule the Limit_Notepad policy to be active from 5:00 A.M. to 9:00 A.M. and from 6:00 P.M. to 10:00 P.M.

1. Open Windows System Resource Manager.
2. Expand the Calendar node.
3. Right-click Schedule and select New Schedule.
4. Enter Notepad_Schedule in the Schedule Name field and My custom notepad schedule in the Description field.
5. Double-click anywhere in the orange schedule area.
6. In the Add Schedule Item dialog box, select Limit_Notepad under the Policy drop-down menu and select 5:00 AM and 9:00 AM as the start and end times, respectively; then click OK (Figure 7-13).
7. Double-click anywhere in the orange schedule area and follow the same procedure outlined in step 6, except this time select 6:00 PM and 10:00 PM as the respective start and end times.
8. Your new schedule will now look like Figure 7-14. Click OK to save the new schedule.

Accounting

The Accounting component of WSRM is used as a central accounting database to view records related to the behavior of managed processes. Accounting is disabled by default in WSRM. To enable it, right-click the Accounting node in the navigation pane and select Enable. By default, the Accounting database is locally stored on the WSRM server. You can set this to a different WSRM server if you want to centralize your accounting data, or you can specify a SQL Server instance to hold all your account data. Local WSRM accounting is the fastest, but if you have many WSRM-enabled servers, you may want to consider having the account data redirected to any of the two other options. To change your account database location, click the Accounting node in the navigation pane, and in the Actions pane click Set DB Server. This will open the Set Accounting Database dialog box (Figure 7-15), where you can specify the alternative database location.

**NOTE** Accounting can add significantly to the resources used by WSRM and can adversely affect the performance of your server. You should consider enabling accounting only when troubleshooting or testing your policies.

Every 10 minutes, the accounting information is updated and can be viewed in the WSRM console on the Accounting page. The default view is a simple dump of all the data.
You can create a reusable schedule to manage different policies and calendar events. To make better sense of the view, you can adjust the output by applying various filters to the data. Furthermore, these filter views can be saved and loaded for later, so if you have a complex filter you
want to use regularly, all you need to do is define that filter once and then click the Save View button in the Configure Accounting View Filter dialog box, as shown in Figure 7-16.
The following options are available for filtering the view:

- **Scope Filter**  
  Specifies the date range for the data you want displayed.

- **Filter Before Grouping**  
  Builds a specific filter for accounting items before they are grouped together.
Chapter 7: Resource Management and Performance Monitoring

- **Group Items**  Allows you to group the output by Process Name, Domain, User, Policy Name, Process Matching Criteria, Program Path, and Command Line.

- **Filter After Grouping**  If Group Items has been defined, an additional filter can be applied to items after they have been grouped using this option.

- **Specify Columns**  With two dozen pieces of information captured for each item in the accounting data such as process name, thread count, CPU time, and more. This allows you to specify exactly which columns you are interested in viewing.

- **Sort Items**  Lets you sort the data output by using this option.

---

**Hands-On Exercise: Archiving Accounting Information**

If you leave accounting enabled for a while, you will eventually see a list of processes that have been recorded by the Accounting component. If you want to archive this data to be reviewed later, follow these steps:

1. Open Windows System Resource Manager.
2. Right-click Accounting and choose Archive or Delete Information.
3. Specify start and end dates for the data in which you are interested.
4. Check the Archive Data checkbox.
5. Browse to select the location where the Archive will be stored.
6. Select the file format in which you want the archive to be saved. For now, leave it as the default (Comma Delimited Text).
7. The Archive or Delete Accounting Information dialog box should look like Figure 7-17. Click OK to save the archive.

---

**Conditions**

The Conditions node in the navigation pane contains a handful of predefined conditions that can be used to trigger a switch in a policy. For example, when a new processor is detected or if the number of processors is greater than a certain number, you can tell WSRM to switch to a different policy. You might find this useful if you want the policy to change when a node that is part of a Microsoft Cluster Service becomes unavailable, for example. You could have a policy that changes the priority of your processes automatically when a cluster node suddenly goes down and then automatically reverts back to your normal policy when that node comes back online.
Resource Monitor

Resource Monitor is covered briefly here, since if you’ve worked with Performance Monitor in previous Windows versions, you already know how to use it. When you click the Resource Monitor node in the WSRM navigation pane, you see a familiar graph interface (Figure 7-18). The x-axis represents elapsed time and the y-axis represents the possible values retrieved from each data source. This can be used for general monitoring of your server resources; but as you will see later, the new Reliability and Performance Monitor is a much more enhanced version of Resource Monitor.

Hands-On Exercise: Using Resource Monitor to Track CPU and Memory Usage

In this exercise, we configure Resource Monitor to track total CPU utilization and a number of Memory Usage statistics.

1. Open Windows System Resource Manager.
2. Click Resource Monitor.
3. Remove any currently displayed counters by clicking each counter and then clicking the red X icon until the graph is blank.
4. Click the green plus (+) icon at the top of the Resource Monitor page.
5. Under the Available Counters list, click the plus (+) sign next to Processor, and then select % Processor Time.
6. Select Total from the Instances of Selected Object list box, and then click the Add button.
7. Scroll up the Available Counters list and click the plus (+) sign next to Memory. Then select Available MBytes and click Add.

8. Under the same Memory counter, select Pages/sec; then click Add.

9. Your Add Counters dialog box should now look like Figure 7-19. Click OK.

10. Resource Monitor will show a graph of your data over time with a refresh interval of 1 second (Figure 7-20).
RELIABILITY AND PERFORMANCE MONITOR

So far, we’ve spent some time getting to know how to control our server resources using WSRM. Just as important is the ability to track server performance, which serves two purposes: for gathering performance metrics for either a baseline data set or for troubleshooting, and for capacity planning. As you know, Resource Monitor can perform some of the basic performance monitoring you want on a server. The Reliability and Performance Monitor is basically an extension of the Performance Monitor that was available in previous Windows versions. This tool still allows you to gather performance metrics, but it also has the ability to track server reliability and stability statistics. When you launch the Reliability and Performance Monitor (by choosing Start | Administrative Tools | Reliability and Performance Monitor), you see a Resource Overview (Figure 7-21), a summary view of your server’s major performance metrics. It displays in realtime the...
CPU utilization, memory hard fault statistics, as well as disk and network activity and utilization. You can drill down further to get more specifics by double-clicking the desired category under the graphs. For example, to find out what processes are taking up CPU cycles, double-click CPU and you will see a process list similar to what you would see in Task Manager (Figure 7-22).

As with Performance Monitor, the Resource Monitor in WSRM lets you add performance counters and track them as you normally would, except in the Reliability and Performance Monitor, the selected counters can also be used to create data collector sets (discussed later in the chapter). The Reliability and Performance Monitor is based on stability statistics. It tracks changes to your server and unexpected errors, which are translated into a stability index that you can use to gauge your server’s overall reliability. In short, the more stable your server, the more reliable it is.

As always, local administrators have access to perform any kind of performance-gathering function; however, two additional built-in groups can be used to grant non-administrators access to performance data. The Performance Log Users group should
Chapter 7: Resource Management and Performance Monitoring

contain user accounts that may schedule logging of performance counters, enable trace providers, and collect event traces both locally and remotely. Performance Monitor Users group members can also view performance counters locally or remotely. This group is useful if you want to allow nonserver administrators such as application developers to view server performance data remotely without granting them full administrative access to your servers.

TIP If you run Reliability and Performance Monitor locally on the server you want to monitor, you need to account for the extra processing and memory resources used by the tool when looking at your overall data. In many cases, it may make more sense to record performance metrics remotely from a different server or workstation so that it minimizes the possibility of skewing the gathered data. The caveat is that if you are tracking network performance, you must also take into account all the extra traffic being generated by running it remotely; so, for the case of network utilization tracking, it would make sense to log that locally on the server instead.
Data Collector Sets

A data collector is a component used to gather performance data about your server. For example, a processor data collector gathers information about the processor, such as its utilization. One or more data collectors can be grouped together to form a data collector set. Data collector sets can be used to define groups of data points for which you are interested in gathering data. You can run these data-gathering sessions on an ad hoc or scheduled basis and then view the data as reports through the Reliability and Performance Monitor console. This eases the task of gathering performance data. For example, you can create a data collector set to encompass a number of standard performance metrics such as CPU and memory utilization, as well as a few Terminal Services–related counters such as active and inactive sessions. You can then schedule this data collector set to run from 6:00 p.m. to 6:00 a.m. so that you can track the performance and utilization of the terminal server during nonbusiness hours. This is an overly simplified example, but I’m sure you get the idea.

Data collector sets can be created in one of three ways: via Performance Monitor, via template, and manually. Creating a data collector set using Performance Monitor involves adding all your counter objects to Performance Monitor and then using that list of counters to create your data collector set. Windows provides a number of out-of-box templates you can use as a starting point for creating data collector sets. You can also create your own templates from existing data collector sets that can be imported and used as a template for creating new data collector sets. You can also manually create a data collector set and pull data—event trace data and system configuration information—from performance counters. Data collector sets can also be used to monitor system performance and generate alerts when certain thresholds are reached.

Hands-On Exercise: Creating a Data Collector Set from the Performance Monitor

In this example, we create a data collector set by monitoring a few key performance metrics about our server, which we will call our Baseline Performance Metric.

2. Expand the Monitoring Tools node in the navigation pane.
4. By default, your % Processor Time is automatically added and is already monitoring your system.
5. Click the green plus (+) icon above the graph to add counters.
6. In the list of Available Counters, add the following counters by expanding the appropriate category and selecting the counter. Click the Add button when you’re done.
   Memory: % Committed Bytes In Use
   Memory: Page Faults/sec
Network Interface: **Bytes Received/sec**  
Network Interface: **Bytes Sent/sec**  
Server: **Logon/sec**  
Server: **Server Sessions**

The Add Counters dialog box should now look like Figure 7-23.

7. Click OK to add the counters to Performance Monitor.
8. Right-click Performance Monitor and select **New | Data Collector Set**.
9. Enter **Baseline Performance Metrics** as the Data Collector Set name, and then click Next.
10. Data collected from the data collector saved is typically stored in `%systemdrive%\perflogs\<Name of Data Collector Set>`. Either leave it as it is with the default path or browse to the path where you want the data to be saved; then click Next.

![Figure 7-23. Adding counters](image-url)
11. Select Save and Close in the Create New Data Collector Set dialog box, and then click Finish (Figure 7-24).

12. Verify that the data collector set has been created by expanding the Data Collector Sets, and then User Defined. Then make sure that the Baseline Performance Metric data collector set appears.

13. Right-click the Baseline Performance Metric Data Collector Set in the navigation pane and select Properties.

14. Click the Directory tab.

15. In the Subdirectory name format, enter mmyydd\-NNNN. This dynamically creates a subdirectory every time this data collector set is executed with the format of the current date followed by a dash and then a serial number (see Figure 7-25).

16. Click OK to save and close.
Hands-On Exercise: Scheduling a Data Collector Set to Run Daily

If you want to gather performance metrics regularly without manually initiating the data collection to occur, you can schedule a data collector set to run at a specific date and time. In this exercise, we schedule the Baseline Performance Metric data collector set we created in the previous exercise to run from Monday to Friday from 9:00 A.M. to 6:00 P.M.

1. Open Reliability and Performance Monitor.
2. Expand Data Collector Sets, and then User Defined.
3. Right-click Baseline Performance Metric and select Properties.
4. Click the Schedule tab.
5. Click the Add button.
6. Set the Start Launch Time to 9:00:00 AM and uncheck Saturday and Sunday; then click OK (Figure 7-26).
7. Click the Stop Condition tab.
8. Check the Overall Duration checkbox and enter 9 Hours for the duration (9:00 A.M. to 6:00 P.M. is 9 hours). Then click OK (Figure 7-27).
Reliability Monitor

Reliability Monitor is a neat way to get an overall sense of how your server is doing from a health and stability perspective. Every day the system is compared to a list of stability reports. It counts the number of software installs and uninstalls that have been performed in the last 24 hours. It looks for application, hardware, Windows, and miscellaneous failures that may have occurred as well. Using an algorithm, this information is then translated into a stability index ranging from 1 to 10, where 10 is the most stable. This index is displayed on a System Stability Chart so you can trend your server's reliability over time (Figure 7-28).

If any recent changes or failures resulted in a lower index, you can find out more information by expanding the relevant category in the System Stability Report section under the chart. Each of these categories contains important information you can use for troubleshooting your server. For example, for a Windows failure, it will indicate the failure type (boot failure or OS crash), OS version, service pack level, failure details including stop and reason codes, and of course the date and time when the failure occurred.
You should note the following about Reliability Monitor, including how it comes up with its stability index:

▼ Recent failures are weighted more heavily that past failures.
■ The system automatically excludes any days in which the server is shut off or is in a sleep state.
■ Dotted lines are used in the chart whenever data is insufficient to calculate a steady stability index. This may happen if the stability is constantly fluctuating or if the server has been recently set up.
▲ Any significant change in the system time is noted on the graph to denote a system time adjustment. It will indicate the old time, the new time, and the date the change occurred (based on the new time).
Reports

Reports are the main reason why you want to use data collector sets when collecting performance metrics about your server. When a data collector set is running, it stores all the data points it gathers into a log file in the directory you specify. When you stop data collection, the data from this capture is made available to you in the form of a report. Reports can display either a summary performance report or a Performance Monitor screen with data from the data file rather than from realtime counters. If, for example, you had the System Performance data collector set running for 12 hours, the resulting Performance Monitor report will display the graph of all the data points collected for this data collector set over the course of those 12 hours. You can then quickly see trends on your servers and use it to track spikes in your system that could indicate capacity or performance issues. For example, if you notice that between 12:00 p.m. and 1:00 p.m. a huge spike occurs in processor utilization, you may want to investigate whether any scheduled tasks are running at that time that could be causing this spike.
Reports are accessed through the Reliability and Performance Monitor. Expand the Reports node in the navigation pane and you will see two categories: User Defined and System. System reports contain reports generated by system-generated data collector sets. Currently, these are LAN Diagnostics, System Diagnostics, and System Performance. The system-generated data collector sets are configured so that their output directory is dynamic and follows the date and numeric sequence when the collector set was executed. Every time you run one of these collector sets, a new report is generated with the date and sequence used as the directory name, as shown in Figure 7-29.

By default, when you open a User Defined Report, you will see only a Performance Monitor view of your recorded data. What you probably want is an actual report that summarizes the gathered data into something that can be presented to management. What I left out in the example for creating a user-defined data collector set is that if you want to be able to generate pretty reports like you see in Figure 7-29 for the system-generated data collector sets, you will need to modify the Data Manager properties of your collector set to enable data management and report generation.

![Figure 7-29. Viewing a report of a System Diagnostics Report](image-url)
Hands-On Exercise: Configuring a Data Collector Set to Generate Reports

In this exercise we configure the Baseline Performance Metrics data collector set that we created in a previous example to allow it to generate a report view.

1. Open Reliability and Performance Monitor.
2. Expand Data Collector Sets and then User Defined.
3. Right-click Baseline Performance Metrics and select Data Manager.
4. On the Data Manager tab, check the Enable Data Management and Report Generation checkbox and click OK (Figure 7-30).

Figure 7-30. Enabling data management and report generation
5. To test the reporting functionality, right-click Baseline Performance Metrics and select Run Start.

6. Wait a few seconds or minutes so it has time to gather some data, and then right-click Baseline Performance Metrics again and select Stop.

7. Expand Reports and then User Defined.

8. Expand Baseline Performance Metrics and select the newly generated report. The report should look similar to Figure 7-31.

9. To switch to the Performance Monitor view, select the report System Monitor Log.blg.

Figure 7-31. Viewing a user-defined data collector set report
CHAPTER SUMMARY

Monitoring your server’s performance effectively and then tuning it so that you can maximize its full potential is really what resource management and performance monitoring are all about. You need to collect enough statistical data about the performance of your server during its typical course of operations to be able to detect changes in capacity and performance over time. This data can be used to justify acquiring additional servers or upgrading your existing ones. It can also help you identify potential bottlenecks in your system and allow you to troubleshoot server and application issues more effectively.

Take advantage of your new ability to monitor your server’s health through the use of its stability index. Although the stability index may be a bit skewed on a freshly built server due to the large number of changes that may occur during its initial build, over time, the stability index can be a clear indicator of which of your servers requires more attention than others. It can also be a great troubleshooting tool by quickly pointing out any application installs or uninstalls that have occurred recently and any failures generated by the OS, software, or hardware on your system.

Data gathered from performance monitoring should then be used to assist in the decision-making around resource management. Windows System Resource Manager is a great tool that can allow you to prioritize your server’s processes to ensure that the server operates at optimal levels at all times and that should resource contention occur, the application you choose to take priority does indeed get those resources. This is especially useful for managing terminal servers and IIS servers, where potentially large numbers of processes may be running at any time, all contending for resources. WSRM lets you throttle how your processes obtain resources and tailor it to suit your business and technical requirements.
Network Policy and Access Services
When we talk about “protecting our network,” we are normally thinking in terms of perimeter security, such as firewalls and related host-based products such as anti-virus programs and patch-management tools. In the bigger scheme of things, though, our biggest threat comes from implicitly trusted hosts—systems we own and manage and for which we are responsible. It seems ironic, but it’s true. Although many organizations do a great job of securing their perimeters through the use of firewalls and funneling remote access through VPN solutions or remote terminal–based systems such as Terminal Services and Citrix, they typically have very little control over what happens inside the network.

Most of us have implemented logical or physical network segmentation to localize network problems, but what do we do when a user brings a laptop home and then returns to work the next day and plugs it straight into our network? For the time period that the device was off the network, it could have been infected by anything and everything under the sun. Suppose it’s an executive user who has Full-Control over his or her laptop and lets his or her kids play with it when it’s at home. Many large enterprises have resorted to granting all laptop users Full-Control because printer installs and help desk situations are often handled more gracefully if the user has Full-Control on his or her laptop. What if the user’s kids download programs from the Internet that could be harmful to the laptop? With mobile computing being on the rise, a common scenario exists: Devices from your controlled and secured network leave your sphere of control to mingle with the insecure world and then return to the secure network as though nothing happened.

Doesn’t it seem logical that when a device is connected to your network it should be considered untrusted by default and not allowed to talk to your trusted systems until it’s been thoroughly examined? This is what Network Access Protection (NAP) is all about. NAP is the Nirvana of network security—a world where an untrusted device is placed in quarantine from trusted devices until it has complied with a series of “health” checks. If it passes the tests, it is granted a pass into your trusted network. If it fails, it is given a chance to remediate the issue either automatically or manually and then undergoes the same health checks to ensure compliance. Not until an untrusted host becomes cleared by the system does it get access to the protected inner sanctum.

**NETWORK ACCESS PROTECTION**

What is NAP? First, I’ll tell you what it’s not. It’s not going to protect you from malicious users. NAP is an overall solution that lets administrators quarantine hosts that come onto the network until they have passed a series of defined health checks. Systems that do not pass the health checks are placed into a restricted state, where they are granted access only to specific hosts as needed to get back to a healthy state. This typically comprises anti-virus and patch-management servers, but it can be any server you need to make available to bring your systems into compliance. Once the health violation has been resolved, the system can then participate in your general trusted network.
Figure 8-1 shows an example of how NAP can be used to partition your network logically through the use of policy rather than topology. In this example, the partitioning is done using IPSec. Any new host entering the network is placed in the quarantine zone. Any host that then wants to get into the protected zone (for example, to communicate with one of your servers) will be subjected to a series of health checks. Those that fail even one of the checks will then communicate with remediation servers that reside in the boundary zone to get themselves compliant. Once compliant, they will be placed in the protected zone, where they are free to communicate with other hosts in that zone. Hosts are allowed to communicate only with other hosts in the same zone or the adjacent zone. Hosts in the boundary zone can talk to any system, while hosts in the quarantine zone cannot talk to any system in the protected zone, and vice versa.

Figure 8-1 is sort of a 100,000-foot aerial view of how NAP works. NAP is built around four major principles: policy validation (health checks), network restriction, remediation (getting healthy), and ongoing compliance. Ongoing compliance means that in order to remain in the protected zone, a system must continue to stay healthy. If a change in the state of the system brings it out of compliance with your NAP policy, it is kicked back into the quarantine zone and forbidden to talk to any protected zone hosts until the issue has been remediated.

For example, let’s say one of your policies states that the Windows Firewall must be on at all times. A user plugs his laptop into the network with the Windows Firewall enabled. It has now passed the health check and is given access to the protected zone members. If during the course of its operation the user decides to shut off the Windows Firewall, the next time the policy is evaluated it is no longer marked healthy and is
disconnected from all protected hosts until either the user turns Firewall back on or your remediation server turns it on for the user. This is a key issue: You can create remediation servers to bring your hosts into compliance automatically. You can also allow users to remediate themselves manually. In practice, you will want to have both methods available so that remediation occurs automatically, and if that automatic remediation step fails, the user is provided some sort of manual method for gaining compliance.

**NAP COMPONENTS**

NAP is actually one gigantic system. Without all the required pieces, it is not effective at all. In fact, one of the most prohibitive aspects of being able to implement NAP in your environment is cost. Depending on the solution you want to provide and how well you’ve kept your network infrastructure up to date, this can require sweeping upgrades across your enterprise. For example, you may need to upgrade older switches that don’t support 802.1X authentication. As a system, NAP comprises several components:

- ▼ IPSec enforcement
- ■ 802.1X enforcement
- ■ VPN enforcement
- ■ Dynamic Host Configuration Protocol (DHCP) enforcement
- ■ Network Policy Server (NPS)/Radius
- ■ NAP Agent
- ■ System Health Agent (SHA)
- ■ NAP administration server
- ■ System Health Validator (SHV)
- ■ Health policy
- ■ Accounts database
- ■ Health Registration Authority (HRA)
- ▲ Remediation server

The list is pretty long, but considering what you’re trying to accomplish as far as network security is concerned, each of these pieces plays a major part in making NAP come to life.

**IPSec Enforcement**

IPSec enforcement works by using X.509 certificates to control network access. Any host without a valid health certificate is not allowed to communicate with hosts that do have one. By using IPSec enforcement, hosts that require access must first request a certificate from the Health Registration Authority (HRA). The HRA checks for a host’s compliance
with the NAP policy. If it passes, the HRA obtains a health certificate from the certification authority (CA), which is then used to allow communication to other IPSec-enabled hosts with valid certificates. If it fails, the client is not given a health certificate but is instead given instructions on how to remediate itself. The host is then granted limited access to the network where the remediation servers reside. Once remediation has occurred, the host is rechecked for compliance and issued a valid health certificate if it passes; otherwise, it must undergo the remediation process again. This is the recommended method for NAP policy enforcement, as it is the strongest method for restricting network access.

TIP If yours is a mixed environment that includes hosts that currently do not support NAP, you can manually grant them access by creating exclusions for hosts and devices from health policy requirements.

802.1X Enforcement

In this network layer–based enforcement method for NAP, hosts requiring access are placed in relative isolation either through IP filters or virtual LAN (VLAN) segmentation until they pass the required health checks defined by the NPS. The 802.1X-compliant client connects to and initiates authentication with the 802.1X-compliant access point, such as an Ethernet switch or wireless access point. The NPS server then asks the client for its Statement of Health (SoH) if the authentication was successful. It then evaluates whether the SoH is compliant or not based on the current network policy. If it is valid, the 802.1X client is granted access to the protected network; otherwise, it is limited to sending traffic to remediation servers and stays with limited access until it finally complies with the health policy. It is important to note that clients can also gain access using a health certificate instead of a SoH when requesting access from the NPS server. Since this operates at the network layer and virtually isolates your untrusted hosts from the rest of the system, it is also a good choice for NAP enforcement and can work well in conjunction with IPSec enforcement.

VPN Enforcement

VPN enforcement is a good way to extend your NAP policy to protect yourself from users accessing your network remotely through VPN. NAP-aware VPN enforcement agents can then check for health compliance and grant or deny VPN access based on the NAP policy. Since inbound VPN connections typically make up the largest number of hosts that connect to your network that you might not directly manage, it is very important to implement some form of VPN enforcement as part of your overall NAP strategy. These remote systems are the weakest entry point into your network as they can easily be compromised. VPN clients connect to your VPN server and authenticate using Protected Extensible Authentication Protocol (PEAP) and MS-CHAP (Challenge Handshake Authentication Protocol) v2. Authenticated clients must then provide a Statement of Health that is evaluated by the NPS. The VPN client either gets an unrestricted connection or a limited connection based on whether it complies with the health policy.
DHCP Enforcement

If you don’t have complex equipment on your network, you can use DHCP enforcement. It involves limiting network access to your resources by either not assigning an IP address or assigning an IP address that has access only to your remediation servers if the host does not pass the necessary health criteria. This isn’t nearly as good as any of the other solutions because it relies on IP routing tables to secure your network. It can easily be defeated if someone knows some information about your network and simply manually assigns the host an IP address. Although not the best solution, it is probably still better than nothing for most environments and is at least an option if upgrading all your network equipment and implementing IPSec across your enterprise can’t be accomplished for one reason or another.

Network Policy Server/Radius

NPS is the replacement for Internet Authentication Service (IAS), Microsoft’s implementation of RADIUS (Remote Authentication Dial-In User Service), so logically NPS performs that role as well with Windows Server 2008. The difference is that NPS has extended that role to act as a policy server for NAP components. Health policy checks are defined in the NPS server, which also acts as the middleman for obtaining health certificates and connections to 802.1x and VPN devices.

NAP Agent

The NAP agent is the client used to collect information from all SHAs and transmit that information to the NAP Enforcement Clients (ECs).

System Health Agent

SHAs are the ultimate know-it-alls for how a component is evaluated in terms of health. Windows Server 2008 and Windows Vista contain a few built-in system health agents that allow it to evaluate information such as firewall and anti-virus status. A large number of Microsoft partners are also working on developing and releasing their own system health agents. You can add these components to your systems to provide a more in-depth health evaluation. For example, you might employ a third-party SHA for making sure that specific applications are installed on the system to be considered healthy. These agents then talk up to the NAP agent to consolidate and communicate this information back to the NPS to obtain the required approvals to gain full access to your network.

NAP Administration Server

The NAP administration server is responsible for taking all the data from SHVs and then determining whether to place a client into remediation or grant access to the protected systems.
System Health Validator

SHVs are the server components that determine whether a client is healthy or not, based on data submitted by the clients through SHAs. This response is then communicated back to the client using a Statement of Health Response (SoHR). These validators sit on the NPS server and compare incoming client requests against the policy set on the server.

Health Policy

These individual policies define the requirements for getting access to the protected network. A policy might ask whether the Windows Firewall is enabled and whether the network has anti-virus software that is not only running but is running the latest virus definitions. Multiple health policies can be defined on a system, one for each type of enforcement client. For example, you can define separate criteria for 802.1X access versus VPN access.

Accounts Database

This database is the central account authentication store. For all intents and purposes, Active Directory fulfills this role for Windows Server 2008.

Health Registration Authority

The role of the HRA is to act as a broker between healthy computers and the CA to obtain a health certificate to prove that the client has indeed passed all health checks. This must be run on a server running Windows Server 2008 and Web Services (IIS).

Remediation Server

When a client does not meet the defined health policy, it must remediate itself somehow. The client is granted access to remediation servers—a generic name that denotes any servers providing services to bring a noncompliant client back into compliance. Remediation servers are placed in the boundary zone between the quarantine and protected zones so that they are reachable by the quarantined clients.

DISPELLING NAP MYTHS

NAP can do many things to help you provide a safer network, but it can’t do everything. *NAP cannot protect you from malicious users.* This is a very important statement. Just because you have implemented NAP in your network doesn’t mean you are 100 percent safe. If you look at each of the different technologies involved in NAP, you will notice they are all about ensuring that the hosts attached to your network comply with specific health requirements. This actually has nothing to do with the user other than authentication.
NAP won’t prevent a malicious user from accessing a healthy system and running applications that might harm your network. NAP is not designed for that; it is simply designed to ensure that computers that participate in your protected network at least comply with standards you set. This certainly helps reduce the threat significantly, but it isn’t 100 percent bulletproof.

**NOTE** When you enable NAP, you might be afraid that you’re going to disconnect everyone from the network because they don’t all comply with the policy you defined. This is not the case. You, as the administrator, have the ultimate say for how your NAP-enabled devices will act in the event that one does not comply with your policies. For example, during initial roll-out, you may choose not to do anything but simply log the fact that a machine is not compliant. This auditing feature is a good starting point because it lets you see what could happen if your policy was in full effect and gives you time to remediate your noncompliant systems. If you’re careful, turning on NAP will be mostly transparent to your users, just as you want it to be.

NAP implementation isn’t going to happen overnight. Don’t expect to buy lots of hardware and have it up and running the next day (although I’m sure many of you—or more likely, many of your upper managers—might want just that). You need to bring to the table all the key players and every group that manages your infrastructure including network, server, and desktop resources. This is because the system cannot work unless all pieces are implemented just right. Your network team needs to ensure that your network infrastructure is up to par if you want to enable 802.1X or VPN enforcement. Your server team needs to make sure that correct infrastructure servers necessary to remediate an unhealthy system are available and accessible in your border zone. Your desktop team will need to verify that your desktops are running NAP-aware operating systems such as Windows XP SP2 (with the appropriate updates) and Windows Vista. Finally, you’ll need to sit together as a group and determine exactly what criteria defines a healthy system along with what action to take if a host is found not to be compliant.

**ARCHITECTURE**

Now that you understand the various components and how they are interdependent, let’s explore the NAP architecture in its entirety. Because pictures can often do a better job of demonstrating interactions and dependencies than words, we’ll start off by taking a look at how a NAP client interacts with the various NAP components in Figure 8-2.

As you can see in Figure 8-2, NAP clients that are both compliant and noncompliant with your health policy must be able to communicate at least with the key infrastructure servers in your boundary zone, including remediation servers if needed. The NPS that holds all your health policies and is responsible for procuring health certificates or validating statements of health records by NAP clients never interacts with the client directly. Instead, it interacts with the various authentication mechanisms sitting in the boundary zone. This is desirable since you want to secure your NPS server as much as possible.
This is not a technical requirement, however. In fact, the NPS can reside on the same server as your DHCP, HRA, VPN, or even remediation servers. Logically, the communication still follows that depicted in Figure 8-2, except that the NPS is technically visible by the NAP client. This is not a recommended setup, but it may be appropriate on small networks or when trying to demonstrate NAP functionality as some form of proof of concept.

**NAP CLIENT ARCHITECTURE**

NAP clients are systems that can participate in a NAP-enabled network because they have the ability to generate statements of health from agents installed on them. These system health agents not only check for the system’s health relative to their specific function (for example, an anti-virus system health agent may be able to query the anti-virus...
running state as well as engine and definition version), but they are also responsible for communicating with their respective remediation server to resolve the issues that mark them as unhealthy. Each client also has an enforcement client component that is responsible for limiting network access based on the medium for which it is responsible. For example, the DHCP enforcement client works with the appropriate NAP-enabled DHCP Server to ensure that the client obtains limited access only. The NAP agent then communicates and manages information regarding health states between the system health agents and the enforcement clients.

**Enforcement Clients**

Since four different methods exist for enforcing NAP client restrictions, four different NAP enforcement clients are responsible for managing the client’s ability to protect the network:

- **IPSec NAP EC** Stores health certificates issued by the NPS server. It then instructs IPSec to use the appropriate certificate during its communication with other NAP-enabled clients. It also controls the Windows Firewall to ensure that IPSec-enabled traffic is allowed through.

- **EAPHost NAP EC** Collects Statement of Health information from the various system health agents that is then sent using PEAP for 802.1X connections. If a health certificate is available, it can also use that to authenticate using 802.1X.

- **DHCP NAP EC** Collects a Statement of Health information and then passes it off to a NAP-enabled DHCP server through the use of DHCP options.

- **VPN NAP EC** Similar to the EAPHost NAP EC, it collects Statement of Health from various health agents that is then sent using PEAP to the VPN server. If a health certificate is available, it can also use that to certify health to the VPN server.

**System Health Agent**

SHAs on the client are matched with their respective SHVs on the server. The SHA’s purpose is to collect system health information that is then sent to the SHV. If the client is not compliant with the current policy, the SHV returns a SoHR to the SHA informing it of what steps it needs to take to remediate itself. This is why, in general, each SHA and its paired SHV must be from the same vendor, so that the SHV knows how to correct any policy violations found from the data provided by the SHA.

**NAP SERVER ARCHITECTURE**

Each NAP server contains a number of NAP Enforcement Server (ES) components, one for each type of authentication/connection method (that is, VPN or IPSec). These components are then matched to the appropriate NAP EC that matches the NAP ES.
For example, the IPSec NAP ES communicates with IPSec NAP-enabled clients. The NAP server in turn talks to the NPS using RADIUS. The NPS server contains the policies, NAP administration server, and SHVs. The NAP administration server acts as the broker between the NPS and various SHVs. It takes SoH records collected from the NAP clients through NPS and distributes them to the appropriate SHVs. It then returns the SoHRs provided by the SHVs back to the NAP clients through NPS.

Since NPS can be installed on a NAP server, it is completely possible, though not particularly recommended, that a NAP server have all the required components on one single server. The major downside to doing this is that you won’t have a central policy server and will need to configure your policy on each NPS server individually. This is both time consuming and error prone, which is why it is not best practice to do so. Out of the box, this architecture offers plenty of flexibility, because now you can add third-party SHAs and SHVs to your NPS for additional functionality. Microsoft has partnered with many solution providers to develop new and, in most cases, more powerful SHAs and SHVs to give administrators more control over what constitutes a healthy system. For example, this might involve SHAs and SHVs that check for registry keys or file versions—or maybe even go as far as checking local group settings.

**Enforcement Servers**

Each EC is matched up to an ES. Windows Server 2008 comes with only three ESs:

- **IPSec NAP ES** The NAP client’s health information is passed to the NPS server by the HRA. Access is controlled using health certificates.

- **VPN NAP ES** Passes health information between NAP clients and the NPS server using PEAP-TLV (Type-Length-Value) through Extensible Authentication Protocol (EAP)-RADIUS (encapsulating the EAP message in a radius message) and then restricts clients by IP packet filtering.

- **DHCP NAP ES** Uses industry-standard DHCP messages to communicate with the DHCP NAP ECs. Access is controlled using DHCP options.

You will notice that no EAPHost NAP ES is available to match the EAPHost NAP EC. Enforcement in this special case is actually handled by the 802.1X-enabled switches and access points by using IP packet filters or VLANs to isolate or grant access to the authenticating host.

**COMMUNICATIONS FLOW**

If all the preceding text didn’t confuse the heck out of you, then congratulations! My head was spinning the first time I tried to grasp the whole NAP concept. It’s really simple once you get to know it, but with so many acronyms, it’s easy to get lost. Figure 8-3 shows how the components communicate with one another for the purpose of evaluating health.
The SHA provides the SoH to the NAP agent. The NAP agent then passes this along to the NAP EC, which then passes it on to its corresponding NAP ES, which then hands it off to the NAP administrator, which then hands it off to the appropriate SHV. The resulting SoHR is passed back through the chain to the SHA on the client either to approve its connection or provide instructions for how to remediate itself. The EC and ES components control network access based on the resulting response.

**Hands-On Exercise: NAP Using DHCP Enforcement**

NAP is a complicated topic that requires expertise in many disciplines. You will notice that this chapter has significantly fewer hands-on exercises than previous chapters. This is simply because creating a full-blown NAP environment means you have to make configuration settings that are far beyond the scope of this book. However, one of the NAP enforcement methods we can use to demonstrate the NAP concept—because of its relative simplicity—is NAP using DHCP enforcement. In this multipart exercise, we set up a simple NAP architecture using DHCP enforcement to control network access for a NAP-enabled client.
Chapter 8: Network Policy and Access Services

Requirements
The minimal types of systems you will need for this exercise are

- Domain controller
- Windows Server 2008 server acting as network policy server
- DHCP Server (either on the domain controller or NPS server)
- DNS Server (required for Active Directory anyway)
- Client computer running Windows Vista

Preparation
For this exercise, set up a lab with two servers running Windows Server 2008 and one workstation running Windows Vista. The domain is called LABDOM.LOCAL and the systems are organized as follows:

<table>
<thead>
<tr>
<th>System</th>
<th>Setup Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>WIN2K8DC</td>
<td>Windows Server 2008</td>
</tr>
<tr>
<td></td>
<td>Domain Controller, Primary DNS Server</td>
</tr>
<tr>
<td></td>
<td>IP Address: 192.168.100.25</td>
</tr>
<tr>
<td>WIN2K8NPS</td>
<td>Windows Server 2008</td>
</tr>
<tr>
<td></td>
<td>DHCP Server, Network Policy Server</td>
</tr>
<tr>
<td></td>
<td>IP Address: 192.168.100.26</td>
</tr>
<tr>
<td>VISTAWKS</td>
<td>Windows Vista Ultimate workstation</td>
</tr>
<tr>
<td></td>
<td>IP Address: DHCP</td>
</tr>
</tbody>
</table>

Installing the Network Policy Server
I’m assuming you already have Active Directory set up and running and that the server you will use as your network policy server is already a member server in that domain. Your next step is to install the NPS.

1. Log on to the server where you are going to install the NPS.
2. Open Server Manager.
3. Click the Add Roles link to open the Add Roles Wizard.
4. Click Next on the Before You Begin screen.
5. Check the Network Policy and Access Services checkbox from the Select Server Roles screen, and then click Next (Figure 8-4).
6. In the Introduction to Network Access Services screen, click Next.
7. In the Select Role Services screen, check the Network Policy Server checkbox and then click Next (Figure 8-5).

8. Confirm the Installation Options, and then click Install.

9. Click Close when the installation has completed.

**Configuring the Network Policy Server**

After NPS is installed, you will need to configure NPS to use the Windows Systems Health Validator. You will configure this validator so that it considers a system healthy only if a firewall is enabled for all network connections.

2. Expand the Network Access Protection node.

3. Select System Health Validators, as shown in Figure 8-6.

4. Right-click Windows Security Health Validator and select Properties to open the Properties page (Figure 8-7).

5. Click the Configure button. If the Configure button is inactive (grayed out), your test server may require a reboot for Network Policy Server to start and make the button available.
6. In the Windows Vista tab, uncheck all the checkboxes except the Firewall checkbox, and then click OK (Figure 8-8).

7. Click OK on the Properties page to save the changes.

8. Expand the Policies node.


10. Enter **WSHV Compliant** in the Policy Name field. Select Client Passes All SHV Checks in the Client SHV Checks drop-down menu, and check Windows Security Health Validator, as shown in Figure 8-9. Then click OK.

11. Right-click Health Policies and select New.
12. Enter **WSHV Noncompliant** in the Policy Name field. Select Client Fails One or More SHV Checks in the Client SHV Checks drop-down menu, and check Windows Security Health Validator. Then click OK.


14. Enter **Full Access** in the Policy Name field and select DHCP Server as the type of network access server (Figure 8-10). Then click Next.

15. On the Specify Conditions screen, click the Add button.

16. In the Select Condition area, scroll down to the Network Access Protection section. Select Health Policies and click Add (Figure 8-11).

17. Select WSHV Compliant from the list of Health Policies and click OK (Figure 8-12). Click Next to continue.
18. Select Access Granted from the Specify Access Permission screen, and then click Next (Figure 8-13).

19. On the Configure Authentication Methods screen, check Allow Clients to Connect without Negotiating an Authentication Method, and uncheck all other checkboxes, as shown in Figure 8-14. Then click Next. Click No when you’re asked to view more help on the warning about selecting an insecure method.

20. Click Next on the Configure Constraints tab.
21. On the Configure Settings screen, select NAP Enforcement from the Settings pane on the left.

22. Select Allow Full Network Access and uncheck the Auto Remediation checkbox (you may need to scroll down to see this), and then click Next (Figure 8-15).

23. Review the Policy settings and then click Finish to save the new network policy.

Installing and Configuring DHCP

You can technically install and configure DHCP on a completely different Windows Server 2008 instance, but for this exercise, we will do this on the NPS server. Once installed, we will need to NAP-enable it and use DHCP options to control how clients are given access to the network based on the NAP policies.

1. Log on to your NPS server with an account that has domain admin privileges.
2. Open Server Manager.
3. Click the Add Roles link to open the Add Roles Wizard.
4. Click Next on the Before You Begin page.
5. Select DHCP Server in the Select Server Roles screen, and then click Next.
6. Click Next on the Introduction to DHCP page.
7. Select the network connection to which you want to bind the DHCP server, and then click Next (Figure 8-16).
8. Enter the preferred (and optionally the alternative) DNS server IP address, and then click Next (Figure 8-17).
9. Select WINS Is Not Required on This Network, and then click Next.
10. On the DHCP Scope screen, click Next, and then click Next again. We will configure scopes later.
Chapter 8: Network Policy and Access Services

Figure 8-11. Specifying policy conditions

Figure 8-12. Selecting a health policy
11. Select No, Do Not Configure This Server for DHCPv6 Stateless Operation Now, and then click Next.

12. On the DHCP Server Authorization screen, select Use Current Credentials, and then click Next. This assumes you followed step 1 and logged in with credentials that have permissions to authorize DHCP in AD. If not, you can also select Use Alternative Credentials and specify those credentials here.

13. Click Install after you have confirmed that the selections for the install are correct.

14. The next set of steps involves creating a DHCP Scope. Open the DHCP Server MMC snap-in: Choose Start | Administrative Tools | DHCP Server.
15. Expand your server and IPv4 in the content tree view.
16. Right-click IPv4 and select New Scope.
17. Click Next on the Welcome screen.
18. In the Scope Name screen, enter **NAP Client Scope** in both the Name and Description fields, and then click Next (Figure 8-18).
19. Enter **192.168.100.200** for the Start IP Address and **192.168.100.210** for the End IP Address. (Depending on your network setup, you may need to change these addresses.) Then Click Next (Figure 8-19).
20. Since we won’t be using exclusions, click Next on the Add Exclusions screen.
21. Click Next on the Lease Duration screen.

22. On the DHCP Options screen, select No, I Will Configure These Options Later, and then click Next.

23. Click Finish to complete the scope creation process.

24. Now you need to enable NAP on the DHCP server. Right-click IPv4 and select Properties.

25. Click the Network Access Protection tab and click the Enable On All Scopes button (Figure 8-20).

26. Click Yes when asked if you want to overwrite the NAP settings.

27. Click OK to close the IPv4 Properties dialog box.

28. Right-click the scope you created earlier and select Properties.
29. Click the Network Access Protection tab. Enable the Network Access Protection settings for this scope and check Use Default Network Access Protection Profile. Click OK to save the changes (Figure 8-21).

30. In the scope you created earlier, right-click Scope Options and select Configure Options.

31. Click the Advanced tab.

32. Select Default User Class in the User Class drop-down menu.

33. Click 003 Router under Available Options, and enter 192.168.100.1 as the IP address. (You will need to change this to the appropriate IP address for your default gateway.) Then click Add.

34. Click 006 DNS Server under Available Options, and enter 192.168.100.25 as the IP address. (You will need to change this to the appropriate IP address for your DNS server.) Then click Add.
Figure 8-17. Specifying DNS settings

Figure 8-18. Entering the DHCP scope name
Chapter 8: Network Policy and Access Services

**Figure 8-19.** Setting the scope IP range

**Figure 8-20.** Enabling NAP on all scopes
35. Click 015 DNS Domain Name under Available Options, and enter **LABDOM.LOCAL** in the String Value field. (You will need to change this to the appropriate value of your domain name.)

36. Click Apply to save the changes for the default user class.

37. Select Default Network Access Protection Class from the User Class drop-down menu.

38. Click 003 Router under Available Options and enter **192.168.100.5** as the IP Address. (This IP doesn’t have to be valid. We will use it to show how a NAP-enforced client will switch over to a different configuration if it is marked as unhealthy.) Then click Add.

39. Click 006 DNS Server under Available Options and enter **192.168.100.25** as the IP Address. (Use the same DNS server you specified for the default user class.) Then click Add.

40. Click 015 DNS Domain Name and enter **LABDOM.LOCAL** as the String Value. (Use your domain name value.)
41. Click OK to save your changes.

42. You can verify that all your settings are correct by clicking Scope Options and looking at the main view to see all the options you set along with their associated classes (Figure 8-22).

43. Expand the IPv4 node, right-click the scope you created, and select Activate. This will enable the scope so that it begins issuing IP addresses in that scope.

### Configuring the Client

Now our infrastructure is ready for our client to participate in NAP. Since the NAP service is set to start manually in Windows Vista, you will need to configure it to run automatically and then enable the NAP enforcement client. You will also need to enable Security Center manually since it is disabled by default when Windows Vista is joined to a domain.

1. Log on to VISTAWKS.

2. Click Start. In the Start Search field, enter `gpedit.msc`, and then press ENTER.

4. Double-click Turn On Security Center (Domain PCs Only) and select Enabled. Then click OK.

5. Open Control Panel.


8. Double-click Network Access Protection Agent, change the Startup Type to Automatic, and then click OK.

9. Click Start. In the Start Search field, enter `napclcfg.msc`, and then press enter. This will open the NAP Client Configuration console.

10. Click Enforcement Clients to open the Enforcement Clients window (Figure 8-23).

**Figure 8-23.** NAP client configuration window
11. Right-click DHCP Quarantine Enforcement Client, and then click Enable in the Actions pane.

12. Close the window and restart the computer.

Testing the NAP Client

Now that everything is set, you need to verify that all your settings are working. If you correctly followed all the steps, after rebooting your Windows Vista client computer, you will be assigned an IP address from the DHCP server as you would normally if you didn’t have NAP enabled. This is because the Windows Vista client computer has the firewall enabled by default on all network interfaces. If you open the Security Center from the Control Panel and disable the Windows Firewall on your network interface, within a few seconds your default gateway will be removed and you will be placed in restricted access mode. If you double-click the NAP client message on the taskbar, you will see the remediation message from the SHV stating that your computer is not compliant with the requirements of the network and that you must enable a firewall program that is compatible with the Security Center, as shown in Figure 8-24. If you re-enable the firewall, the NAP client detects this and then renews its IP address with full unrestricted access.
CHAPTER SUMMARY

Network Access Protection is an excellent solution for providing an additional layer of security for your network. Although it cannot protect you from malicious users that get on trusted computers, it does prevent unauthorized machines from accessing your network and potentially affecting your systems. It lets you establish minimum health requirements for any system that joins your network. This can be done through IPSec, 802.1x, VPN, or DHCP enforcement. IPSec is the recommended method since it allows your network to be logically subdivided into protected, border, and quarantine zones using health certificates as the controlling access method. DHCP is the easiest to implement but can easily be defeated by anyone who knows your internal network structure.

The System Health Agents either built into Windows or provided by a third-party manufacturer report health information to the NAP client, which is then sent to the NPS server for validation by the System Health Validators through NAP enforcement client and server components. If IPSec is involved, the NPS server then negotiates with the CA for a health certificate if the client passes all requirements.

This chapter gave you a taste of NAP's capabilities. What’s important to note is that NAP is now an integral component to Windows Server 2008. It is truly part of Microsoft’s strategy for a more secure computing platform. The great thing about NAP is that you are not restricted to Microsoft’s own technology. In fact, Microsoft is continually working with many third-party software developers to create more feature-rich NAP agents to accommodate a wide variety of different methods for measuring system health.

Unless you control all aspects of your network, you will need to gather a team of subject matter experts to get something as complicated as NAP up and running in a production environment. Careful coordination is necessary to ensure that all parts of your infrastructure can successfully co-exist in your NAP environment. For example, you may need the network team to upgrade the switches and access points to be 802.1x capable. The good news is that NAP can be implemented so that you can see who would be blocked based on your policies, and fine-tune them or remediate those noncompliant systems prior to enforcing your restrictions. This way, you can ease the system into your environment and prevent any unwanted side effects—such as disconnecting the CEO’s laptop from the network because she changed her configuration from your normal standards.
CHAPTER 9

Terminal Services
More and more companies are realizing the value of allowing employees and even clients to access their applications remotely and securely anytime, anywhere. Although virtual private network (VPN) solutions are a good choice for this, they can be less than ideal. For example, you may want to expose only one or two applications or control the environments in which an application runs. Terminal Services (TS) has fulfilled this role since its introduction with Windows 2000. It has also been extremely useful for administrators performing remote administration of their servers. While Terminal Services has improved in both performance and functionality, it has almost always needed help from third-party products to make it production-worthy. For example, many environments rely on Citrix layered on top of Terminal Services to provide direct application-based access rather than giving users a full remote virtual desktop environment through the standard Terminal Services.

Windows Server 2008 has added some much needed functionality to Terminal Services that gives it a much better value out of the box. A Terminal Services Gateway role now allows connections to occur securely over the Internet via HTTP over SSL (HTTPS) without the use of VPN connections. Terminal Services Remote Programs is another new feature that allows individual applications to be executed remotely while appearing to be local to the desktop. Finally, Terminal Services Web Access allows remote programs to be accessed through a web-based portal. These three new major features of Terminal Services fill in the missing functionality that previously forced administrators to look at third-party solutions to provide remote application access to their users. Some additional minor, yet useful, functionality has also been added to Terminal Services. Certain Plug and Play devices connected to the client computer can now be made available in the remote session. Terminal Services also supports monitor spanning and even supports the ability for a Vista desktop theme if the client or server hardware is sufficient.

If you read the previous paragraphs and think that Terminal Services in Windows Server 2008 forever replaces all the third-party remote access solutions out there, you are mistaken. Microsoft has designed Terminal Services to be a better value out of the box, but it is still designed for environments with minimal complexity. If your system has many applications and many users and needs to tailor the user experience, you will still need those third-party solutions. What Terminal Services for Windows Server 2008 does is provide a viable solution for companies with simple remote application requirements to implement centralized application hosting without having to implement much more costly and complex third-party solutions.

**TERMINAL SERVICES CORE FUNCTIONALITY**

The classic functionality that has existed with Terminal Services continues in Windows Server 2008 with a few changes, including improved usability, performance, and security. Windows Server 2008 comes with Remote Desktop Connection 6.1 and adds some nice eye candy to Terminal Services. For starters, it is capable of 32-bit color and font-smoothing. It allows you to view a session on multiple monitors, and it supports Terminal Services
Gateway servers, Network Level Authentication, and even certain Plug and Play devices, specifically media players and digital cameras. It even supports point-of-sale devices that use Microsoft Point of Sale (POS) for .NET 1.1.

Remote Desktop Connection 6.1

Remote Desktop Connection now supports a maximum resolution of 4096×2048. This applies even when using multiple monitors. To run Remote Desktop Connection using a custom resolution, you specify the width and height of the screen at the command prompt—or, if you have an .RDP file, you can add or edit the `desktopwidth` and `desktopheight` values there. For example, to set your remote desktop session to 1280×1024, you can run the following:

```
mstsc.exe /w:1280 /h:1024
```

You can span any number of monitors provided that the total resolution doesn’t exceed the maximum resolution for a remote desktop connection. This can be done in an .RDP file by changing the `span` value to 1 or by using the command prompt, like so:

```
mstsc.exe /span
```

If you are connecting from a Windows Vista workstation and want the same Vista desktop experience even when connecting to a Windows Server 2008 server, you can add the Desktop Experience feature on the server to which you are connecting using Server Manager’s Add Features Wizard. With the Desktop Experience feature, Terminal Services supports Windows Aero, the dynamic desktop experience for Windows Vista that provides visual enhancements such as translucent windows and taskbar buttons with automatic thumbnail previews. You simply need to enable and start the Themes service on Windows Server 2008 and apply the appropriate theme in the Appearance and Personalization Control Panel applet. If you set the theme to Windows Vista, Windows Server 2008 will attempt to use the Windows Vista theme. If it doesn’t have the requisite hardware to do this, it will still remain enabled. If a client computer does have the requisite hardware and connects using Terminal Services, Terminal Services will automatically use the Windows Vista theme for that connection.

SINGLE SIGN-ON

If you ask any user what he dislikes most about network security, remembering account names and passwords will be at or near the top of his list. Single sign-on isn’t just a buzz word; it’s something users want as part of their experience. Most users don’t want to have to log in multiple times. When you normally log in to Terminal Services, you are prompted for credentials for logging in to the terminal server. Although this is nice if you want to specify alternate credentials, many times you are simply re-entering the user credentials you used to sign into the workstation in the first place. Terminal Services on Windows
Server 2008 now supports single sign-on, which means the same user credential you used to log in to the workstation can be used to log in to the terminal server. The catch is that the participating systems must meet a few requirements before single sign-on can occur:

- The user accounts you want to set up for single sign-on must have rights to log on to both the workstation (via domain logon) and the terminal server.
- The client computer and terminal server must be part of a domain.

**Hands-On Exercise: Configuring Single Sign-On**

You need to make configuration changes to the client and the server to make single sign-on work. For the server, follow these steps:

1. Choose Start | Run. Enter `tsconfig.msc`, and then click OK. This will open the Terminal Services Configuration screen (Figure 9-1).
2. In the Connections section, right-click RDP-Tcp and choose Properties.
3. On the General tab, make sure that the Security Layer value is set to either Negotiate or SSL (TSL 1.0), as shown in Figure 9-2. Then click OK.

On the client side, you need to make some changes to the local group policy (although you could also configure this centrally using Group Policy objects, or GPOs):

1. Choose Start | Search. Enter `gpedit.msc`, and then press `enter`.
4. Select Enabled, and then click the Show button.
5. In the Show Contents screen, click the Add button.
6. Enter `termsrv/Win2k8srv1`, and then click OK. (Replace `Win2k8srv1` with the name of your terminal server.)

Figure 9-2. RDP-Tcp connection security layer
7. The Show Contents screen should now look similar to Figure 9-3. Repeat steps 5 and 6 for all the terminal servers you want to configure for single sign-on.

8. On the Show Contents screen, click OK to save the changes.

9. Click OK again.

10. Open a command prompt and run `gpupdate` to refresh the local policy.

Test your configuration by opening the Remote Desktop Connection client on your client computer (choose Start | All Programs | Accessories | Remote Desktop Connection). Then connect to the server you configured for single sign-on. You should automatically be logged in without having to enter additional user credentials.

**NOTE** Make sure you are logged on to a workstation with an account that has rights to log on to the terminal server.
INSTALLING TERMINAL SERVICES

So far we’ve been using the built-in administrative mode of Terminal Services, a core functionality that allows administrators to log in and administer a Windows Server 2008 server remotely. However, the full-blown Terminal Services is designed to host many more clients simultaneously and to be a remote application host. Getting this running requires installing the actual Terminal Services role on the server.

Hands-On Exercise: Installing Terminal Services

Installing the Terminal Services role is no different from installing any other role on Windows Server 2008.

1. Run Server Manager.
2. Click the Add Roles link to open the Add Roles Wizard.
3. Click Next on the Before You Begin screen.
4. On the Select Server Roles screen, check the Terminal Services checkbox, and then click Next (Figure 9-4).

![Selecting the Terminal Services role](image.png)
5. On the Introduction to Terminal Services screen, click Next.

6. On the Select Role Services screen, make sure that the Terminal Server checkbox is checked and all other boxes are unchecked, as shown in Figure 9-5. Then click Next.

If you are attempting to install Terminal Services on a domain controller (DC), you will see a warning message at this point in the installation process. Installing Terminal Services on a DC is not recommended due to performance and security considerations.

7. Click Next on the Uninstall and Reinstall Application for Compatibility screen.

8. On the Authentication Method for Terminal Server screen, select Do Not Require Network Level Authentication, and then click Next.

Remember that it’s best practice to require Network Level Authentication, as it is the more secure; however, it will limit operating systems that do not support Network Level Authentication, such as Windows XP systems, from connecting.

9. In the Specify Licensing Mode screen, select the Per Device licensing mode and then click Next (Figure 9-6). (If you are setting up an actual production server, you will need to select the appropriate licensing mode that you have

---

**Figure 9-5.** Selecting Terminal Server role services
purchased for your Terminal Services environment. The Terminal Services licensing options are covered later in this chapter.

10. On the Select User Group Allowed Access to This Terminal Server screen, add users or groups to grant them access to connect to the terminal server. Then click Next.

11. On the Confirm Installation Options screen, click Install. You’ll be warned that you may need to reinstall existing applications. This is normal, as you should install these applications after Terminal Services is installed. However, if an application was installed prior to the Terminal Services installation, you might need to install it after Terminal Services installation completes so it will become available on the terminal server.

12. After the installation completes, click Close. Then restart the server.

13. After restarting, you will see a warning indicating that the server cannot contact a licensing server. This is normal, since you did not install the terminal server licensing server as part of the installation.
TERMINAL SERVICES LICENSING

If you haven’t already established a terminal server environment and followed the steps in the preceding hands-on exercise, you will receive a warning that the server cannot locate a licensing server and that you have 120 days to configure a licensing server. This was designed to give administrators more than enough time to set up required license servers for their server farms. You should never go to production with a server running this provisional license.

Windows Server 2008 includes a service called Terminal Services Licensing that is used to manage Terminal Services licenses throughout your environment. It isn’t installed by default, but must be selected as a role service. You don’t need to install TS Licensing on every terminal server in your environment. One TS Licensing server can service multiple terminal servers. Without a TS Licensing server available, your terminal servers will be able to issue only temporary tokens rather than permanent ones to client devices. Terminal Services not only provides centralized license management, but it allows for license auditing and reporting for both Per Device and Per User licensing modes. This simplifies the license installation process, since you can simply update the licenses on your TS Licensing server and that will automatically be available to your terminal server farm.

Terminal Services Licensing is an efficient and lightweight service. In fact, even at high utilization, it doesn’t take up much memory or CPU utilization because the service is active only when the terminal server requests a token; otherwise, it is mostly idle. It uses minimal memory, typically no more than 10MB, and the database grows only 5MB for every 6000 tokens issued.


License Types

When you install Terminal Services on Windows Server 2008, you are prompted to indicate the license mode to use. You can choose Per Device license mode or Per User license mode. Per Device license mode is used if you want your licensing to be based on the number of devices connecting. Each new client device that connects will be issued a client license token. In a Per User license mode, client access licenses (CALs) are issued on a per-user basis rather than a per-device. This type of licensing scheme is nothing new—it’s the same type of licensing model used for most of Microsoft’s products.

The most cost-effective choice for your organization will be based purely on how your users access the system. Say, for example, that your company has 1000 users who
will access Terminal Services. If a one-to-one relationship exists between users and devices (that is, each user uses one unique device exclusively), then either licensing mode will do. On the other hand, if those 1000 users share 500 workstations because the users are split up into shifts, choosing a Per Device license mode cuts your licensing costs in half, since you would need to purchase only 500 CALs versus the 1000 CALs you would need if you chose a Per User license mode. Alternatively, if those 1000 users each accessed Terminal Services using both a workstation and a laptop, you would have to purchase 2000 CALs in a Per Device license mode as opposed to 1000 CALs in a Per User license mode. Needless to say, in that situation, going with Per User CALs is much more cost-effective.

You need to look at how your organization will connect to Terminal Services. Compare your user base with the number of devices used to access terminal servers. Whichever has the lower number will typically drive what licensing model best suits your environment. Microsoft changes its licensing plans quite a bit, so you should contact your Microsoft representative if you have any questions about what licensing scheme works best for your situation.

NOTE You should consider one additional factor when selecting one of the two licensing schemes. If you want to track Per User CALs, your terminal server and license server must be members of a domain, since it uses Active Directory Domain Services to track licenses. This will work even if you are running a Windows Server 2003 Active Directory.

Installing and Configuring TS Licensing

Three main steps are required to get TS Licensing up and running:

1. Install the TS Licensing role service.
2. Activate the TS Licensing server.
3. Install CALs on the TS Licensing server.

Installing TS Licensing works the same as installing the Terminal Server role service. The only difference is that if you are going to install TS Licensing on a pre-existing Windows Server 2008 terminal server, you will need to use the Add Role Services Wizard instead of the Add Role Wizard. Activation occurs once per server and can be accomplished using a number of methods—via a Web browser, telephone, or Internet connection. The Web browser and Internet connection activation methods differ. The Web browser method is used if you want to activate a terminal license server that does not have direct Internet connectivity. Instead, from any computer that has access to the Internet, you key in the registration information manually through a Web site and obtain the activation code from Microsoft. The Internet option offers automatic activation directly by the TS Licensing server to Microsoft’s servers over the Internet.
Hands-On Exercise: Installing TS Licensing Role Service

In this example, we add the TS Licensing role service to the existing terminal server we installed earlier. The minimal overhead of the TS Licensing service makes this a viable option in all but the largest of Terminal Services environments.

1. Open Server Manager.
2. Expand Manage Roles and select Terminal Services.
3. Click the Add Role Services link.
4. Select TS Licensing in the Select Role Services screen, as shown in Figure 9-7. Then click Next.
5. Select This Domain in the Configure Scope for TS Licensing screen, as shown in Figure 9-8. Then click Next.

![Select Role Services](image)

**Figure 9-7. Selecting the TS Licensing role service**
6. Click Install on the Confirm Installation options.
7. Click Close when the installation completes.

**Hands-On Exercise: Activating the TS License Server**

A production TS License server isn’t really any good until it has been activated. You have several options for activation: You can go directly through the Internet, fill out a form on the Web, or use the good old telephone method. This exercise demonstrates the steps for activating your TS License server over the Internet.

2. Right-click the server you want to activate and select Activate Server to open the Activate Server Wizard (Figure 9-9).
3. Click Next on the Welcome screen.

4. On the Connection Method screen, select Automatic Connection (Recommended) from the Connection Method drop-down menu (Figure 9-10). Then click Next.

5. Enter your Company Information, and then click Next.

6. Enter any optional additional company information you want to include, and then click Next. This will initiate the online activation.

7. Uncheck the Start Install License Wizard Now checkbox, and then click Close.
Hands-On Exercise: Installing Client Access Licenses

If you don’t install Client Access licenses for your terminal server, your activated license server can issue only 90-day temporary licenses. You will need to purchase and activate appropriate Per Device or Per User CALs to allow the server to issue permanent licenses. As for TS License server activation, you can install CALs using a Web browser, the telephone, or a direct Internet connection. This example uses the Web browser method, assuming that your server doesn’t have direct Internet access.

2. Right-click the server on which you want to install CALs and select Properties.
3. On the Connection Method tab, make sure that the Connection Method field is set to Automatic Connection (Recommended), as shown in Figure 9-11. Then click OK to close the Properties window.

4. Right-click the server again and select Install Licenses to open the Install Licenses Wizard.

5. Click Next on the Welcome screen.

6. On the License Program screen, select License Pack (Retail Purchase) in the License Program field (Figure 9-12), and then click Next. Depending on your licensing situation, you must select an alternate license program.

7. On the License Code screen, enter the License Code for each license you have purchased in the available fields, clicking Add after each code has been entered. When you’re done, click Next (Figure 9-13).

8. Click Finish.
Figure 9-12. Selecting the license program

Figure 9-13. Entering license codes
TERMINAL SERVICES GATEWAY

The Terminal Services Gateway lets you access terminal servers that reside in your corporate network, including secure servers that are protected by firewalls, from anywhere on the Web. It does this by encapsulating Remote Desktop Protocol (RDP) traffic over an HTTPS tunnel. This is a big advantage, because without TS Gateway, you would have to open up port 3389 for RDP connections. This has the added benefit of eliminating the need to implement a VPN solution if the VPN connection is used only for accessing terminal servers. You can configure policies to restrict access based on local user groups or Active Directory resources to which they can connect, or even domain membership of the client computer. You can even control whether device or disk redirection is allowed and whether smart cards are required for authentication. TS Gateway is tightly integrated with Network Access Protection (NAP), which allows you to limit access further, based on NAP policies. For even more protection, you can place TS Gateway servers in your private network by implementing a Microsoft Internet Security and Acceleration (ISA) server in your perimeter network.

NOTE

TS Gateway gives you access to any RDP-enabled service so you can use this to connect to your terminal server, as well as your clients, with Remote Desktop enabled.

Since it relies on other services to provide some of its functionality, TS Gateway requires the following:

▼ Windows Server 2008 server
■ Remote Procedure Call (RPC) over HTTP Proxy service
■ Web Server (IIS 7.0)
■ Network Policy Server
▲ SSL Certificate

TS Gateway Architecture

Depending on the number of users and servers you need to support, many of the previous services can sit on the same server or multiple servers. Figure 9-14 shows how the client connects to your terminal servers or other RDP hosts. From the Internet, a client will establish an SSL tunnel to the TS Gateway. Before a connection is granted, it checks the client credentials with its Connection Authorization Policies (CAPs) to determine whether the client is authorized to connect to the gateway. If authorized, the client can then request access to the resources on the private network. The gateway then checks whether the requested resource is listed in the gateway’s Resource Authorization Policies (RAPs). If this authorization check is successful, the gateway then connects to the requested resource. It completes the process by establishing a secure tunnel between the client and requested resource. The gateway acts exactly as a gateway should, by facilitating communication between the client and resource. At this point, the user must authenticate to that resource just as it would if it had attempted the connection from the local network. The only difference is that the communication is being encapsulated over HTTPS traffic through the TS Gateway.
Hands-On Exercise: Installing and Configuring TS Gateway

TS Gateway relies on multiple pieces. This hands-on exercise will focus on a specific scenario and then implement each piece as part of one big exercise. The scenario is straightforward: We will allow a Windows Vista client to access a Windows Server 2008 terminal server called WIN2K8TS through a Terminal Services Gateway, WIN2K8TSG, using the following steps:

1. Install the TS Gateway role on a Windows Server 2008 server called WIN2K8TSG.
2. Configure a certificate for the gateway.
3. Define CAPs on the TS Gateway.
4. Define RAPs on the TS Gateway.
5. Connect to the terminal server WIN2K8TS from the Windows Vista client.

This exercise will assume the following:

▼ An Active Directory domain controller is configured.
■ The Windows Server 2008 server called WIN2K8TS is running Terminal Services, which was installed and configured in the previous hands-on exercise, and is a member of a domain.
■ The Windows Vista client is a member of a domain.
▲ A cleanly installed Windows Server 2008 server is available for use as your Terminal Services Gateway server and is named WIN2K8TSG.
Installing the TS Gateway Role

Our first step is to install the TS Gateway Role onto our cleanly installed Windows Server 2008 server:

1. Open Server Manager.
2. Click the Add Roles link to start the Add Roles Wizard.
3. Click Next on the Before You Begin screen.
4. Select Terminal Services from the Select Server Roles Screen, and then click Next.
5. Click Next on the Introduction to Terminal Services screen.
6. On the Select Role Services screen, select TS Gateway. When prompted to install additional required role services, as shown in Figure 9-15, click Add Required Role Services. Click Next.
7. Select Choose a Certificate for SSL Encryption Later, and then click Next.

![Figure 9-15. Adding required role services for TS Gateway](image-url)
8. Select Later on the Create Authorization Policies for TS Gateway screen, and then click Next.
10. In the Role Services screen, verify that Network Policy Server is selected, and then click Next.
12. Click Next on the Role Services screen.
13. Confirm the Installation Options, and then click Install.
14. Click Close when the installation completes. If you are prompted to restart, do so now.

TS Gateway Certificates
TS Gateway relies on Transport Layer Security (TLS) 1.0 (SSL 3.0) for encrypting the communications between the client and the gateway. TLS 1.0 requires that an SSL-compatible x.509 certificate be installed on the server. You can obtain a certificate in many ways. If you already have a certificate issued by a root certification authority (CA) that participates in Microsoft’s Root Certificate Members Program and meets the requirements for TS Gateway servers, you can simply use that. If your company has an enterprise CA, you can use that to issue your certificate, provided that it is co-signed by a root CA that participates in Microsoft’s Root Certificate Members Program. If you don’t have an existing certificate, you have two options: You can purchase one or you can create and import a self-signed certificate. The only problem with using a self-signed certificate is that clients will receive warnings that the certificate comes from an untrusted source whenever they try to connect unless the clients have your self-generated root certificate imported into their trusted root certificate stores. Since we’re setting up only a test environment here, we can use a self-signed certificate and simply ignore the warnings.

The certificate must also comply with additional certificate requirements:

▼ The name in the Subject line must match the name configured in the TS Gateway server.
■ The Extended Key Usage (EKU) is Server Authentication (1.3.6.1.5.7.3.1).
■ It must have an associated private key.
■ It cannot be expired.
▲ If you configure TS Gateway with NAP support, it must also support encryption. The object identifier (OID) for this type is 2.5.29.15.

Creating a Self-Signed Certificate (Required if You Don’t Have a Certificate)
If you don’t have a certificate you can generate or use and don’t want to purchase one for the purpose of testing, your only option is to create a self-signed certificate. To create
a self-signed certificate, you can either generate one at the time of install, or, as this exercise will show, you can generate it at any time using the TS Gateway Manager.

1. Choose Start | Administrative Tools | Terminal Services | TS Gateway Manager.
2. Right-click your server name and select Properties.
3. Click the SSL Certificate tab and select Create a Self-signed Certificate for SSL Encryption. Then click the Create Certificate button.
4. Note the location where the certificate will be generated and click OK (Figure 9-16).
5. Click OK on the successful creation message.
6. Click OK on the server Properties window to save the changes.

**Installing a Certificate on the TS Gateway Server**

You should now have a certificate you can import into your certificate store for use on your TS Gateway server, whether it was something you generated or a certificate you already have. Your next step is to load the certificate onto your server. If you self-generated your certificate, this was automatically done for you by the self-signed certificate creation process. This exercise will show you how to import a certificate that you already have or that was generated manually.

1. Locate your certificate (that is, at the C:\users\sysadmin\documents\WIN2K8TSG.cer path you created) in your file system and double-click it to view detailed information (Figure 9-17).
2. Click Install Certificate. This will open the Certificate Import Wizard.
3. Click Next on the Welcome screen.
4. In the Certificate Store Selection screen, make sure Automatically Select the Certificate Store Based On the Type of Certificate is selected. Then click Next.
5. Click Finish to complete the installation.
6. Click OK on the successful import dialog box.
7. Click OK on the Certificate details screen.

**Configuring TS Gateway to Use the Certificate**

The purpose of getting a certificate is to configure it for use with the TS Gateway.

1. Choose Start | Administrative Tools | Terminal Services | TS Gateway Manager.
2. Right-click the TS Gateway server and select Properties.
3. On the SSL Certificate tab, you should see the Issued To, Issued By, and Expiration Date fields of your server certificate, as shown in Figure 9-18. If you generated a self-signed certificate, this information has already been imported and the certificate information will be displayed here. These fields will say “Not available” until you actually tell TS Gateway which certificate to use.

4. Click Select an Existing Certificate for SSL Encryption (Recommended), and then click Browse Certificates.

5. Select the Server Certificate you imported in the previous exercise, and then click Install.

6. Your server certificate information should now be displayed in the SSL Certificate tab where it previously said “Not available.” Click OK to close the dialog box.

![Figure 9-18. TS Gateway server SSL Certificate properties](image)
Configuring Connection Authorization Policies

Connection Authorization Policies define who or what can connect to the TS Gateway. For example, you can create policies that require a user to be a member of a certain AD Security Group or policies that require that a computer belong to a particular domain. This is your opportunity to limit who gets to connect through your gateway. In this example, we create a basic CAP that allows users that are part of the Domain Users group to connect.

NOTE You can create multiple CAPs on your TS Gateway. They will be evaluated in order and access will be granted as soon as a match is made.

1. Choose Start | Administrative Tools | Terminal Services | TS Gateway Manager.
2. Expand the node on the navigational tree representing your TS Gateway server.
4. Click Create New Policy from the Actions pane, and then select Wizard.
5. Select Create Only a TS CAP, and then click Next (Figure 9-19).

![Figure 9-19. Creating new authorization policies for TS Gateway](image)
6. In the Enter a Name for the TS CAP field, enter **Domain User Access**, and then click Next (Figure 9-20).

7. On the Requirements screen, make sure that Password is checked and Smartcard is unchecked (unless of course you use smart cards).

8. Click the Add Group button next to the User Group Membership list box. Enter **TESTDOM\Domain Users**, and then click OK (Figure 9-21). If you logged in locally, you will need to enter domain credentials at this point. Click Next to continue.

9. On the Device Redirection screen, make sure that Enable Device Redirection for All Client Devices is selected, as shown in Figure 9-22. Then click Next.

10. Review the TS CAP summary and click Finish to create the policy.

11. Click Close on the Confirm Policy Creation screen.
Chapter 9: Terminal Services

Configuring Resource Authorization Policies

CAPs perform a very limited function: They are designed to allow access to the TS Gateway. To access resources behind the gateway, those resources must be listed in the RAPs. In this example, we configure a local resource group to include all our terminal servers—which in this case consists of only WIN2K8TS—and then allow any member of the Domain Users group to connect to it.

1. Using Active Directory Users and Computers, create a new Computer Group called TS Servers, and then add WIN2K8TS to this group.
2. Choose Start | Administrative Tools | Terminal Services | TS Gateway Manager.
3. Expand the node on the navigational tree representing your TSG server.
5. Click Create New Policy from the Actions pane, and then select Wizard.
6. On the Authorization Policies screen, select Create Only a TS RAP, and then click Next.
7. Enter **WIN2K8TS Access** as the Policy Name, and then click Next.
8. On the User Groups screen, click Add Group. Enter **Domain Users**, and then click OK. Click Next to continue.
9. On the Computer Group screen, make sure that Select an Existing Windows Group is selected, and then click the Browse button.
10. Enter **TESTDOM\TS Servers** as the group name and click OK (Figure 9-23). Click Next to continue.
11. Click Next on the Allowed Ports screen.
12. Review the TS RAP Summary, and then click Finish to create the policy.
13. Click Close after the confirmation has been displayed.
Connect to Terminal Server Using a Client Through TS Gateway

Now all the legwork is done and we are ready to access our terminal server (WIN2K8TS) using our Windows Vista client through the TS Gateway. Since I can’t assume that you’ve created a fully segmented network for this exercise, we will validate that the client is indeed going to the server using the gateway and not directly, which is entirely possible, and we will monitor our gateway for the connections.

**NOTE** We use Windows Vista because it is already built with Remote Desktop 6.0. You can, however, perform this same exercise using Windows XP SP2 simply by installing the RDP 6.1 client from Microsoft.

1. Log on to your Windows Vista workstation.
2. The following steps are required only if you used a self-signed certificate. If you used a certificate from a Microsoft-trusted source such as VeriSign, you can safely skip to step 17.
3. Copy the root certificate that you copied over and installed on your TS
Gateway server to this workstation. If you created a self-signed certificate
using the exercise in this book, this file should be called WIN2K8TSG.cer. I
assume that you copied the file locally to C:\MyCertificates.

4. Choose Start, type MMC in the Start Search field, and then press ENTER.

5. Choose File | Add/Remove Snap-In.

6. Select Certificates, and then click Add.

7. Select Computer Account, and then click Next.

8. Select Local Computer, and then click Finish.

9. Click OK to close the Add or Remote Snap-ins dialog box.

10. Expand the Certificates node.

11. Right-click Trusted Root Certification Authorities and select Import from the
All Tasks pop-up menu to open the Certificate Import Wizard.

12. Click Next on the Welcome screen.

13. Click Browse, select C:\MyCertificates\WIN2K8TSG.CER (use the path to
which you copied your root certificate on the workstation), and then click
Open. Click Next to continue.

14. On the Certificate Store screen, verify that Place All Certificates In the
Following Store is selected and that Trusted Root Certificate Authorities is
specified as the Certificate Store, and then click Next.

15. Click Finish to import the certificate. You may be prompted with a security
warning; click Yes to install the certificate and click OK on the success message.

16. Close the MMC console and don’t save the console window.

17. Choose Start | All Programs | Accessories | Remote Desktop Connection.

18. Click the Options button.

19. Click the Advanced tab and click the Settings button under Connect from
Anywhere (Figure 9-24).

20. Select Use These TS Gateway Server Settings. Enter WIN2K8TSG.TESTDOM
.LOCAL in the Server Name field, select Ask for Password (NTLM) as the
Logon Method, and uncheck Bypass TS Gateway Server for Local Addresses.
Then click OK (Figure 9-25).

Make sure that the gateway server name you specify here matches the subject in
the certificate you installed; otherwise you will get an error that the server name
and subject name don’t match. For example, if you specified just the server name
and not the fully qualified domain name, you would see the error.
Figure 9-24. Remote Desktop Connection Advanced tab

Figure 9-25. Remote Desktop Connection Gateway Server settings
21. Go back to the General tab.

22. Enter **WIN2K8TS** as the server name, and then click Connect. Enter your user credentials when prompted.

23. To verify that you are actually going through the gateway and not going straight to your terminal server, log on to your TS Gateway server and open the TS Gateway Manager snap-in.

24. Expand your server in the tree view and select Monitoring.

25. In the Monitoring view (Figure 9-26), you will notice that you now have an open connection going to WIN2K8TS. Congratulations! You have successfully connected to a terminal server through TS Gateway.

---

**Figure 9-26.** Open connection to a Terminal Server through the TS Gateway
TS Gateway and NAP

TS Gateway is NAP-aware and can participate in your NAP infrastructure and enforce your policies. In addition to your CAPs and RAPs, you can also specify health policies that the client must meet to gain access to your terminal server environment. I won’t go into a fully detailed exercise on how to configure NAP on TS Gateway, but you can enable NAP enforcement by going into the properties of your TS Gateway in the TS Gateway Manager MMC snap-in. You will have a choice either to use a local Network Policy Server or go to a central NPS. You will then need to configure your System Health Validators and create new CAPs. What’s important to remember that is that for every policy you create, you must define two CAPs—one for PASS and one for FAIL. If you don’t configure a FAIL CAP, the user will receive a generic CAP failure message, if he or she is unable to connect due to system health policy violations, rather than a NAP-specific message which helps the user to understand and remediate the issue.

TERMINAL SERVICES REMOTE PROGRAMS

One of some users’ biggest issues with Terminal Services is that they end up with virtu-
ally two desktops—one local and one remote—when they connect to the terminal server. Depending on how tech savvy your users are, this can cause a lot of confusion, since they may not understand the difference between the two. All they know is that they have two Start menus with different programs on each. Terminal Services Remote Programs solve this problem by allowing users to access their applications remotely through Terminal Services while making it appear as though it were a local application. This blurring of lines between remote and local applications enhances the user experience by eliminating the annoyances of presenting two separate desktops.

You can access applications through a number of methods: Such as via Remote Desktop Protocol (.RDP) files with the appropriate connection information, or by adding the application directly on the user’s Start menu using a specially configured MSI (Windows Installer) file. The MSI file can also associate certain file extensions with a remote program. For example, if you allow users to use the Microsoft Office suite only through Terminal Services and a remote program, you can associate Word (.DOC) files with a remote program, so when a user double-clicks a Word file, it automatically initiates the Terminal Services session and begins the remote program. Finally, you can also set up Terminal Services Web Access and access the program by clicking a link on a Web site.

Remote programs are ideal for remote users or roaming local users. Rather than having an application loaded on multiple workstations, you can centrally host the application on your terminal server as a remote program and the users can access it anywhere without having to install the software on their computers. Suppose, for example, that you don’t want users to store sensitive business information on their workstations. By centralizing the information in an application in Terminal Services, you can limit and control access by funneling all connections through it. Remote programs can also simplify
application deployment. If your application is accessed across several hundreds or even thousands of different workstations, rather than deploying a fat client to each station, you can simply install the application in Terminal Server and deploy a smaller MSI file to allow users to connect to the remote program.

Requirements

Clients must be running one of the following operating systems to access TS Remote Programs:

- Microsoft Windows Server 2008
- Microsoft Windows Vista
- Microsoft Windows Server 2003 with Service Pack 1 or later
- Microsoft Windows XP with Service Pack 2

It’s no surprise that these requirements are the same for installing Remote Desktop Connection 6.1, since Remote Programs leverages RDC 6.0 for its functionality.

Installing Applications

Windows Server 2008 includes a few built-in applications such as Paint and Notepad, but you probably don’t want to set up remote programs for those applications. You’ll probably want to install “real” applications, such as Microsoft Office, on your terminal server and make them available to your users. Installing an application on a terminal server is not much different from installing it on any workstation, except that some nuances are involved in getting an application to work correctly in a TS environment—especially in a TS farm. You’ll need quite a bit of understanding of how the applications work and how Terminal Services deals with settings such as user registry keys and files that get loaded onto a user’s profile. We won’t go into too much detail here about these various tweaks, but you do need to pay attention to some very important steps.

Whenever you are installing an application on a terminal server, you must first change the server mode from Execute to Install. When logged on to Terminal Services, you can run in one of these two modes. You’ll usually choose Execute mode when you are simply logging in and running a bunch of applications. Install mode is a special mode used when you are installing an application on the server. In this mode, Terminal Services monitors changes made to the HKEY_CURRENT_USER registry key to capture changes made by the application installer. This information is then stored as shadow keys that are used to apply these settings to users who subsequently log on to the server in Execute mode.

Before installing an application, you should change the server to Install mode by running the following:

Change user /install
You can then install the application and configure it as needed. Once you are done, you should switch back to Execute mode using the following command:

```
Change user /execute
```

**NOTE**  Explicitly changing to Install mode and then back to Execute mode is not required if the installer is an MSI package. This is because Terminal Services is smart enough to recognize that an installation is taking place, and it automatically switches to Install mode and then back to Execute.

**Hands-On Exercise: Configuring a Remote Program**

In this exercise, we make the Windows Calculator available as a remote program. We then create an RDP file and an MSI file that we can use to distribute to users who will access the remote program.

1. Log on to the server that is running Terminal Services.
2. Choose Start | Run. Type `remoteprograms.msc` and click OK.
3. Click Add RemoteApps from the Actions list to start the RemoteApp Wizard.
4. Click Next on the Welcome screen.
5. Select Calculator, and then click Next.
6. Review the settings, and then click Finish.
7. Select Calculator from the RemoteApps list.
8. From the Remote Calculators Actions pane, select Create .RDP File.
9. Click Next on the Welcome screen.
10. In the Specify Package Settings screen (Figure 9-27), note the location where the RDP package will be saved (by default, it’s C:\Program Files\Packaged Programs). Then click Next. Note that if you want to specify TS Gateway settings, you can click the Security button and enter the settings there.
11. Review the settings and then click Finish. This will open the folder containing the RDP file that was just generated.
12. Close this window for now.
13. Select Calculator from the Remote Programs list.
15. Click Next on the Welcome screen.
16. Note the path to the package’s save location (which should be the same as the path shown in step 10), and then click Next.
17. On the Configure Distribution Package screen (Figure 9-28), check both the Desktop and Start Menu Folder checkboxes, and then click Next.

18. Review the settings and then click Finish. This will open the folder containing the MSI file that was just generated.

To access the remote program, you can either copy the calc.rdp or run the calc.rap .msi file on a client PC. If you copy the .rdp file, the user simply needs to double-click the file to initiate the session. If you install the Calculator link using the .rap.msi file, icons are created on the desktop and Start menu that can be used to initiate the session.

Here's how to run the Calculator using the .rdp file:

1. Copy the calc.rdp file to the desktop of your Windows Vista client.
2. Double-click the calc.rdp file to initiate the connection.
3. Enter your credentials to log on to your terminal server.

4. Select which devices on your local machine you want to make available on your Remote Programs session. Then click Yes (Figure 9-29).

5. The Remote Programs session will be initiated and a status dialog box will be displayed, as shown in Figure 9-30.

6. The Windows Calculator application will be displayed on your computer, as shown in Figure 9-31. Notice that the Calculator interface looks like a regular window. You are not running the Calculator program on your computer, however. Instead, you are running Calculator from your terminal server without the added clutter of a second desktop.
Figure 9-29. Remote Program trust prompt

Figure 9-30. Remote Programs Starting dialog box
The .RDP and .RAP.MSI methods for deploying icons to clients’ desktops to access remote programs are good choices, especially if you want to provide access to these programs in a transparent and seamless method. You can also provide access through a Web site using TS Web Access. You can think of TS Web Access as a portal into your remote programs. You can either use the standard default Web page included with TS Web Access or reuse the Web part into your own portal such as Microsoft Windows Sharepoint Services.

TS Web Access is a role service that can be installed onto a Windows Server 2008 server. It can be a terminal server, but it doesn’t have to be. In fact, if you are configuring TS Web Access so that your users can access remote programs over the Internet, you can use a plain Windows Server 2008 server as your TS Web Access server and configure it to use your TS Gateway to provide secure and easy access. By default, TS Web Access uses Active Directory as its source of remote programs. It does this by making available a RAP MSI file you have published to the user through a GPO. You can also configure TS Web Access to pull its list of remote programs directly from one of your terminal servers.

**Hands-On Exercise: Installing and Configuring TS Web Access**

In this exercise, we install TS Web Access on a Windows Server 2008 server and configure it to use a terminal server as its data source. For simplicity’s sake, we will reuse the TS Gateway server we set up earlier (WIN2K8TSG) to host our TS Web Access. We will then configure it to use the Active Directory as a data source and publish our previously created MSI package using a GPO.

1. Log on to WIN2K8TSG.
2. Open Server Manager.
3. Expand the Manage Roles item and select Terminal Services.
4. Click Add Role Services from the Terminal Services Role Services screen.
5. Select TS Web Access, and then click Add Required Role Services when prompted to install depended services (Figure 9-32). Click Next to continue.
7. Click Next on the Select Role Services screen.
8. Verify the installation options and click Install.
9. Click Close when the installation completes.
10. Since the server on which TS Web Access is installed is different from the server running Terminal Services and hosting our remote application, we have to add our TS Web Server to the TS Web Access Computers group on our terminal server.
11. Log on to WIN2K8TS.
12. Open Server Manager.
13. Expand Configuration | Local Users and Groups | Groups.
14. Double-click TS Web Access Computers and add WIN2K8TSG (make sure you specify to search for Computers as the object type). Click OK to save the changes.
15. While still on WIN2K8TS, share the folder where you created the calc.rap .msi file from the previous exercise. By default, this should be in C:\Program Files\Packaged Programs. When sharing it, make sure that domain users and WIN2K8TSG have read-only access to the share. Type in PackagedPrograms$ as the share name.

Note that Windows Server 2008 doesn’t install File Server by default, so you will need to add this particular role to the server before you can create a server share.
16. Log on to your domain controller or a computer where you can access Active Directory Users and Computers.
17. Open Active Directory Users and Computers.
18. Create an OU and name it **Remote Users**. This is not necessarily a requirement, but it allows you to test the GPO you are going to create without affecting the rest of your domain.
19. Create a regular user account in the Remote Users OU and name it **testuser**.
20. Add testuser to the Remote Desktop Users local group on WIN2K8TS. (Alternatively, you can add testuser to a global group and then add that global group to the Remote Desktop Users local group on WIN2K8TS.)
22. Click the Group Policy tab.
23. Click New to create a new GPO and type **Remote Programs** as the GPO name.
24. Select the Remote Programs GPO and click Edit.
27. Select `\WIN2K8TS\Packaged Programs$\calc.rap.msi`, and then click Open (Figure 9-33).

![Figure 9-33. Selecting the remote Calculator application](image)
28. Select Published as the deployment method, and then click OK.
29. Close the Group Policy Object Editor window.
30. Create a new OU called Terminal Servers and move the WIN2K8TS computer account to this new OU.
31. Right-click the Terminal Servers OU and select Properties.
32. In the Terminal Servers Properties dialog box, click the Group Policy tab, and then click Add.
33. Click the All tab (Figure 9-34), select Remote Programs, and then click OK. This will link your previously created GPO to this OU as well.
34. With the Remote Programs GPO selected, click the Properties button, and then click the Security tab.
35. In the Permissions area of the screen, grant WIN2K8TS Read and Apply Group Policy permissions to the Remote Programs GPO (Figure 9-35). Then click OK.
36. Click OK on the Terminal Servers Properties dialog box to save the changes.

**Figure 9-34.** Linking the Remote Programs GPO to the Terminal Servers OU
37. From WIN2K8TSG, open Internet Explorer, go to http://WIN2K8TSG/ts, and, if prompted, specify credentials of an account that is a member of the local administrator group.

Note that due to Internet Explorer’s enhanced security, you may need to add http://win2k8tsg to your list of trusted sites before you can access it.

38. On the TS Web Access Web page, click the Configuration button.

39. In the Editor Zone section, select Populate the Web Part from Active Directory Domain Services (Figure 9-36). Leave the Refresh the Web Part checkbox checked. Then click Apply.

40. You are now ready to test the new configuration. Log on to your Windows Vista client as testuser.

41. Open Internet Explorer and add http://win2k8tsg to the list of trusted sites.

42. Go to http://WIN2K8TSG/ts.
43. Notice the Calculator icon in the RemoteApp Programs screen (Figure 9-37).
   If you don’t see the Calculator icon when you connect to TS Web Access, it might be because your group policies haven’t updated yet. Try running `gpupdate` and then going back into TS Web Access to see if that helps.

44. Click the Calculator icon.

45. You will see a trust warning. Check the box that says not to warn you again, and then click Yes.

46. Enter your credentials for logging on to WIN2K8TS. Then click OK.

47. Click Yes on the prompt to trust the computer to which you are connecting.
   If you don’t want to be prompted again, simply check the Don’t Prompt Me Again for Connections to This Computer checkbox.

The Calculator application should now be running on your computer just like a regular application.
Chapter 9: Terminal Services

PROGRAM PLACEMENT AND PERFORMANCE

In a real production environment, you will undoubtedly have more than just one terminal server hosting your applications. Your decision of which servers will be hosting your applications will be strongly based on two criteria: the number of users simultaneously accessing the applications relative to the server’s resources, and the application’s ability to co-exist with other applications. For example, if an application has many dependencies that are version-specific, such as database clients and other runtime engines (such as Sun’s Java Runtime Environment), you may want to install the application on a separate terminal server from other applications with similar dependencies to avoid any conflicts.

You should leverage the performance management tools inherent in Windows Server 2008 to create baselines for your terminal servers and to track capacity and utilization as applications are installed and used. CPU and memory utilization of applications vary depending both on the application and the functionality being used, so there’s no real rule of thumb to define how many users can simultaneously use your server without severely...
degrading performance and usability. You will need to compare your benchmark data with data you retrieve once users start accessing TS-hosted applications to get more accurate metrics for your applications. Your goal should be a reasonable estimate per user, per application, in terms of CPU and memory use. This will let you easily estimate how many users can co-exist on one server of particular server specifications. This, however, is much easier said than done, since many applications use resources differently depending on what function they’re performing. If you trend your data long enough, though, you should be able to come up with reasonable numbers. Work your servers to about 75 to 80 percent utilization and stop there. Any fluctuations in resources required by your clients can be handled by the server.

CHAPTER SUMMARY

Out of the box, Terminal Services is a useful and feature-rich product. Upgrades to the security capabilities of Terminal Services, such as network level authentication, give you the option of increasing your security while still providing a user-friendly experience. The TS Gateway role service is a much welcomed addition to the TS services lineup in that it facilitates securing your terminal servers by controlling access to them at your perimeter and allows you to keep your servers within your secure network. TS Gateway is also NAP-aware and can participate in verifying client health to ensure that only clients that comply with your client health policies can access it. For those who host extranets, TS Web Access adds even more value by making it easy to link to applications through your Web server. This is useful if you don’t control the clients connecting and deploying .RDP or .MSI files.

At the surface, it may seem that the new functionality Microsoft has provided for Terminal Services in Windows Server 2008 is designed to eliminate the need for third-party systems such as Citrix. This is certainly not the case. Although Terminal Services now has much of the missing functionality that many administrators found with third-party companies in the past, it still is not a truly enterprise production-scale solution. The out-of-the-box solution is a good option for small environments or environments with medium to light remote application use and that have fairly uncomplicated configuration requirements. For true, enterprise production-class remote access server farms, you will probably still want to look at Microsoft’s partner solutions that layer on top of this new core functionality to get what you need.

Terminal Services in Windows Server 2008 is a far more mature product than ever before. It is an excellent new feature that can open the doors for many organizations that have wanted to provide remote application access but have been significantly hampered by its associated cost.
CHAPTER 10

Windows DNS, BitLocker Drive Encryption, and Itanium Support
Windows Server 2008 includes some important enhancements to Windows Domain Name System (DNS). This isn’t such a surprise; since Active Directory is so reliant on DNS, whenever we see major changes to Active Directory, we can expect some equivalent changes in Windows DNS. DNS for Windows Server 2008 now supports background zone loading, IPv6, GlobalNames Zones, Read-Only DNS, and a feature for DNS clients called link-local multicast name resolution (LLMNR). Another great addition to Windows Server 2008 is the inclusion of BitLocker Drive Encryption for added security through a combination of hardware and software components. This feature helps prevent unauthorized access to server volumes even if physical access is somehow obtained to the actual drives.

As enterprise computing requirements are increasing, so is the demand for 64-bit computing. Windows Server 2008 for Itanium-based systems is a highly specialized version of Windows Server 2008 that is designed to be a great platform for applications that require scale-up in terms of local resources (processing power, memory, and so on) rather than scale-out, which means load-balancing across multiple servers.

**DOMAIN NAME SYSTEM**

DNS is a hierarchical naming resolution service for TCP/IP. It is the primary name resolution service used to navigate through the Internet and is also the primary name resolution service used by Active Directory. Its function, first and foremost, is to translate host or domain names into IP addresses. It can optionally be used to perform reverse lookups where hostnames are resolved from IP addresses. Active Directory uses DNS in a special way, in that it uses a particular record type called **SRV records** to locate key Active Directory infrastructures such as domain controllers.

Chapter 4 covered some of the basics of DNS and Active Directory. Some new features are specific to the DNS implementation in Windows Server 2008 and are covered in this chapter. Windows Server 2008 DNS provides the following features out of the box:

- **Active Directory Domain Services support**  Windows DNS is the DNS server solution Microsoft recommends to support Active Directory Domain Services (AD DS). Although technically you can use third-party DNS solutions that support the SRV record types for your Active Directory, you will not be able to take advantage of its tightly integrated features such as the ability to store the DNS data in the AD domain or application partition. Windows DNS also supports the ability to perform secure dynamic updates of DNS records by clients participating in the domain. This way, your host entries in your DNS server will always contain the correct DNS entry for that hostname.

- **Stub zones**  Stub zones contain only a partial copy of a zone that contains only resource records needed by the authoritative DNS servers for that zone. This increases DNS resolution efficiency by keeping records of authoritative DNS servers for its child zones.
Integration with other MS networking services  Windows DNS supports integration with services such as Windows Internet Name Service (WINS) and Dynamic Host Configuration Protocol (DHCP).

Better administration tools  Windows Server 2008 comes with enhanced interfaces to make managing Windows DNS easier. This includes the addition of wizards to help simplify administration.

Dynamic update support  Windows DNS supports dynamic updates as specified by RCF 2136. This is an important feature if you use DHCP in your environment and need to keep your DNS records up to date without administrative intervention.

Incremental zone transfers  To optimize replication, Windows DNS supports incremental zone transfers to other DNS servers so that only records that have been updated get replicated to those servers.

Conditional forwarders  You can forward unresolvable addresses to another DNS server. Conditional forwarders allow you to specify name resolution requests for a particular domain to a specific DNS server. For example, you can create a conditional forwarder to forward any attempt to resolve hosts in the Microsoft.com domain to a specific DNS server instead of your normal forwarder server.

Background Zone Loading

Windows Server 2008 DNS includes a number of performance enhancements, including background zone loading. This feature allows the DNS server to begin responding to clients almost immediately after it has been restarted; in the past, the server would have had to wait to retrieve the DNS data from AD DS. Although it can’t respond to requests for host information that hasn’t yet been loaded from AD DS, it can begin to respond to requests that are designated to be forwarded (for example, requests for Internet sites) for any host information stored in files.

When a Windows Server 2008 DNS service starts up, it follows this procedure:

1. Enumerates all zones to be loaded
2. Loads root hints
3. Loads all file-backed zones (any zone information not stored in AD DS)
4. Immediately begins responding to clients’ requests
5. Spawns new threads to load zone stored in AD DS

In previous Windows Server DNS systems, step 4 (responding to clients) was contingent upon step 5 (obtaining all the zone information from AD DS) to complete. Naturally, depending on the number of records your DNS hosts, this could be a very lengthy process, effectively neutering your DNS server until it has retrieved the requisite data. Larger organizations will typically notice a significant performance advantage with this new architecture.
IPv6 Support

IPv6 (IP version 6) is slowly gaining popularity, mostly out of necessity. The fact is, if we don’t convert to IPv6 in the near future, we will simply run out of usable IP addresses. IPv6 uses 128 bits to specify IP addresses versus the traditional 32 bits used by IPv4. The catch is that in order to take advantage of IPv6, you will need to have IPv6-capable networking equipment as well as operating systems that are IPv6 capable, such as Windows Server 2008 on the server side and Windows Vista on the desktop side.

Since the change in addressing will affect every piece of infrastructure that deals with TCP/IP, Microsoft has included IPv6 support into Windows Server 2008 DNS. This allows DNS entries to be specified either as IPv4 or IPv6 addresses. In addition to this, command-line tools for managing DNS, such as DNSCMD.EXE, also support using IPv6 as parameters. This support doesn’t stop with host entries, as it can also forward to or perform recursive queries on IPv6 servers. DNS also supports the ip6.arpa domain namespace for reverse name resolution of IPv6 addresses.

**IMPORTANT** Microsoft strongly recommends that your DNS clients are upgraded to support IPv6 as well. This is because name resolution against a Windows Server 2008 DNS server can result in either an IPv4 (A) record or an IPv6 (AAAA) record. This isn’t a hard and fast requirement, but it is recommended because it might cause some problems if your DNS client receives an IPv6 address response from your DNS server.

GlobalNames Zone

If you are still operating WINS, you will be very interested in the GlobalNames Zone feature. Many organizations to this day still use WINS in addition to DNS to provide name resolution. WINS provides a single name to IP address mapping. In some cases, legacy applications drive the need for WINS in the environment. WINS is based on NetBIOS over TCP/IP, which isn’t a bad protocol, but it is nonetheless obsolete. Windows Server 2008 DNS allows you to create a new type of zone called the GlobalNames Zone. The replication scope of this zone is forest level to ensure that the names are unique across the entire forest. This can help facilitate many organizations’ goals to move to a strictly DNS-driven environment.

Read-Only DNS Zone

In Chapter 4 you read about the new Active Directory features in Windows Server 2008, including a new role called a read-only domain controller (RODC). To complement this feature, Windows Server 2008 DNS now supports a new zone called a *primary read-only zone*. This zone provides a read-only copy of the DNS zone information to requesting RODCs. RODCs replicate the DNS application partition and store this as a read-only zone. Administrators can view any entry in the read-only copy just as they could a regular DNS server, but if changes are to be made, they must be done on a server that is not set to read-only mode.
Chapter 10: Windows DNS, BitLocker Drive Encryption, and Itanium Support

Windows Link-Local Multicast Name Resolution

Microsoft Windows link-local multicast name resolution (LLMNR) might sound complicated, but it’s simply the way Windows Server 2008 (and even Windows Vista) can resolve hosts local to their network segment without the use of DNS. In the past, this feature was facilitated by WINS or NetBIOS, but WINS and NetBIOS support only IPv4. Now that Windows Server 2008 supports IPv6, a different way to resolve names without DNS is needed. Why would you ever need LLMNR? Suppose you’re on a small network running a workgroup rather than a domain. Your client’s DNS server addresses may be pointing to a DNS server out there on the Internet; how are you supposed to resolve names of hosts in your own network? LLMNR is the answer.

For example, suppose you wanted to ping a host called WIN2K8TEST from your server. Your server first queries its configured DNS server. If a DNS server is found, it attempts to query the server for that hostname. If that server cannot resolve the hostname—and assuming your DNS server is pointing to a host out on the Internet, then it won’t—it sends a multicast query over UDP for that hostname. Each host on your network that supports LLMNR checks to see if the hostname matches its own hostname. If it doesn’t, it discards the packet. If it does match, the matching host then sends a UDP packet back with its IP address.

Since LLMNR is specialized, it can respond only to requests where a single hostname is entered. If you enter a fully qualified domain name (FQDN), LLMNR will not resolve it. LLMNR is also responsible for making sure that its hostname is unique in its segment. This ensures that requests for name resolution don’t result in duplicate matches. If the LLMNR-enabled host receives a request for name resolution and it has not yet checked whether the name is unique, it marks its response back to the requesting host of its address but also sends an indication of this particular condition—that is, the requesting host receives two replies. The host that has performed a uniqueness check gets accepted and the one that hasn’t performed the check does not (even though there is in fact a conflict).

LLMNR is enabled by default on all Windows Server 2008 installations. On some occasions, you or your security policies might dictate that this functionality be disabled. LLMNR can be disabled on all network interfaces or on a specific network interface. To disable LLMNR on all network interfaces, create and set the following registry value to 0 (zero):

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DnsCache\Parameters\EnableMulticast
```

To disable LLMNR on a specific network interface, create and set the following registry value to 0 (zero):

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters\<adapterGUID>\EnableMulticast
```

Note that every network adapter is assigned its own unique GUID. You will need to find out which GUID represents your network adapter and replace `<adapterGUID>` with that value.
WINDOWS BITLOCKER DRIVE ENCRYPTION

One of the biggest challenges with servers in a remote location is ensuring physical security. It’s not always possible to secure physical access to a remote datacenter. Servers must sometimes be placed in less than ideal locations, where the risk of physical compromise is greater. For example, organizations with small satellite offices might require that their data reside on local servers for performance reasons (that is, when high-speed WAN links are cost prohibitive). In such situations, it’s not unusual for the server to be placed in an unused closet or even under someone’s desk. Your server might also be physically co-located in shared datacenters with servers from other organizations, and it’s not always possible to control who outside of your organization can physically access your servers.

Although every systems administrator works hard to avoid insecure setups whenever possible, options and choices are sometimes limited, and we do the best we can with what we have. The biggest problem with a less than ideal setup is that some unauthorized individual with physical access to the servers might try to get to the data directly on the disks using boot CDs or USB drives, or the person may actually move the physical hard drive to a different computer. Such a threat requires that data protection be in place to avoid the system being compromised, even if physical access is somehow obtained.

BitLocker, introduced in Windows Vista Enterprise and Ultimate editions, is a security feature that can protect data from physical access. It protects operating system files and any other volume you designate as being protected by BitLocker. It works in conjunction with the Trusted Platform Model (TPM) chip on the system to make sure that all components that load during the operating system’s boot process are not compromised. This protection remains in place even if the operating system is shut off. TPMs contain special registers called Platform Configuration Registers (PCRs) that store the hash value of the various startup components, including the BIOS, Master Boot Record (MBR), boot sector, and boot manager code.

Requirements

A few requirements are necessary prior to your enabling the BitLocker Drive Encryption feature. You can install this feature at any time, but to take advantage of its full functionality, your system must meet the following requirements:

- A system with a version 1.2 TPM chip
- A Trusted Computing Group (TCG) compatible BIOS
- At least two partitions on your system: one system partition set to active and another in which you will load your operating system (the boot partition)
- A BIOS that supports the USB mass storage device class for booting from a USB flash drive
BitLocker Architecture

BitLocker performs full-volume encryption, which makes it mostly transparent to the system, except during startup when a few additional steps might be needed for authentication before the volume can be unlocked. The volumes are protected using a 256-bit full-volume encryption key. This key is then protected by a 256-bit volume master key. The volume master key is in turn protected by several methods, depending on the authentication method you have specified. The following authentication methods are available:

- **TPM only**
- **TPM plus PIN (4–20 digits)**
- **TPM plus startup key**
- **Clear key**
- **Startup key or recovery key**
- **Recovery password**

Each method provides various levels of protection for the volume master key. Your selection of a method depends on your environment or particular scenario and requirements for balancing the need to safeguard the data with ease of use and recoverability.

**TPM Only Authentication**

As its name implies, *TPM only authentication* means that the volume is unlocked directly by the TPM using a 2048-bit key. This provides a good but relatively low level of security, because starting the system physically will simply boot it up as normal, since the TPM will automatically unlock the drive as long as the startup files are not altered. It protects the data on the volume only from being read from a completely different system. By moving the hard drive to a different computer or replacing a motherboard, the TPM will no longer match and the server will not boot unless a successful recovery takes place. TPM only authentication method also protects the system by ensuring that the startup files are not tampered with, in which case it would fail the checks in the TPM. This option is ideal if your servers are generally in a secure location to begin with or if they are remotely located so that you cannot easily interact with the system during the startup process.

**TPM Plus PIN**

While TPM only authentication beats not having any authentication whatsoever, it is still slightly vulnerable since the TPM contains all the data required to authorize unlocking the volumes. One way to mitigate this risk is to leverage multifactor authentication. In this case, we can also require that a PIN be entered in addition to the TPM checks to succeed. TPM plus PIN method combines the data from a 4- to 20-digit PIN encoded in SHA256 with the TPM’s 2048-bit key to unlock the volume. Requiring that a PIN be entered increases the level of security, since one of the keys needed to retrieve the volume
master key is no longer physically on the system but rather in someone’s head (and, hopefully, not written on a piece of paper next to the server).

**TPM Plus Startup Key**
This authentication method is similar to the TPM plus PIN method, except that instead of typing a PIN, we are required to insert a USB flash drive containing a startup key. The 2048-bit TPM key reads the hash values in the PCR and generates a 256-bit intermediate key. This intermediate key is then masked with the 256-bit startup key using the XOR (Exclusive OR) operator to retrieve a second 256-bit intermediate key that then unlocks the volume master key.

**Clear Key**
This isn’t really an authentication method, but in this form, the volume master key is stored in a symmetric format on the boot volume, essentially making it readable. This method is not secure at all and is in effect only if you disable (but not uninstall) BitLocker. You might use the clear key method, for example, if you need to restart a server that is configured to use the TPM plus startup key method and you are in a remote location and are unable to connect the physical USB key device to the server to allow it to boot.

**IMPORTANT** You should avoid using the clear key method whenever possible, but if it is your only choice, you can minimize your risk by re-enabling BitLocker as soon as physically possible.

**Startup Key or Recovery Key**
This option is your only choice if your system doesn’t support TPM or if your TPM module is unavailable (it’s been shut off or it’s malfunctioning). You can configure your server to retrieve the volume master key or a recovery key directly from a USB flash drive. The recovery key might be needed if for some reason the original authentication method cannot be performed, for example, because the TPM isn’t working or was replaced, the user forgot the PIN, or the USB key holding the startup key is unavailable. Recovery keys allow new keys to be generated safely and efficiently.

**Recovery Password**
The recovery password method is exactly the same as the recovery key method, except the former requires that you enter a password. It is recommended at the very least that when enabling BitLocker on a system, a recovery password is set should the data on the drive need to be recovered when none of the other authentication methods are available.

**Hands-On Exercise: Preparing for and Installing BitLocker**
One of the main requirements of BitLocker Drive Encryption is that you must have at least two partitions on the system. The first partition is the active partition, otherwise known as the system partition. This should be 1.5GB in size and formatted using NTFS.
Chapter 10: Windows DNS, BitLocker Drive Encryption, and Itanium Support

Windows Server 2008 is installed on the second partition. This partition can be any size, provided of course that it fits your Windows Server 2008 installation. In this exercise, we will prepare for the BitLocker installation by creating the two required partitions using the diskpart.exe command.

NOTE This exercise will erase the contents of the hard drive for the system on which you run it. Make sure no important data resides on the hard drive before proceeding.

1. Boot the system using the Windows Server 2008 DVD.
2. Select the language to install, time and current format, and keyboard or input method, and then click Next.
3. On the Install Now screen, click the Repair Your Computer link.
4. If the drive you have contains an existing operating system, it will be selected by default on the System Recovery Options screen. Make sure that none of the operating systems detected are selected by clicking the white space under the operating system names. Then click Next (Figure 10-1).
5. Click the Command Prompt link to open a new command prompt, as shown in Figure 10-2.
6. At the command prompt, type `diskpart`, and then press ENTER.

![System Recovery Options](image)

**Figure 10-1.** Make sure none of the operating systems are selected.
7. Type `list disk` and press `enter`. This will display the list of detected disks on your system, as shown in Figure 10-3. Take note of the disk number of the disk you want to partition. On my test system, I had only one disk (Disk 0), as shown in the figure.

8. Type `select disk 0` and press `enter`. Your disk will also probably be Disk 0 as well, but if it isn’t, simply replace 0 with the disk number you saw in the previous step. You will receive confirmation that the disk has been selected.

9. Type `clean` and press `enter`. This will erase the existing partition table.

10. Type `create partition primary size=1500` and press `enter`. This will create a primary partition that is 1500MB in size.

11. Type `assign letter=Z` and press `enter`. This assigns the letter Z to this partition. (You can use whatever drive letter you want; I prefer Z since it is the last usable drive letter.)
12. Type `active` and press `enter` to make this the active partition.

13. Type `create partition primary` and press `enter`. This will create a second primary partition on the hard drive that fills up the remaining available space. Windows will be installed in this partition. You can also append the `size=XXX` parameter to this command if you want to specify the exact size of this partition.

14. Type `assign letter=C` and press `enter`. We will assign the letter C to this partition since Windows is typically installed on the C: drive. (Again, you can choose whatever drive letter is appropriate to your environment.)

15. Type `list volume` and press `enter`. This will list all the available volumes so that you can make sure the partitions were created successfully and that the correct drive letter was assigned (Figure 10-4).

16. Type `exit` and press `enter` to quit diskpart.

17. Type `format C:/q /fs:NTFS` and press `enter`. This will quick format the C: drive in NTFS format. Press y when prompted to Proceed with Format, and then enter the Volume label you want for this partition.

---

**Figure 10-3.** Disks detected by `diskpart`
18. Type `format Z: /q /fs:NTFS` and press enter. This will quick format the Z: drive in NTFS format. Press y when prompted to Proceed with Format, and then enter the Volume label you want for this partition.

19. Close the command prompt.

20. Close the System Recovery Options window to continue with the installation of Windows Server 2008. Do not click Shut Down or Restart. Doing so won’t cause any harm, but you will have to boot from the Windows Server 2008 media again to continue with the installation. Simply closing the System Recovery Options window allows you to save some time.


22. You can now continue with the regular process of installing Windows Server 2008. Just make sure that when it comes time to select the partition to which you will install, you select the larger drive and not the 1.5GB Z: drive (Figure 10-5).

23. Windows Server 2008 includes BitLocker Drive Encryption as a feature, but it is not installed by default. To do so, open Server Manager.
24. Click the Features link.
25. Select BitLocker Drive Encryption, as shown in Figure 10-6, and then click Next.
26. Click Install to continue with the installation.
27. Restart the computer upon completion.

**TIP**  You can also install this feature quickly by running the following command at the command prompt:

```
ServerManagerCmd.exe -install BitLocker -restart
```
Initializing BitLocker

Installing BitLocker is only part of the equation. Until you initialize and enable it, it isn’t going to do anything for you. If you have a TPM-capable system, you will first want to initialize the TPM by running through the TPM Initialization from the BitLocker Control Panel applet. If you like to automate things through scripting, you’ll be happy to learn that TPM includes a management API that can be leveraged to initialize TPM programmatically as well. You must have local administrator privileges to initialize BitLocker and should always create a recovery password in the event that all other authentication methods fail and you need to get access to the drive. Once BitLocker has been initialized, non-administrative users can access the system as usual, with the added benefit of the behind-the-scenes encryption protecting their data.
Hands-On Exercise: Enabling BitLocker Drive Encryption

In this exercise, we will enable BitLocker Drive Encryption and encrypt the Windows installation volume.

1. Open Control Panel.
2. Double-click BitLocker Drive Encryption.
3. If you have not initialized your TPM yet, you will see the Initialize TPM Security Hardware Wizard. Simply follow the wizard and restart the computer. If you do not have a TPM module on your system, you will need to perform the following steps to allow you to enable BitLocker without a TPM:
   a. Choose Start | Run.
   b. Type `gpedit.msc` and press Enter.
   d. Double-click Control Panel Setup: Enable Advanced Startup Options, as shown in Figure 10-7.

Figure 10-7. Access the BitLocker Drive Encryption Group Policy item.
e. On the Properties page, select Enabled and make sure the Allow BitLocker without a Compatible TPM checkbox is checked. From the drop-down lists below this, you can select startup key and pin options of your choice for computers with a TPM; then click OK (Figure 10-8).

f. Close the local group policy editor, and then open a command prompt and run `gpupdate`.

**TIP** You can also make this change centrally if you have Active Directory by setting these preferences in a Group Policy object on your domain. Also, in the same policy template, you can enable BitLocker backup to Active Directory.

g. Go back into the BitLocker Drive Encryption Control Panel applet.

4. Click Turn On BitLocker, as shown in Figure 10-9.

5. Click Continue with BitLocker Drive Encryption when asked if you want to use BitLocker Drive Encryption, as shown in Figure 10-10.

![Control Panel Setup: Enable advanced startup options Properties](image)

**Figure 10-8.** Allowing BitLocker without a compatible TPM
Chapter 10: Windows DNS, BitLocker Drive Encryption, and Itanium Support

Figure 10-9. Turn on BitLocker.

Figure 10-10. Confirm that you want to encrypt the volume.
6. Select a BitLocker startup preference. You can choose Use BitLocker without Additional Keys, which uses TPM only authentication. You can then choose Require PIN at Every Startup or Require a Startup USB Key at Every Startup. The last option is your only choice if you do not have a compatible TPM on your system, as shown in Figure 10-11.

7. You will then be prompted to save the recovery password (Figure 10-12). Select the location that best suits your needs and click Next.

8. Make sure that the Run BitLocker System Check is enabled and click Continue, as shown in Figure 10-13.

9. Click Restart Now to begin the encryption process.

---

**Figure 10-11.** Configure BitLocker startup preferences.
Chapter 10: Windows DNS, BitLocker Drive Encryption, and Itanium Support

Figure 10-12. Specify the location at which to save the recovery password.

Figure 10-13. Verify that Run BitLocker System Check is enabled.
BitLocker Recovery

Whenever you talk about encryption, you must include a discussion about how to recover the data if the original protection unlocking mechanism doesn’t work. In the case of BitLocker, you might have required TPM plus PIN authentication. What if the user forgets the PIN or the PCM somehow malfunctions? From a BitLocker perspective, something has been compromised, and it will keep the data safely encrypted. The process of recovering data-protected volumes involves the use of a recovery key or password that gives administrators a back door into the system should something like this happen.

The following scenarios might trigger the need for a recovery to be performed:

- The user forgets the PIN and you don’t have a record of it anywhere else.
- The user has a damaged or missing USB flash drive containing the key.
- An error occurs in the TPM or the TPM is different.
- The TPM is disabled or cleared.
- Any of the early boot files are modified, thereby causing a signature mismatch with what’s stored in the TPM.

In these scenarios, your only choice is to go through the recovery steps. Since the detection of this state occurs even before Windows Server 2008 is allowed to load, you must either insert the USB flash drive containing the recovery key or enter the recovery password. The encrypted drives will not be readable until you have unlocked them using the recovery key or password.

NOTE When entering the recovery password, you must use function keys rather than the regular numbers on the keyboard. Numbers 1 through 9 are represented by F1 through F9, with F10 representing 0.

Hands-On Exercise: Recover Access to BitLocker Encrypted Volumes

In this exercise, we will regain access to the BitLocker encrypted volumes. The easiest way to simulate this on your system that is TPM enabled is to turn off TPM and restart the computer. Take the following steps when you have an inaccessible volume due to missing authentication requirements:

1. If the computer is turned off, turn it on. You will be presented with the BitLocker Drive Encryption Recovery Console.
2. If you have a USB flash drive containing the recovery password, insert that now and press esc. This will automatically enter the recovery password for you and restart the computer.

Or, if you do not have a USB flash drive with the recovery password but have the password available, press enter, and then type the recovery password using the function keys. Press enter to restart.
Turning Off or Uninstalling BitLocker Drive Encryption

At some point, and for various reasons, you’re probably going to want to disable BitLocker temporarily or completely disable BitLocker and decrypt all the encrypted drives. The most common reason why you’d want to disable BitLocker temporarily is to perform updates on the operating system or make changes to the TPM. Either of these actions would lock the drive and prevent access until a recovery key is entered. By temporarily disabling BitLocker, you can perform the changes and re-enable BitLocker after the next reboot so that the system can update the TPM with the new signature. Decrypting, on the other hand, removes BitLocker protection permanently.

Hands-On Exercise: Disable BitLocker Drive Encryption

This exercise assumes you already have a BitLocker-protected system with at least one encrypted drive.

1. Open BitLocker Drive Encryption from the Control Panel.
2. On the BitLocker Drive Encryption page, select the volume on which you want to shut off BitLocker.
3. When prompted for the level of decryption, select either Disable BitLocker Drive Encryption (if this is temporary) or Decrypt the Volume (which permanently removes BitLocker protection).

WINDOWS SERVER 2008 ITANIUM SUPPORT

Windows Server 2008 for Itanium processors is designed to function primarily as an application or database server, where scaling up in terms of local processing power and RAM is important. Many roles and features are not supported on an Itanium-based system when running Windows Server 2008. If you are currently running Windows Server 2003 for Itanium-based systems, most of your applications should transition over to Windows Server 2008 for Itanium-based systems.

Windows Server 2008 does not support Terminal Services in application mode on Itanium-based systems. It also doesn’t support a number of systems that are designed for distributed processing. If you are using an Itanium-based system, you will need to migrate some roles to servers running more conventional processors where those roles are fully supported. For example, SharePoint is not supported with Windows Server 2008 on Itanium-based systems. To see a complete list of roles that are supported and unsupported on Itanium-based systems, go to Microsoft’s Windows Server Web site. Windows Server 2008 for Itanium-based systems is a specialized platform that is purposely built to function as an application server. Due to this design constraint, many common features are not supported. This includes but is not limited to being able to run Media Player and use other multimedia tools, Bluetooth, wireless and IrDA, modems/TAPI (Telephony Application
Program Interface), and Windows Messenger. You probably wouldn’t be running these services anyway on a server unless it is fulfilling a very specific multimedia server role. However, the server does support some very critical optional components, such as cluster services, Microsoft Data Access Components (MDAC), network load balancing, storage area network (SAN), and Windows System Resource Manager. Although there is support for a 32-bit emulated environment, application functionality could potentially be limited by the core functionality that the server can support. It is recommended that you check with your application vendor before running an application on Itanium-based servers to ensure it will function correctly.

CHAPTER SUMMARY

This chapter covered some key changes to DNS and discussed the use of BitLocker Drive Encryption to help protect the system. Windows Server 2008 natively supports IPv6. As a result of this, Microsoft has had to enhance the functionality of DNS in Windows Server 2008 to be able to accommodate IPv6 entries. Local-link multicast name resolution (LLMNR) has also been created to address the need to resolve IP addresses of hosts on the local segment without the need of WINS or NetBIOS since neither of these two systems support IPv6. Furthermore, the introduction of a read-only domain controllers role in Windows Server 2008 has resulted in the creation of primary read-only DNS zones in the Windows DNS server system.

BitLocker provides an extra layer of security that can help keep your server secure. It does add a bit of complexity, especially when you are performing updates on the system that could alter any of the boot-related components; on the upside, it can prevent unauthorized access through physical access. If you have a server in a remote location where physical security cannot be guaranteed, the BitLocker solution gives you extra peace of mind. As much as possible, take advantage of multi-factor authentication, such as TPM plus PIN, so that in the unlikely event of a maliciously altered TPM, the PIN adds a secondary line of defense for the system.

Windows Server 2008 also continues to support Itanium-based systems. Since this is a highly specialized build of Windows Server 2008, it provides limited support for roles that are not directly related to the ability of the server to host applications. You should reserve the use of Windows Server 2008 for Itanium-based systems only for applications that require the performance and scale-up support that it offers.
Routing and Remote Access
When we talk about connectivity to systems beyond our local network, we start getting into the world of Routing and Remote Access Services (RRAS). The Routing portion of RRAS provides LAN-to-LAN, LAN-WAN, virtual private networks (VPNs), and Network Address Translation (NAT) routing services. The Remote Access portion provides remote connectivity to your LAN through dial-up or VPN access. With the proliferation of broadband, VPN is gaining greater popularity and can provide faster and more efficient remote connectivity at a lower cost than dial-up services that typically require specialized hardware such as modem banks to become useful. RRAS has existed since Windows NT 4.0, but has steadily evolved into RRAS, now an important part of Windows Server 2008. The implementation of RRAS that is included with Windows Server 2008 includes the ability to integrate seamlessly with Network Access Protection (NAP) to give administrators more granular control over the types of systems allowed to connect to a network remotely. Although many organizations use specialized hardware routers to provide routing services, RRAS also gives you the option of using a Windows server as a full-fledged router.

**ROUTING SERVICES**

Windows Server 2008 RRAS provides multiprotocol routing services for LAN-LAN, LAN-WAN, VPN, and NAT connections. To use the routing feature of RRAS, you need a solid understanding of network protocols. The ultimate goal, of course, is to have hosts on one network segment communicate with hosts on another segment—that is, internetwork communications.

Although Windows Server 2008 does provide routing services through RRAS, they really don’t compare to the power of dedicated router equipment. You may be wondering why this feature should even exist in the operating system, when practically any organization that uses routers would usually choose a dedicated router over a multi-homed Windows server. In some special circumstances, using Windows Server 2008 and RRAS may actually be your best option. After all, it costs nothing but an extra network interface. It’s a good option if you want to connect a small satellite office to your main office with minimal cost and you expect only a light load to be placed on the server as a result of its routing function. In reality, though, routing is probably the least used of the two primary features that RRAS can provide.

**Routing Basics**

Before going any further, you should understand how routing works. Although RRAS provides multiprotocol routing capabilities, this discussion will focus on TCP/IP since that is by far the most commonly used protocol on a Windows network.

Packets used in TCP/IP communication have source and destination addresses. A subnet mask is applied to each address to determine which part of the IP address is the network address and which part refers to the host. When a packet is being sent to its destination by a host, it first determines whether the destination address is on the same subnet.
If it is on the same subnet, the packet is simply sent out over the physical medium for the destination host to pick up. If it determines that the destination is part of a different network, it sends this packet to a router either defined by its routing table or, if no match is found, to its default gateway.

It is the router’s responsibility to examine packets being sent out to a different network to determine where they should be sent off to next. The router intelligently determines to which of its known interfaces it will send the packet to reach its final destination. This is dictated by routing tables in the router that define rules that govern how packets should be delivered, based on destination addresses. Some routers are configured with redundant links to the same destination. For example, the router might be connected to another network via a T1 line going to one Internet Service Provider (ISP) and another T1 going to a completely different ISP. The purpose for such a configuration is that if one connection fails, it has an alternative route it can take. Cost metrics are then associated with each of these lines so that the router can make intelligent decisions around which path to use for traffic. You can configure load balancing for traffic across both lines or configure one line as the primary that can switch over to the other line if the primary is unavailable.

Figure 11-1 shows two networks separated by two routers that are directly connected. The best way to understand this concept is to illustrate the process of how the packets get from one host to another. Use Figure 11-1 to follow along.

If Client A with the IP address 192.168.10.10/24 wants to send a packet to Client B with IP address 192.168.10.15/24, it first determines that they are both on the same network—namely 192.168.10.0. In this case, Client A puts the packet “on the wire” for Client B to pick up. Now change that scenario a bit, and say that Client A needs to send a packet to Server A
with the IP address 192.168.100.10/24. The server is on the network 192.168.100.0. The client is on the network 192.168.10.0. In order for the packet to get to the server, it needs to go to Router A on its 192.168.10.1/24 interface. It then sends the packet to Router B through its 192.168.15.1 interface. Router B then receives that data on its 192.168.15.5 interface and then forwards it over to Server A through its 192.168.100.1 interface.

That all seems logical enough; but the real question is how did Router A know to send the packet over to Router B and how did Client A know to send the packet to Router A in the first place? The answer is routing tables, the rules that dictate how packets should be sent based on the destination address. Workstations also have routing tables but are more generally defined. Workstations are usually configured to send any traffic not destined to the local subnet to its default gateway. If you look at Figure 11-1 again, that’s typically how Client A would know to send packets outside its segment to the default gateway, which we will assume is configured to Router A’s 192.168.10.1 interface.

When Router A receives the packet destined for 192.168.100.10/24, it determines that the destination is not a network that is on either of its interfaces (192.168.10.1 or 192.168.15.1). There are essentially three ways Router A can be configured to send the packet over to Router B. The first way is a default route, which is really the same as a default gateway. It will send any packet for networks it doesn’t know about to the default route—which in this case would be configured to Router B’s 192.168.15.5 interface. Another way would be through a static route. Router A could contain an entry that tells it that any packet destined for the 192.168.100.0 network should be forwarded over to Router B. Lastly, Router A can be configured for dynamic routes, in which case it uses a dynamic routing protocol to discover automatically that the 192.168.100.0 network is accessible through Router B.

**TIP** If you aren’t familiar with the notation of 192.168.10.10/24, it is merely a shortcut naming convention to say that the IP address 192.168.10.10 has a subnet mask in which the first 24 bits represent the network address. In standard notation for IPv4, the subnet mask would then be 255.255.255.0.

**Dynamic Routing**

When configured correctly, dynamic routing can reduce the administrative burden of managing routers. If your network is complex, dynamic routing practically eliminates or significantly reduces the amount of administrative overhead needed to maintain static route information. Dynamic routing works through a process called router discovery by which it automatically detects other routers in its neighborhood that are also configured for dynamic routing. They then share information with each other regarding other networks that are accessible through its interfaces, and using that information, each router builds a dynamic routing table. Dynamic routing has the added advantage of being able to evolve with your network as its topology changes. It can discover new routes as they become available and redefine its table if links start to go down. Windows Server 2008 supports Routing Information Protocol (RIP) version 2 as a dynamic routing protocol for IPv4.
Routing Information Protocol

RIP is a relatively easy-to-configure dynamic routing protocol. It works by configuring routers to broadcast its list of known networks. A router accepts these messages and adds a route to those networks in its own routing table. In RIPv1, these route announcements are done on a periodic basis regardless of any changes in its known networks. RIPv2 improves on RIP by multicasting as soon as any of its known routes change. Not only does this improve performance by minimizing the network traffic RIP generates, it also lets other routers update their routing information as soon as a change of route has been detected. It also supports clear text username and passwords for preventing unwanted changes to the routing table from unknown devices. The limitation to RIP is that it can go only as far as 15 hops.

RIP also operates in one of two modes: periodic update mode and auto-static update mode. In periodic update mode, updates are sent out on a periodic basis as defined by the administrator. This is the default setting for RIP. If a route is unavailable when this update occurs, that route is deleted from the routing table. Although that sounds ideal, it doesn’t work for dial-on-demand connections where one route is not always connected but rather initiated when needed. The route is still valid for that network segment, but since it’s not available all the time, the router will convey this information out to other routers, which will subsequently delete the entry for that route in their routing tables. To address this, RIP supports auto-static update mode, in which a router automatically converts route information it receives into static routes in the routing table. This way, the route persists forever until the administrator deletes the entry. As an added optimization, routers do not provide their routing table information until a router requests it.

Hands-On Exercise: Installing Routing and Remote Access

Before you can configure RRAS, you must install it. If you look at the various server roles and features you can install on Windows Server 2008, you will find that RRAS is missing. In fact, it isn’t missing at all: You simply have to select RRAS as part of Network Access Services to install it. Follow these steps to install RRAS:

1. Open Server Manager.
2. Click the Add Roles link to start the Add Roles Wizard.
3. Click Next on the Before You Begin screen.
4. Select Network Policy and Access Services on the Select Roles screen, and then click Next.
6. Select Routing and Remote Access Services on the Select Role Services screen (Figure 11-2). Then click Next.
7. Confirm the installation options and click Install.
8. Click Close when the installation completes.
Routing Configuration with RRAS

Now that you understand how routing works and have installed RRAS, you can move on to learning how routing is configured in Windows Server 2008 using RRAS. Before you begin, however, you need to make sure your server has at least two network interfaces. After all, without two different interfaces, there’s nothing to route.

Hands-On Exercise: Configuring and Enabling RRAS

In this exercise, we configure RRAS for LAN routing and enable the RRAS service.

1. Click Routing and Remote Access from the Administrative Tools Start menu item to open the Routing and Remote Access management console.

2. Right-click your server name and choose Configure and Enable Routing and Remote Access. This will open the Routing and Remote Access Server Setup Wizard.
3. Click Next on the Welcome screen.
4. Select Custom Configuration on the Configuration screen, as shown in Figure 11-3. Then click Next.
5. Select LAN Routing from the Custom Configuration screen (Figure 11-4). Then click Next.
6. Click Finish on the completion screen.
7. Select Start Service when prompted to start the RRAS service.

**Configuring Network Interfaces for Routing**

When you open the Routing and Remote Access management console and expand your computer name, you will see four nodes in the navigation tree: Network Interfaces, Remote Access Logging & Policies, IPv4, and IPv6, as shown in Figure 11-5. You can view
your available network interfaces by clicking Network Interfaces. Here you can connect, disconnect, enable, or disable any of your interfaces simply by right-clicking the interface and selecting the appropriate action. Since right now we’re more concerned about routing than remote access, let’s skip Remote Access Logging & Policies. The two most important items are the IPv4 and IPv6 menu items. These two protocols are supported natively by Windows Server 2008.

When you expand either protocol, you see two child items: General and Static Routes. The General option displays each of your interfaces again, except this time it shows the important pieces of information regarding that protocol on the network interface. It shows the interface name, type, IP address, incoming bytes, outgoing bytes, static filters, administrative status, and operational status (Figure 11-6). You can interrogate each network interface to find out additional information by right-clicking the interface and selecting the appropriate item. Items that can be displayed include TCP/IP information, address translations, IP addresses, IP routing tables, TCP connections, and UDP listener ports.

Additional properties about a network interface can be obtained by double-clicking the interface. The IPv4 interface properties are displayed across three tabs: General, Multicast Boundaries, and Multicast Heartbeat, as shown in Figure 11-7. The second two tabs refer to the multicasting properties of IPv4 and are used to configure its scope,
time to live (TTL), and heartbeat detection settings. The General tab contains the bulk of the information that is critical to managing your interface. These settings are described in Table 11-1.

**Routing Protocols**

Windows Server 2008 supports the following routing protocols:

- Dynamic Host Configuration Protocol (DHCP) Relay Agent
- DHCPv6 Relay Agent
- Internet Group Management Protocol (IGMP) Router and Proxy
- NAT
- RIPv2 for Internet Protocol
The DHCP relay agents facilitate DHCP request forwarding over the routers. They come in both standard IPv4 and the new IPv6 flavors. IGMP is used between routers to negotiate and manage multicast groups. NAT is typically used when you want to connect or hide a number of internal hosts from another network. For example, if you want to connect your computers to the Internet and only one valid external IP address is available, you can use NAT to translate all your internal IP addresses out through your external IP. RIPv2 is used for dynamic route discovery between routers, up to 15 routers deep.

**Hands-On Exercise: Installing and Configure RIPv2 for IP**

In this exercise, we install RIPv2 for IP and configure its various properties. We configure RIP to work only with RIPv2-compatible routers to increase its efficiency.
1. Open the Routing and Remote Access management console.
2. Expand your server name.
5. Select RIP Version 2 for Internet Protocol (Figure 11-8), and then click OK.
6. Right-click the newly create RIP icon under IPv4 and select New Interface.
7. Select Local Area Connection, and then click OK (Figure 11-9).
8. Select Periodic Update Mode as the Operation Mode.

Figure 11-7. IPv4 network interface properties
<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable IP Router Management</td>
<td>Toggles whether this interface participates in IP routing. It toggles the administrative status from up and down. Unless this checkbox is enabled, you cannot route through this interface.</td>
</tr>
<tr>
<td>Enable Router Discovery Advertisements</td>
<td>Defines whether the router enables automatic router discovery through router discovery messages. When checked, it enables the advertisement-related options.</td>
</tr>
<tr>
<td>Advertisement Lifetime (Minutes)</td>
<td>Indicates the number of minutes an advertisement is valid. Once the advertisement has expired, it is no longer accepted by the clients.</td>
</tr>
<tr>
<td>Level of Preference</td>
<td>Defines a numeric value to indicate this router’s level of preference. The higher the number, the more clients will prefer to use this router.</td>
</tr>
<tr>
<td>Send Out Advertisements Within This Interval</td>
<td>Defines the minimum and maximum time intervals when advertisements are sent. This values provides routers with a range of time when it can randomly select to advertise to avoid having all routers advertise at the same time.</td>
</tr>
<tr>
<td>Inbound Filters</td>
<td>Defines filters for which packets are allowed in through this interface. For example, you can block all traffic coming from a specific network and even restrict by protocol.</td>
</tr>
<tr>
<td>Outbound Filters</td>
<td>Works the same as Inbound Filters except applies to outbound packets through this interface.</td>
</tr>
<tr>
<td>Enable Fragmentation Checking</td>
<td>Allows the router to block any fragmented packets. This is useful to prevent denial-of-service attacks caused by fragmented packets. Use this with caution, however, because some applications work by using fragmented packets, and enabling this option will prevent that application from communicating.</td>
</tr>
</tbody>
</table>

Table 11-1. IPv4 Network Interface General Properties
10. Select RIP Version 2 Only as the Incoming Packet Protocol. The General tab should now look like Figure 11-10.

11. Click the Security tab and make sure that all routes are accepted for incoming routes and all routes are announced for the outgoing routes (Figure 11-11).

12. Click the Neighbors tab. Here you can define how this router will interact with other RIP routers. Select Use Broadcast or Multicast Only, as shown in Figure 11-12. If you want to provide specific routers with which RIP can communicate, you could enter their IP addresses here as its neighbors.

13. Click the Advanced tab, change the Periodic Announcement Interval to 60 seconds, and then click OK to save the changes (Figure 11-13).
A Closer Look at RIPv2 Properties

The preceding exercise covered installing and configuring RIPv2 for IP on Windows Server 2008 and covered only a handful of settings for RIPv2. Let’s look more closely at the different options you can set and what effect they have on the RIP behavior. Referring to Figure 11-10, you can change the following settings on the General tab:

▼ **Operation Mode**  Can be Periodic Update (default for LAN interfaces) or Auto-static Update (default for demand-dial interfaces). In general, you want auto-static whenever a link is not connected at all times but you want that route to stay valid on all your routers between updates even when the connection is down.
Outgoing Packet Protocol  Controls how RIP packets are sent out from the router. RIPv2 multicast is the most efficient since it uses multicasts rather than broadcasts to send updates to routers. RIPv1 or RIPv2 broadcast are general methods for broadcasting RIP packets. You should select RIPv1 Broadcast only if you need the compatibility with RIPv1 routers. Silent RIP can also be selected if you don’t want this router to send any RIP advertisements. This is useful if you want this router to listen for and take in route information from other routers but not share its routing table with anyone else.

Incoming Packet Protocol  Specifies whether to listen for either RIPv1, RIPv2, or both types of RIP announcements. You can also ignore all incoming packets if you choose not to take updates from any other router.

Added Cost for Routes  Adds an integral cost value to this interface. Be careful when using this, since if the value ends up becoming high, the route may not be used at all.
Tag for Announced Routes  Adds a tag for announced routes. This feature is not used by Windows Server 2008 but can be used by other routers.

Activate Authentication/Password  Configures a password that prevents any updates from occurring between routers unless they have matching passwords. This can be useful for preventing accidental updates between unknown routers. However, don’t think of this as a security feature since the password is sent out in clear text.

The Security tab helps protect what routes RRAS will accept. If yours is a well-known network (which it should be), you should restrict accepting or even advertising routes to specific networks that you know about. This can help prevent malicious attacks that rely on sending bogus route information to your routers to redirect your data elsewhere.
The following descriptions are offered for each of the properties listed on the Security tab shown Figure 11-11:

- **Action**   Defines whether the settings apply to incoming or outgoing routes.

- **Accept/Announce All Routes**   Specifies whether all incoming routes are accepted or all outgoing routes are announced.

- **Accept/Announce All Routes In the Ranges Listed**   Unlocks the From and To fields so that you can add IP ranges that you will allow for either incoming or outgoing route updates.

- **Ignore/Do Not Announce All Routes In the Ranges Listed**   The reverse of Accept/Announce All Routes In the Ranges Listed. The From and To fields now signify IP ranges you either want to ignore or not announce.
Although RIPv2 can use broadcasts or multicasts to send and receive information from any router that also supports RIP, you can make the updates more efficient by defining neighbors. In this model, you can specify exactly to which routers you want to send updated information. This reduces the amount of traffic being generated by RIP and allows you to control exactly who communicates with the router. The Neighbors tab (Figure 11-12) shows the various neighbor properties that are explained further here:

- **Use Broadcast or Multicast Only**  Restrictions RIP to use only broadcasts and multicasts for announcing route information. This is the default setting.

- **Use Neighbors In Addition to Broadcast or Multicast**  Allows you to define neighbors to which the router will announce routes directly but also still broadcasts or multicasts the route announcements.

- **Use Neighbors Instead of Broadcast or Multicast**  Restricts the router only to announce route information to the specified IP addresses.
The Advanced tab contains a number of other special properties of RIP that can be used to tweak its behavior. Refer to Figure 11-13 for the RIP Advanced tab and look at the following complete details about these properties:

- **Periodic Announcement Interval (Seconds)** The number of seconds between announcements.
- **Time Before Routes Expire (Seconds)** The amount of time before a route is marked as expired. If the router receives an update for a route, it resets the counter so this would affect only routes for which you don’t receive any updates.
- **Time Before Route Is Removed (Seconds)** The amount of time before expired routes are completely removed from the routing table.
- **Enable Split-Horizon Processing** Enabled by default to prevent routing loops since split-horizon processing prevents broadcasting about a specific route on the segment from where that route was learned.
- **Enable Poison-Reverse Processing** When used in conjunction with split-horizon (hence the dependency), prevents routing loops by broadcasting the route learned from a network as unreachable (metric 16).
- **Enable Triggered Updates** Keeps your routers up to date as quickly as possible by forcing an announcement as soon as any change in routes is detected.
- **Send Clean-Up Updates When Stopping** When enabled, RRAS announces that any route it is handling is unavailable so that any adjacent routers can update their routing tables with this change of state.
- **Process Host Routes In Received Announcements** By default, RRAS ignores any host-specific route information it receives in announcements. If you don’t want RRAS to ignore this, you can enable it by checking this option.
- **Include Host Routes In Sent Announcements** Defines whether any host-specific routes that are present in RRAS are sent in its announcements as well.
- **Process Default Routes In Received Announcements** Select if you want RRAS to take in any default route information it finds in received announcements.
- **Include Default Routes In Sent Announcements** Select if you want to include known default routes for this router in its announcements.
- **Disable Subnet Summarization** Applies only if you limit your outbound packets to RIPv2 (either broadcast or multicast). When checked, tells RRAS to send all route information for routers in other subnets rather than summarizing it in the form of a class-based network ID.
DHCP Relay Agent

DHCP Relay Agents are installed just as you install RIP, except that you select DHCP Relay Agent instead of RIP when selecting the routing protocol. The purpose of a DHCP Relay Agent is to allow hosts that are configured to acquire IP addresses using DHCP to obtain those from DHCP servers sitting in a completely different subnet. To specify to which DHCP servers the relay agent should forward requests, you simply right-click DHCP Relay Agent under the IPv4 section of your server’s RRAS configuration, select Properties, and enter the IP addresses for the DHCP servers, as shown in Figure 11-14. You can then create new Interfaces for the DHCP Relay Agent to enable the relay for a specific network interface.

Figure 11-15 shows the properties for a new DHCP Relay Agent interface I created with my Local Area Connection network interface. In this window, you can enable or disable the relay of DHCP packets as well as configure hop-count and boot thresholds. DHCPv6 Relay Agent is similar, except it forwards requests for IPv6 addresses instead.
Internet Group Management Protocol

Internet Group Management Protocol (IGMP) is another routing service you can install on RRAS. You install it as you would any routing protocol (refer to the earlier exercise where we install RIP). Once installed, you can attach it to various network interfaces. You can then enable or disable IGMP on a given interface, configure its mode (router or proxy), and specify its version. When configured in IGMP router mode, various options are available in the IGMP Properties Router tab (see Figure 11-16).

Network Address Translation

NAT is a nice added feature to RRAS in Windows Server 2008. NAT is used everywhere nowadays. In fact, almost every cable/DSL router works in NAT mode: The external interface is configured with a valid IP address for the Internet, and the hosts behind it are typically configured with private addresses, which are then NATed to the external address.
This allows multiple computers to share the same Internet connection. With Windows Server 2008, you can accomplish the same thing using RRAS. NAT in RRAS can even distribute IP addresses to whichever interface you designate as your private interface, so that connectivity can be established quickly and easily. Configuring the DHCP allocator can be done from the NAT Properties window under the Address Assignment tab, as shown in Figure 11-17.

**Hands-On Exercise: Installing and Configuring NAT**

In this exercise, we install the NAT routing protocol. We will configure one network interface as an external interface and another network interface as a private interface. We will allow connectivity from hosts communicating through the private interface to the external interface by NATing their addresses to the external IP address.
NOTE As with all routing services, you will need to make sure you have two network interfaces (excluding Internal or Loopback) for this to work.

1. Install the NAT routing protocol by following the exercise for installing RIPv2, except select NAT instead of RIPv2 for IP.

2. In the Routing and Remote Access management console, expand your server, then IPv4, and then right-click NAT and select New Interface.

3. Select one of your network interfaces. In this case, select Local Area Connection (Figure 11-18). Then click OK.

4. Select Private Interface Connected to Private Network, and then click OK (Figure 11-19).
Figure 11-18. Creating a new IPNAT interface

Figure 11-19. Configuring private network interface
5. Right-click NAT and select New Interface.

6. This time, select your other network interface. Here, it’s Local Area Connection 2 (Figure 11-20). Then click OK.

7. Select Public Interface Connected to the Internet as the Interface Type, and check the Enable NAT on This Interface checkbox, as shown in Figure 11-21.

8. Click the Address Pool tab, and then click Add. Enter the range of IP addresses your ISP provides (see Figure 11-22). NAT for RRAS requires that you know your external IP or range of IPs.

9. Click the Services and Ports tab. If you want to allow services on your private network to be available for Internet users, use this tab to create NAT port redirection rules for that service. For now, leave this blank and click OK to save the settings (see Figure 11-23).

**Static Routes**

Static routes are nothing more than hard-coded routes to various networks or hosts. You can define static routes for both the IPv4 and IPv6 protocols. Static routes are a good option for defining routes without having to resort to dynamic routing protocols. You can
apply metrics to each of the routes you create to define its relative cost. Routes can be created from the RRAS management console interface or from the command line. Adding, viewing, and modifying the routing table from the command prompt involves the use of the `route add` command. For example, to add a route to the 172.16.0.0 network with a subnet mask of 255.255.0.0 through the gateway with the IP address 192.168.10.254 and a metric of 1, you would use the following command:

```
route add 176.16.0.0 255.255.0.0 192.168.10.254 1 16
```

The last parameter of this command (16) refers to the number representing the network interface with which you want this route associated. On my server, my first local area network connection has an interface number of 16. The question is, of course, how
do you know what the interface number is in the first place? All you have to do is run the following command:

```
route print
```

This command outputs quite a bit of information, such as the interfaces that are on your server, along with any routes that have been defined for IPv4 and IPv6. Figure 11-24 shows the output of the `route print` command. The first section of the command’s output is an Interface List. The interface number is listed followed by the MAC address and then the interface description. You use the interface number from this command when you need to specify the interface for the `route add` command.

Although using the command prompt for adding routes can be helpful, it’s by far easier to use the RRAS management interface to define static routes. To create a new static route using the interface, simply expand the protocol (IPv4 or IPv6) in the RRAS
Figure 11-23. Configuring NAT port redirection rules

Figure 11-24. Output of the `route print` command
management interface, select New Static Route, and enter the relevant information. Figure 11-25 shows how you would fill out the New Static Route dialog box to add the same route information we added using the command prompt.

**REMOTE ACCESS**

Windows Server 2008 provides two different methods for remote access: dial-up and VPN. Dial-up obviously requires a modem or modem bank to allow remote connectivity using POTS (plain old telephone system) or ISDN. VPNs are designed to reduce the overall cost of remote access by leveraging the Internet to establish a secure tunnel for communicating between remote computers and the corporate private network. VPN use has proliferated as more and more users have gained quick and easy access to the Internet from practically anywhere. The best part about VPN technology is that all you need from the corporate network is a relatively fast Internet connection that can handle the load of connections generated by the remote users.

**Dial-Up Networking**

Dial-up networking (DUN) is one of the most traditional methods for providing remote access. It uses modems to connect directly to your corporate private network over telephone lines. DUN has the advantage of providing fairly secure end-to-end connections between a remote host and your private network. It is also easily accessible since you can
connect from wherever you want as long as you can get to a phone line. The problem is
the relatively slow speed of the modem connection, which can be overcome to a certain
extent using a process called multilinking, where connections across multiple phone or
ISDN lines can be virtually grouped together so they act as a single larger data pipe.

DUN is also costly to manage since you have to provide and pay for multiple lines to
allow these connections to be established. Naturally, the more users you have the more
lines you need from your phone provider. Your users also shoulder some of the cost since
they are charged by their phone company whenever they dial out. This cost can either
be offset or centralized by using callbacks, phone cards, or even toll-free, dial-up access
numbers.

**Point-to-Point Protocol**

Point-to-Point Protocol (PPP) is used for DUN. It allows hosts to communicate using
TCP/IP over serial links such as DUN or even serial cable connections. PPP actually
uses six different protocols, listed here in the order in which they are used to establish
communications:

- **Link Control Protocol (LCP)**  LCP is in charge of negotiating link parameters,
maintaining those links, and then terminating it when done. You can think of
LCP as acting within the physical layer of the network stack for PPP.

- **Challenge Handshake Authentication Protocol (CHAP)**  CHAP is in charge
of authenticating the client using login credentials to decide whether the user is
supposed to have access or not.

- **Callback Control Protocol (CBCP)**  This protocol manages callback, which
allows you to configure the server to hang up the connection and call the client
back to establish communications. This is used to centralize cost since the client
is connected only for a very brief period, after which the server calls the client
back and assumes the charges for the connection. It is also used for security.
If you know exactly where the client is supposed to be calling from, hanging
up and then reconnecting to the client ensures that connections can’t be
established from any other number.

- **Compression Control Protocol (CCP)**  As you would expect, this protocol
is in charge of negotiating compression parameters between server and
client. Although software compression is useful, in reality, you should rely on
hardware compression as it is faster and frees up CPU cycles.

- **IP Control Protocol (IPCP)**  This protocol is in charge of IP negotiation such
as maximum transmission unit (MTU).

- **Internet Protocol (IP)**  At this point, PPP simply acts like any TCP/IP
connection over a regular LAN connection with speed being the only
differentiator. IP packets are sent back and forth over the connection and any
protocol that can stack on top of this can work just as it would on any “regular”
network connection.
Virtual Private Networks
VPNs provide near ideal solutions for remote access needs. They are cheaper to implement and manage since you have to be concerned only about your network bandwidth to the Internet. The proliferation of reliable and fast broadband connections as well as widespread use of wireless Internet access has made VPN even more accessible than dial-up networking ever was. Although security risks are a concern since technically the packets are being transmitted through the Internet, where potentially anything can happen, these risks can be mitigated by creating secure and encrypted tunnels for the data to pass through. Your clients make use of whatever Internet connection they can get to and connect over the Internet to get to your VPN servers and gain access to the corporate private network, as shown in Figure 11-26. Since all the client needs is an Internet connection, they can connect from virtually anywhere Internet service is provided (as long as it isn’t blocked by the provider). This includes their homes, hotels, airports, and even wirelessly through wireless hotspots or even via wireless broadband cards.

Encapsulation and Tunneling
VPNs work by encapsulating regular data that you may want to send to the remote host into another protocol so that it can be safely and securely transmitted over the Internet. The best part is that your application won’t require any changes to make it work. All that’s needed is a VPN client and server that can encrypt and encapsulate a “regular” packet and then reverse this process on the other side so that the application can proceed to work as normal on the unencrypted data. Windows Server 2008 supports three different types of tunneling protocols: Point-to-Point Tunnel Protocol (PPTP), Layer 2 Tunneling Protocol (L2TP), and Secure Socket Tunneling Protocol (SSTP). Tunneling refers

Figure 11-26. Typical VPN
to creating a virtual connection between two networks over another network or set of networks, where the data being transmitted between the two networks participating in the tunnel cannot be deciphered by the intermediary network (such as the Internet).

PPTP is simple and easy to set up. Requests for connections are initiated, and then the server goes through a series of challenge and response questions with the client before attempting to authorize the user. Once the user is authorized, the tunnel is created and the session is encapsulated via the Generic Routing Encapsulation (GRE) protocol, which is simply a generic packet that states that its contents contain encapsulated data. This data is also typically encrypted using Microsoft Point-to-Point Encryption (MPPE). Data is then sent through this tunnel just as it would be on a regular private network, except the PPTP layer takes care of all the encryption and encapsulation work as well as reversing this process on the receiver end.

Layer 2 Tunneling Protocol/IP Security (L2TP/IPSec) is a more complicated tunneling protocol, but its complication is really born out of its ability to be more flexible. L2TP by nature is a very insecure tunneling protocol because it provides no encryption

**SSTP Connection in Detail**

SSTP is designed to make client/server VPN connections much easier with fewer complications, but it does not support (and is not designed for) creating site-to-site VPN connections. Those kinds of connections are best fulfilled using one of the other two tunneling protocols. When an SSTP connection is initiated, it undergoes a series of steps to establish the tunnel, establish authentication, and manage that connection through its lifetime. The following steps take place during this connection:

1. The SSTP client establishes a TCP connection to port 443 on the server.
2. The client indicates that it wants to establish a connection by sending an SSL Client-Hello message.
3. The server sends its computer certificate to the client.
4. The client validates the server certificate and generates an SSL session key that is encrypted using the public key of the SSTP server.
5. The client sends the SSL session key to the SSTP server.
6. The server extracts the SSL session key using its private key, and the SSL key is used for all future communication.
7. The client sends an HTTPS request to the server.
8. The client negotiates an SSTP tunnel with the server.
9. The client negotiates a PPP connection with the server, which also authenticates the user and configures IPv4 and IPv6 settings.
10. The communication between the server and the client is sent over the tunneled PPP link.
or authentication. This problem is resolved by pairing it up with IPSec to manage the security associations and encryptions for the channel. L2TP then takes advantage of this secure channel to establish a tunnel between the client and server. It no longer needs to worry about encryption or authentication since those have already been established by its partner, IPSec.

Secure Socket Tunneling Protocol (SSTP) is a new tunneling protocol introduced with Windows Server 2008. It is a very exciting new protocol because it was created specifically to address the issues PPTP and L2TP have when working through certain firewall configurations. An SSTP session is established using an HTTP over SSL (HTTPS) session between the server and the client. It reduces the cost for implementing VPN access because it simplifies your deployment. You can safely place your RRAS server behind NAT and you don’t need any third-party VPN software to establish connectivity. As with all SSL-based technology, you will need to have the root CA for the server’s computer certificate installed on the SSTP client for the connection to work. If you have your own CA, you have probably already distributed your root CA using Group Policy, or you can leverage a third-party CA such as VeriSign to sign your computer certificate.

**Hands-On Exercise: Configuring RRAS for Remote Access**

The little bit of information presented earlier is meant to be a general overview of how DUN and VPN work. A discussion on DUN and VPN technology and protocols could go on forever, but what you’ve read so far is all you need to know to configure remote access on Windows Server 2008.

In this exercise we install and configure RRAS for remote access. Your server should be part of a domain and a DCHP server must be on your network that can assign IP addresses to VPN clients.

**NOTE** If you are using the same server for this exercise that you used for the routing exercise earlier in the chapter, you must first disable RRAS in the RRAS management console by right-clicking the server name and selecting Disable Routing and Remote Access.

1. Log on to the server using an account that is a member of the Domain Administrators group.
2. Install the Routing and Remote Access role service if it is not already installed.
4. Right-click your server name and select Configure and Enable Routing and Remote Access. This will open the Routing and Remote Access Server Setup Wizard.
5. Click Next on the Welcome screen.
6. Select Remote Access (Dial-up or VPN), and then click Next (Figure 11-27).
7. In the Remote Access screen, check both VPN and Dial-up, since we are going to use both later (Figure 11-28). Then click Next.
Figure 11-27. Configuring RRAS for remote access

Figure 11-28. Specifying remote access connection methods
8. On the VPN Connection screen, select the network interface that is connected to the Internet, check the box to Enable Security on the Selected Interface By Setting Up Static Filters if it isn’t already checked (Figure 11-29), and then click Next.

9. On the IP Address Assignment screen, select Automatically to indicate how you want to assign IP addresses (Figure 11-30). Then click Next.

10. In the Managing Multiple Remote Access Server screen, select No, Use Routing and Remote Access to Authenticate Connection Requests since we will not be using RADIUS for the following exercises (Figure 11-31). Then click Next.

11. Review the setup summary, and then click Finish.

12. Click OK on the message box that pops up informing you that you must configure the properties of the DHCP Relay Agent with the IP address of the DCHP Server.
Figure 11-30. Specifying how IP addresses will be assigned

Figure 11-31. Specifying whether to use RADIUS
DHCP Integration with RRAS

Whether you use RRAS for DUN or VPN, the clients connecting will need to obtain IP addresses to communicate with other hosts on your network. Typically, you will want this handled by DHCP unless you have very specific requirements to use static IP addresses. Optionally, you can create a static address pool to provide dynamic IP address allocation to RRAS clients without using DHCP. If your RRAS server is also a DHCP server, your server will automatically be able to assign IP addresses. If you are running a separate DHCP server, you will have to configure the DHCP Relay Agent on the RRAS server to forward to the correct DHCP servers that will handle the RRAS connections.

Hands-On Exercise: Configuring DHCP Relay Agents for RRAS

If you don’t have DHCP server installed on your RRAS server, you will need to configure the DHCP Relay Agent so that it forwards the DHCP requests to the appropriate server. Windows Server 2008 supports both IPv4 and IPv6, so you can configure the DHCP Relay Agent for each of these protocols independently. For example, you may direct IPv4 DHCP clients to one DHCP server and have IPv6 addresses handled by another server. In this exercise, we configure the IPv4 DHCP Relay Agent to go to our DHCP server.

1. Open the Routing and Remote Access management console.
2. Expand the server name, and then expand IPv4.
3. Right-click DHCP Relay Agent and select Properties.
4. In the DHCP Relay Agent Properties window, add the IP addresses for your DHCP servers (Figure 11-32).
5. Click OK to save the changes.
6. Right-click DHCP Relay Agent and select New Interface.
7. Select the network interface that is connected to the subnet of the clients that need to have their DHCP requests relayed, and then click OK.
8. In the Internal Properties window, make sure Relay DHCP Packets is checked and adjust the hop-count and boot thresholds to appropriate values for your network (Figure 11-33). Then click OK.

Configuring RRAS Server Properties

At this point, your server is already capable of accepting inbound DUN connections, provided you have the appropriate hardware listening to your configured modem lines. The general options for RRAS are actually configured at the server level. If you open
Figure 11-32. DHCP Relay Agent Properties window

Figure 11-33. Internal Properties window
your server’s properties from within the RRAS management console, you’ll see six tabs for configuring your server properties. One of the most important is the General tab (Figure 11-34). On this property sheet, you can control whether the RRAS server will act as a router and enable or disable the Remote Access Server. This is useful if you need to disable remote access for any reason without having to disable Routing and Remote Access, which would wipe out your configuration.

The Security tab (Figure 11-35) contains options for configuring authentication providers for validating user credentials for remote access. This can either be your standard Windows Authentication or RADIUS for a more centralized remote access management solution. You can further refine your authentication methods by clicking the Authentication Methods button to access additional properties. As shown in Figure 11-36, you can configure Extensible Authentication Protocol (EAP) methods, MS-CHAP v2, CHAP, and PAP, or even allow remote systems to connect unauthenticated (however, you should never use this unless you are troubleshooting a situation for which you think your authentication method is causing problems).
Figure 11-35. RRAS Server Properties Security tab

Figure 11-36. RRAS Authentication Methods
The IPv4 and IPv6 tabs control IP forwarding for those protocols. The PPP tab lets you configure options related to PPP connections. You can permit multilink connections and enable dynamic bandwidth control. You can also configure whether the Link Control Protocol (LCP) extension or software compression will be enabled. By default, all these options are selected, as shown in Figure 11-37. Although these options give PPP greater flexibility and functionality, you should look at each of these options carefully and determine whether it is appropriate for your environment. For example, do you really want to give your users the ability to tie up multiple lines by supporting multilink connections? If you have hardware compression enabled, does it make sense to still use software compression?

The last tab is the Logging tab (Figure 11-38). By default, RRAS logs both errors and warnings. These logs are stored in the %WINDIR%\logs folder. You can select to log errors only, errors and warnings, all events, or no events at all. Another option allows you to log extended information that could help for debugging. This option should be used sparingly, since it does increase the log size significantly. The default should be adequate for most of your needs. If you want to reduce logging, at the very least you should log errors only so that you have something to fall back on when errors are encountered.
Hands-On Exercise: Configuring VPN Using PPTP

Windows Server 2008 supports PPTP, L2TP/IPSec, and SSTP. If you followed the exercise for configuring Remote Access, your server is now set up to receive incoming VPN connections. You server is configured by default to listen to a number of VPN ports for PPTP, L2TP, and SSTP. Although theoretical limits exist on the maximum connections your server can support, realistically your server hardware will limit the actual number of VPN clients it can support. The default is set to 128 connections each for PPTP, L2TP, and SSTP. In this exercise, we increase the number of SSTP connections to 256 and disable PPTP and L2TP for remote access.

1. Open the Routing and Remote Access management console.
2. Expand your server name node and select Ports. Notice how all the available ports are listed in the Ports details pane on the right (Figure 11-39).
3. Right-click Ports and select Properties.
Chapter 11: Routing and Remote Access

4. You’ll see a list of all the devices that RRAS is using. We are interested only in WAN Miniport (PPTP), WAN Miniport (L2TP), and WAN Miniport (SSTP) (Figure 11-40).

5. Double-click WAN Miniport (SSTP).

6. Make sure that Remote Access Connections (Inbound Only) is checked and leave the Phone Number for This Device field blank since we won’t be using SSTP over DUN. Change the number of Maximum Ports to 256, and then click OK (Figure 11-41).

7. Double-click WAN Miniport (PPTP).
8. Uncheck all checkboxes, and then click OK (Figure 11-42). Unchecking the Remote Access Connections (Inbound Only) checkbox effectively disables PPTP.


10. Uncheck all checkboxes, and then click OK.

11. Click OK on the Ports Properties dialog box to save the changes.

**NOTE** When you later configure your VPN client to connect to your RRAS server, you will need to make sure that the user account being used has the appropriate rights to connect. This can be done either using a Remote Access Policy or by explicitly allowing or denying remote access through the Dial-in tab of the user account properties in Active Directory Users and Computers (Figure 11-43).
CHAPTER SUMMARY

Routing and Remote Access are great ways to let your users access your network from virtually anywhere at any time. In fact, in most organizations, VPN access has evolved into critical parts of their infrastructure, as users demand and even expect the ability to have access to their systems around the clock. Although dial-up networking using modems and ISDN lines is certainly still an option, it has become less and less popular due to the proliferation and general availability of high-speed Internet connections from practically anywhere. VPN connections are more cost-effective and offer a better-performing user experience than DUN could ever dream of, even when taking advantage of multilink to pool multiple DUN connections together to achieve a higher data transfer rate. SSTP is one of the most exciting new features in RRAS that can significantly reduce the cost for implementing and supporting VPN access.

![User Account Dial-in tab set to allow remote access](image)
RRAS in Windows Server 2008 isn’t that different from what was available in Windows Server 2003, other than the support for IPv6 and SSTP. Although RRAS does provide the ability for the server to provide routing services, you should only do so in situations where implementing a true dedicated hardware router is either impractical, cost-prohibitive, or both. While using RRAS makes it easy to provide remote access functionality to your users, you should carefully review and test each and every RRAS option to ensure that you don’t sacrifice the security of your network for the functionality that RRAS provides. RRAS servers are typically deployed outside the trusted network boundary and are exposed to the Internet. This means they are prime targets for malicious users attempting to gain access to your protected private network. Take every precaution to harden your server and ideally place the RRAS server in your DMZ behind hardware firewalls to minimize exposure.
This page intentionally left blank
CHAPTER 12

Enterprise Public Key Infrastructure
Communication security is achieved only when the intended recipient(s) receive and read the intended communication. In the physical world, this could mean whispering into someone’s ear or holding a closed-door meeting. This physical closeness minimizes the risk of private messages being intercepted in transit. If you need to communicate securely over larger distances, you can use a courier to transport your message, or you can use the digital age equivalent: e-mail. The problem, however, is that you have no real control over who reads that message while it’s en route. Messages can easily be intercepted and read. The solution is to perform some kind of encryption on the message so that even if it’s intercepted, the message cannot be read unless it can first be deciphered.

Encryption can be performed in many ways. The easiest way is to transform the message to be sent using some kind of key, and then, using the same key, to reverse the process on the recipient’s end to decode the information. Without the key, the message is relatively secure, since nobody can read it without resorting to some form of brute force key attack that could take months, if not years, to yield useful results.

A public key infrastructure (PKI) encrypts and decrypts data using digital keys that are applied to data to generate ciphertext (the encrypted form of the data) that can then be freely transmitted anywhere in the world. You need not be concerned about potential interceptions, because the data is useless without the appropriate key to convert the ciphertext back to plain text (unencrypted data). PKI actually uses two keys to encrypt and decrypt data: a public key and a private key. The public key can be made available to anyone and can be used to encrypt data. However, the data can be decrypted only by someone who has the private key.

Throughout this book, we’ve discussed a number of services that utilize digital certificates to perform encryption and authentication services. Certificates are simply keys generated by the PKI system, in this case a certification authority (CA). Whether you realize it or not, you interact with PKI on a daily basis. For example, when you make an online purchase, you are typically redirected to a section of the seller’s Web site that is secured using Secured Socket Layer (SSL), which uses digital certificates generated by a PKI system to guarantee the server’s identity and to establish a secure encrypted connection between the client and server. PKI is all about providing encryption and identity management services through the use of private and public keys to encrypt and decrypt data. An enterprise PKI system allows you to centralize all aspects of key and certificate management from generating, issuing, and even revoking keys and certificates.

**PKI USES**

PKI can be used for any number of applications that support the technology. Typically, all the services that utilize PKI do so because as long as the private keys are kept secure, PKI is by far one of the most secure methods for encrypting and digitally signing data. PKI comprises multiple elements that can be used for different purposes. The most important of these elements is the CA, which manages the certificates over which it has scope.
Chapter 12: Enterprise Public Key Infrastructure

Essentially, you can think of a certificate as a public key for your PKI. You can and sometimes must use PKI in the following scenarios:

- Digitally sign e-mail to certify authenticity of its origin.
- Encrypt e-mail so it can be viewed only by intended recipients.
- Allow computers to communicate securely using certificates even over an insecure network such as the Internet (IPSec).
- Secure Web site traffic using SSL and certificates, essential for e-commerce.
- Verify the authenticity of software (including device drivers) using signed publisher certificates.
- Support authentication via certificates loaded on smart cards.
- Authenticate network connections using 802.1x.
- Encrypt user files, as used by the Encrypted File System (EFS).

PKI can be used to facilitate secure communications or to validate an identity in many more situations as well. In general, PKI is an ideal solution when you need very secure communications. The security of PKI relies on the security of the private keys and the number of bits used for encryption, however. For example, using 128-bit encryption may be a bit weak by today’s standards, so you might want to use 4096-bit or higher encryption to make it that much more difficult to crack. The downside, however, is that the higher the encryption, the more time it will take to encrypt and decrypt the data.

**Digital Signatures**

Digital signatures don’t prevent data from being read. Instead, encryption is used to digitally tag the message to guarantee its authenticity. Most people think digital signatures act like regular signatures—that is, the signature is the same whenever it’s used and forms a basis for comparison. The reality, however, is that when you apply your digital signature on some data such as an e-mail message, that signature is different each and every time you use it.

Digital signatures work by passing the data that you want to sign through a hashing algorithm that is used to generate a message digest. This digest is then encrypted using the sender’s private key, which in turn generates the digital signature. The message is then sent to the recipient either with the public key attached or relying on the recipient having the public key readily available. The recipient decrypts the signature using the sender’s public key. If the digital signature can be decrypted successfully using the sender’s public key, it ensures that the message came from the sender and was not tampered with in any way during transit. This feature is especially important for today’s software industry, for which digitally signed software is required to certify its authenticity and curb software pirating.

**IMPORTANT** A digital signature’s only purpose is to ensure that a message is authentic and was in fact generated by the sender. It doesn’t hide the data and therefore does not guarantee confidentiality.
DIGITAL CERTIFICATES

Digital certificates are no more than public keys encapsulated in a format that contains additional metadata regarding use and origin. Typically, the digital certificate contains not only the public key but also the name of the certificate’s owner and its CA. Although, technically, anyone can generate a digital certificate, without the weight of its trusted CA behind it, it’s practically useless. For example, if you self-generate a certificate for use on your SSL-enabled Web site, clients that do not have your CA registered in their trusted root certificate store will either be prompted that the certificate is from an untrusted source or perhaps denied the connection outright. Your digital certificate is the virtual equivalent of a passport or other form of identification that confirms your identity.

CERTIFICATION AUTHORITIES

The CA is the most critical component of PKI. Without CAs, there would be no digital certificates, and without digital certificates or public keys, there would be no digital signatures. The CA controls all aspects of certificate management. It is in charge of creating and then issuing the certificates to authorized users and computers. If a certificate has been compromised, you can revoke it at the CA and it will be added to the Certificate Revocation List (CRL).

A CA is nothing more than a certificate-generating entity. What prevents anyone from generating certificates haphazardly and doing whatever they want with them? Nothing! Anyone can set up a CA to work completely alone and issue certificates, and this is perfectly fine for certain applications. In the real world, certificates are used to interact with entities across company boundaries. You and another company can either add each other’s root certificates to your list of trusted certificates, or you can configure your CA to be part of a larger hierarchy of CAs that implicitly trust one another. For example, you can have your CA’s root certificate cosigned by a trusted commercial CA that can vouch for your identity.

Large commercial CAs are responsible for verifying the identity of the person or entity that is either applying for one of the CA’s certificates or seeking the ability to issue their own certificates. Companies such as VeriSign perform these types of verification and signing services. Essentially, they perform digital notarization of an individual’s or company’s credentials by issuing a certificate that is signed by their CAs. If a CA becomes authorized to issue certificates as part of the certificate hierarchy, the CA will be issued a certificate of its own that is signed by the commercial CA that is automatically trusted by Windows operating systems.

Any certificate then issued will be implicitly cosigned by the parent CA, its parent, and so forth. At the top of the certificate chain is always a root CA. Since the root CA has no additional parent and is implicitly trusted by all its child CAs, it is typically held under heavy physical security and disconnected from the network to prevent any possibility of being remotely compromised. VeriSign’s root CAs usually fall within this category. If you are creating your own internal CA hierarchy, you should consider heavily securing your root CA using the same precautions.
Chapter 12: Enterprise Public Key Infrastructure

TYPES OF CAs

The Certificate Service that is part of Windows Server 2008 supports two different types of CA configurations: Enterprise and Stand-alone. Both configurations can issue certificates. The difference is in their dependencies, the types of certificates they can issue, and to what extent they can be used. Each of these CA types can be used to create a certificate hierarchy comprising root CAs and subordinate CAs. The creation of subordinates within an organization is typically used to delegate certificate management to smaller groups, where they can be more closely managed. For example, if you are managing a global organization, you might use subordinate CAs to manage and control certificates issued for each country in which your company has a presence.

Enterprise CAs

As you would expect, an Enterprise CA installation type requires Active Directory Domain Services (AD DS) to be in place. Enterprise CA is designed to manually or automatically issue certificates to users, computers, and even child CAs. You must be an Enterprise Administrator to install an Enterprise CA in your environment, which requires or uses the following technologies:

▼ Active Directory
■ Group Policy to propagate certificates to client root certification authority stores
■ Authentication to the domain using smart cards loaded with appropriate user certificates
▲ Enterprise Exit Module used to manage how certificates are handled after they are issued

Since an Enterprise CA is heavily integrated with AD, it has the added advantage of being able to authenticate the user automatically with AD before issuing the appropriate certificate based on whatever template the user is requesting. In addition, metadata typically associated with certificates, such as name and contact information, can be pre-populated using data obtained from AD. Finally, by default, Enterprise CAs either accept or reject requests for certificates since they can quickly look up the criteria required to complete the request. The CAs don’t need to put the request in a pending state unless you explicitly configure them to do so.

Stand-alone CAs

While Enterprise CAs are heavily focused on providing certificate services for inside the organization and being able to cater to the automatic issuance of certificates, a Stand-alone CA is typically deployed to issue certificates to outside entities. Stand-alone CAs do not require AD since, for the most part, the CAs will be processing requests for individuals or systems that are outside of your management scope. All incoming certificate requests to a Stand-alone CA are marked as pending until such time that an administrator can verify
the information and make the appropriate decision either to approve or reject the request. Since there is no integration with AD, the generated certificate must be distributed manually and loaded onto the user’s certificate store. Certificates issued by Stand-alone CAs cannot be used to authenticate and log on to your systems using smart cards.

**CRYPTOGRAPHIC SERVICE PROVIDERS**

Cryptographic service providers (CSPs) are a set of hardware or software components used to implement a specific cryptographic function. For example, you might have a CSP that knows how to digitally sign e-mail messages or authenticate your wireless LAN using 802.1x. Out of the box, Windows Server 2008 (in fact all Windows operating systems) includes a predefined set of commonly used CSPs. Additional CSPs can be loaded at any time to support cryptographic methods.

**CERTIFICATE TEMPLATES**

Certificate templates are a set of rules and settings that govern certificates and form the basis for new certificates. For example, the template can be defined to allow the certificate to be used only for IPSec communications or only for signing e-mails. You will also need to define enrollment parameters such as whether automatic enrollment or web enrollment will be allowed.

**IMPORTANT** You must carefully design certificate templates before they are deployed, including considering a number of design options, such as enrollment parameters, ahead of time. Although certificate templates can be modified after their creation, doing so may result in you having to reissue updated certificates to replace old certificates that have already been issued using the prior template.

A subject name is associated with each certificate using the template. The subject name defines the holder of the private key. This can be a user, computer, program, or any other object that can participate in certificate management. You must determine how the subject name will be defined. Will the subject name be automatically populated using Active Directory or will the data be entered manually by a user via web-based enrollment? What the subject should be is really application dependent. For example, when used to sign computer certificates, the subject name might be the fully qualified domain name of the computer it is issued to.

You must also decide how many certificates each subject will get. Do you want each subject to have many individual and specialized certificates used for each different function, or do you want fewer, more generalized certificates that are multipurpose in use? Multipurpose certificates may sound like a great idea, but they can reduce your ability to control the specific uses of the certificates. Each template must be associated with an appropriate CSP.
Chapter 12: Enterprise Public Key Infrastructure

Since PKI relies on CSPs to perform the actual cryptographic function, your selection of the most appropriate CSP for your organization is an important decision. You must also decide on the length of the key used by the CSP for its cryptographic function. The longer the key length, the greater its security—but the trade-off is time. A long key will take additional processing time to use. If the key CSP is heavily used—for example, if it is used for securing network traffic using IPSec—the added processing time could severely decrease throughput. If you intend to use smart cards, for example, the template needs to be associated with the specific CSP for that smart card. If you assign the wrong CSP, the smart card will not work.

**NOTE** For better security, your certificates should not last forever. They should be set to expire so you can renew the certificates when appropriate to decrease the chances of their being compromised. You must balance the certificate life span so that it doesn’t become an administrative burden, while making the life span short enough to minimize the risk of compromise.

Templates also define key usage that restricts how a certificate can be used. For example, you may not want certificates designed for signing data to be used for encryption because you don’t want to have your data encryption public key to be generally available like your general purpose signing key.

When you install a CA, a number of default certificate templates are installed, as shown in Table 12-1. Table 12-1 defines the most common certificate types your server will need to handle.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Key Usage</th>
<th>Subject</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator</td>
<td>Sign and authenticate</td>
<td>Signature and Encryption</td>
<td>User</td>
</tr>
<tr>
<td>Authenticated</td>
<td>Sign operations for authenticating to a Web server</td>
<td>Signature</td>
<td>User</td>
</tr>
<tr>
<td>Authenticated</td>
<td></td>
<td>Encryption</td>
<td>User</td>
</tr>
<tr>
<td>Session</td>
<td></td>
<td>Encryption</td>
<td>Computer</td>
</tr>
<tr>
<td>Basic EFS</td>
<td>Encrypt data on EFS</td>
<td>Encryption</td>
<td>User</td>
</tr>
<tr>
<td>CA Exchange</td>
<td>Key storage for keys marked for private key archival</td>
<td>Encryption</td>
<td>Computer</td>
</tr>
<tr>
<td>CEP Encryption</td>
<td>Ability for holder to act as a registration authority for certificate enrollment protocol (CEP) requests</td>
<td>Encryption</td>
<td>Computer</td>
</tr>
<tr>
<td>Code Signing</td>
<td>Digitally sign code</td>
<td>Signature</td>
<td>User</td>
</tr>
<tr>
<td>Computer</td>
<td>Authenticate computer to the network</td>
<td>Signature and Encryption</td>
<td>Computer</td>
</tr>
</tbody>
</table>

**Table 12-1. Default Certificate Templates**
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Key Usage</th>
<th>Subject</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-Certification Authority</td>
<td>Cross-certify and qualify subordination</td>
<td>Signature</td>
<td>CrossCA</td>
</tr>
<tr>
<td>Directory E-mail Replication</td>
<td>E-mail replication within AD</td>
<td>Signature and</td>
<td>DirEmailRep</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encryption</td>
<td></td>
</tr>
<tr>
<td>Domain Controller</td>
<td>Certificate for domain controllers</td>
<td>Signature and</td>
<td>DirEmailRep</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encryption</td>
<td></td>
</tr>
<tr>
<td>Domain Controller Authentication</td>
<td>Authenticate AD users and computers</td>
<td>Signature and</td>
<td>Computer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encryption</td>
<td></td>
</tr>
<tr>
<td>EFS Recovery Agent</td>
<td>Decrypt files previously encrypted with EFS</td>
<td>Encryption</td>
<td>User</td>
</tr>
<tr>
<td>Enrollment Agent</td>
<td>Request certificate on behalf of another subject</td>
<td>Signature</td>
<td>User/Computer</td>
</tr>
<tr>
<td>Exchange Enrollment Agent</td>
<td>Request certificate on behalf of another subject and by supplying the subject name in the request; used for offline requests</td>
<td>Signature</td>
<td>User</td>
</tr>
<tr>
<td>Exchange Signature Only</td>
<td>Issue certificates for digitally signing e-mail; used by MS Exchange Key Management Service</td>
<td>Signature</td>
<td>User</td>
</tr>
<tr>
<td>Exchange User</td>
<td>Issue certificates for encrypting e-mail; used by MS Exchange Key Management Service</td>
<td>Encryption</td>
<td>User</td>
</tr>
<tr>
<td>IPSec</td>
<td>Digitally sign, encrypt, and decrypt network traffic</td>
<td>Signature and</td>
<td>Computer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encryption</td>
<td></td>
</tr>
<tr>
<td>Key Recovery Agent</td>
<td>Recover archive private keys</td>
<td>Encryption</td>
<td>KRA</td>
</tr>
<tr>
<td>RAS and IAS Server</td>
<td>Remote Access Service (RAS) and Internet Authentication Service (IAS) server identity authentication</td>
<td>Signature and</td>
<td>Computer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encryption</td>
<td></td>
</tr>
<tr>
<td>Root CA</td>
<td>Prove identity of the root CA</td>
<td>Signature</td>
<td>CA</td>
</tr>
</tbody>
</table>

Table 12-1. Default Certificate Templates (Continued)
Chapter 12: Enterprise Public Key Infrastructure

RECOVERY KEYS

Many organizations are concerned about what would happen if the key required to decrypt the data is lost. For example, if the head of HR encrypts all her files using EFS and then loses her key, how would the organization regain access to that data? The solution is to use recovery keys. Recovery keys are implemented as special-purpose certificates that can be used by recovery agents to decrypt data. Recovery agents are users who can recover data using recovery keys. Although recovery keys do allow decryption of data, they typically cannot be used to regenerate the original keys for encrypting that data. This is important, because it means that although a recovery key can be used to recover data, it can’t be used to recover signing keys, nor can it be used to impersonate someone else for the purpose of encrypting data. This satisfies the need to secure the integrity of the user’s identity. By default, the Administrator account is designated as the recovery agent for the CA. You can also delegate this authority to other accounts as desired.

Hands-On Exercise: Installing AD Certificate Services

Enterprise PKI for Windows 2008 refers to Active Directory Certificate Services, the role service you can install on your Windows Server 2008 server that allows your server to function as a CA.

In this exercise, we will install the AD Certificate Services role on a server. But before we install this role, we need to make a few decisions: Will this be an Enterprise CA or

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Key Usage</th>
<th>Subject</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smart card Logon</td>
<td>Authenticate using smart cards</td>
<td>Signature and Encryption</td>
<td>User</td>
</tr>
<tr>
<td>Subordinate CA User</td>
<td>Prove identity of the root CA for the subordinate</td>
<td>Signature</td>
<td>CA</td>
</tr>
<tr>
<td>Trust List Signing</td>
<td>Digitally sign certificate trust lists, authenticate, e-mail sign, and encrypt, and EFS</td>
<td>Signature</td>
<td>User</td>
</tr>
<tr>
<td>Web Server</td>
<td>Prove identity of Web servers</td>
<td>Signature and Encryption</td>
<td>Computer</td>
</tr>
<tr>
<td>Workstation Authentication</td>
<td>Authenticate workstation to servers</td>
<td>Signature and Encryption</td>
<td>Computer</td>
</tr>
</tbody>
</table>

Table 12-1. Default Certificate Templates (Continued)
Microsoft Windows Server 2008 Administration

a Stand-alone CA? Will we allow certificates to be requested through a Web site? For this exercise, we will install and configure AD Certificate Services to be an Enterprise CA. We will also enable certificates to be requested through a Web site.

**NOTE** The server on which you are installing AD Certificate Services must be a member of a domain, and you must perform the installation with a user account that has permissions to add the CA as the enterprise root CA.

1. Open Server Manager.
2. Click Add Roles to open the Add Roles Wizard.
3. Click Next on the Before You Begin screen.
4. Select Active Directory Certificate Services, and then click Next (Figure 12-1).

![Add Roles Wizard](image)

**Figure 12-1.** Selecting the Active Directory Certificate Services role
6. Select Certification Authority and Certification Authority Web Enrollment role services, and then click Next. Click Add Required Role Services when prompted.

7. Select Enterprise as the setup type, and then click Next. Select Root CA as the CA type, and then click Next.

8. Select Create a New Private Key to set up a private key, and then click Next.

9. By default, the CSP selected for the CA is RSA#Microsoft Software Key Storage Provider. Leave that in the Select a Cryptographic Service Provider field and ensure that the Key Character Length is set to 2048 bits. Select the sha1 hash algorithm (Figure 12-2). Then click Next.

10. Enter the Common Name for This CA. By default, this field is set to DOMAIN-SERVER-CA. The Distinguished Name Suffix should be set to the distinguished name for your domain. Leave these at the default values for now and click Next (Figure 12-3).

![Figure 12-2. Configuring cryptography for the CA](image)
Note that you cannot change the identity of your CA after it is installed, so make sure this information is exactly what you want before proceeding.

11. By default, the validity period for this root certificate is set to 5 years. This is fine for our purposes, but when you’re installing AD Certificate Services, you should consider how long a certificate should be for your environment. Leave it set at 5 years for now, and then click Next (Figure 12-4).

12. Set the Certificate Database Location and Certificate Database Log Location. By default, both are located in %WINDIR%\System32\CertLog (Figure 12-5). Click Next.

14. Click Next on the Role Services screen.

15. Confirm the Installation Options and click Install to continue. Click Close when the installation completes.

**CERTIFICATION AUTHORITY MANAGEMENT CONSOLE**

You manage your CA by using the Certification Authority MMC snap-in located in the Administrator Tools Start menu item. When you expand your CA server in the management console, you will see five folders that help you manage templates, requests, and certificates, as shown in Figure 12-6. The Revoked Certificates and Issued Certificates
folders contain revoked and issued certificates by the server. The Pending Requests folder contains any certificate requests that require manual approval (typically used when the server is configured as a Stand-alone CA). Failed Requests includes all requests for certificates that have failed. The Certificate Templates folder contains templates for all the different kinds of certificates this CA can issue.

Each CA server has its own set of properties that you can configure. To access these properties, right-click the CA server name and select Properties. You’ll see 10 tabs used to display the configuration of your server and in many cases to allow you to change various aspects of its behavior.

The General tab (Figure 12-7) shows the CA certificates assigned to your server. If this is a clean install, you will see only one certificate—the certificate you generated during the installation. If you renewed or created new certificates since it was first installed, a list of those certificates will be displayed, including the provider and hash algorithm used by your certificate.
The Policy Module tab displays the active policy module being used by the server. In this case, we installed only the Windows default policy module, so that module is displayed. For the Windows default policy, if you click the Properties tab, you can configure how your server will handle requests (Figure 12-8). You can allow it to use whatever settings you’ve configured for your certificate template (selected by default) or you can set all certificate request statuses to pending, which means someone will have to manually approve each certificate that is generated.

Exit modules are used to create procedures for what occurs after a certificate is issued. The Windows default exit module is typically used, which can be configured to publish new certificates to Active Directory. You can also publish new certificates to the file system by opening the Properties window of the Windows default exit module and checking the Allow Certificates to Be Published to the File System checkbox (Figure 12-9). These certificates get stored in %SYSTEMROOT%\system32\certsrv\certenroll.

The Enrollment Agents tab (Figure 12-10) contains options for configuring which accounts can act as enrollment agents and which certificate templates can be applied. The default is not to restrict enrollment agents, but if you need to limit who and what gets access, this is the place to do it.
Figure 12-7. General tab

Figure 12-8. Windows default policy module request handling
Figure 12-9. Enabling certificates to be published to the file system

Figure 12-10. Enrollment Agents tab
NOTE  Restricting enrollment agents can be enforced only by servers running Windows Server 2008.

The Auditing tab lets you configure what CA events get logged to the security event log (Figure 12-11). As with all types of auditing, you should select events that are meaningful to you so you can trace what happened without cluttering up your log with events you don’t really need.

The Recovery Agents tab (Figure 12-12) gives you access to data encrypted using a certificate without having the original key. If you have recovery agent certificates configured on your server, you can use this tab to archive keys for certificate templates that request to do so and allow those recovery agents to gain access to those keys.

Security permissions around the CA store can be configured in the Security tab (Figure 12-13). Four different permissions can be allowed or denied:

- **Read**  Lets you view certificates within the store.
- **Issue and Manage Certificates**  Allows you to issue, revoke, and manage certificates within the store.
- **Manage CA**  Covers all CA management-related tasks not directly relating to issuing and managing certificates.
- **Request Certificates**  Lets you request a new certificate. This permission can apply to both user and computer accounts and of course security groups containing either object type.

![Figure 12-11. Auditing tab](image-url)
Chapter 12: Enterprise Public Key Infrastructure

Figure 12-12. Recovery Agents tab

Figure 12-13. Security tab
The Extensions tab (Figure 12-14) lets you configure locations of various CA extensions, such as the CRL Distribution Point.

The Storage tab (Figure 12-15) displays where the certificate database and request log are located. You can’t change either of these values, but this information is provided so you can easily locate them. The Active Directory checkbox is checked and grayed out if you have an Enterprise CA, since you have no choice but to keep the configuration data in Active Directory. On the other hand, if you have a Stand-alone CA server that is a member of an Active Directory domain, you can optionally check this box to store its configuration in AD as well.

The Certificate Managers tab (Figure 12-16) can be used to create additional restrictions for the users specified in the Security tab for managing certificates. By default, all certificate managers are unrestricted, but you can restrict certificate managers to certain certificate templates here.
Chapter 12: Enterprise Public Key Infrastructure

Figure 12-15. Storage tab

Figure 12-16. Certificate Managers tab
Hands-On Exercise: Backing Up Your CA

Backing up your CA is by far one of your most important tasks, since despite the ability to recover certain data using recovery agents, the design of PKI specifically prohibits the ability to generate exactly the same key to prevent identities from being compromised. Lucky for us, backing up and restoring our private key, CA certificate, and certificate database is as easy as running a wizard.

1. Create a folder on the local drive to which the CA will be backed up (that is, C:\CABackup).
2. Open the Certification Authority management console if it is not already open.
3. Right-click the CA server instance, select All Tasks, and then select Back Up CA. This will launch the Certification Authority Backup Wizard.
4. Click Next on the Welcome screen.
5. Check the Private Key and CA Certificate, and, Certificate Database and Certificate Database Log checkbox (Figure 12-17). Click Browse select the folder you created in step 1, and then click Next.
6. Enter and confirm a password that will be used to secure the private key and CA certificate file, and then click Next (Figure 12-18).
7. Click Finish to complete the backup process.
To restore your CA from this backup, simply run the Certification Authority Restore Wizard and reverse this process.

**Hands-On Exercise: Renewing Your CA Certificate**

During the installation of our CA, you created your CA certificate by supplying a password for your private key and you also specified how long this certificate will be valid. Eventually, you will need to renew this CA certificate or it will expire and will no longer be valid. You might also want to renew your CA certificate if your signing key has been compromised or you need a new CA certificate to create a new CRL.

1. Open the Certification Authority management console if it’s not already open.
2. Right-click your CA server instance, select All Tasks, and then select Renew CA Certificate.
3. Since you cannot renew your CA certificate while AD Certificate Services is running, you will be prompted to close AD Certificate Services. Click Yes.

4. Select No when asked to create a new signing key (Figure 12-19), and then click OK. You would select Yes if you wanted to generate a new signing key in addition to a new certificate.

5. A new certificate will be generated and AD Certificate Services will be started.

6. To verify that a new certificate has been created, right-click the CA server and choose Properties. On the General tab, a new CA certificate will be visible in addition to the previous CA certificate. Its expiration date will be equal in length to the previous certificate, so if the old certificate was valid for five years, for example, the new certificate will also be valid five years from when it was issued.
ISSUING CERTIFICATES

An Enterprise CA can issue certificates using a number of different methods. Users can request certificates directly using the Certificates MMC snap-in from a computer that is joined to the domain. Certificate requests can also be submitted through the Web using the Web Enrollment Agent. Computers that are part of the domain can automatically obtain computer certificates if automatic enrollment has been enabled through Group Policy. A good use for automatic enrollment is to make issuing certificates for computers that participate in IPSec much easier. Instead of having to create and install certificates manually for each device, you could put them in an organizational unit (OU) and create a Group Policy object (GPO) to enable automatic enrollment and allow those computers to retrieve the correct certificate based on the template you configure for that policy.

Hands-On Exercise: Obtaining a Certificate Using Web Enrollment

One of the easiest ways to instruct your users to request and obtain a certificate is through the Web Enrollment Agent. All they will need is Internet Explorer and connectivity to your CA server.

1. Open Internet Explorer. Choose Tools | Internet Options, and open the Security tab. Click Trusted Sites, and then click the Sites button. Add http://win2k8ca (replace win2k8ca with the appropriate server name that hosts your CA) to your list of trusted sites. Make sure you uncheck the Require Server Verification (https) for All Sites in This Zone checkbox. Click Close.

2. In the IE Trusted Sites Zone security settings, click the Custom Level button. Scroll down to the ActiveX Controls and Plug-Ins section, and set both Download Unsigned ActiveX Controls and Initialize and Script ActiveX Controls Not Marked As Safe for Scripting to Enable (Figure 12-20). Without this option enabled in Internet Explorer 7, your browser will not allow certificates to complete the certificate enrollment process. Click OK to save the changes. Click OK again to close the Internet Options window.

3. Go to http://win2k8ca/certsrv and enter your domain credentials when prompted. Click Request a Certificate on the Welcome screen (Figure 12-21).

4. Select User Certificate on the Request a Certificate screen (Figure 12-22).

Note that you can also use the Web Enrollment Agent to request other certificate types by clicking Advanced Certificate Request.

5. If you are logged in with domain credentials, no additional information is needed since the server automatically obtains this data from AD. Click Submit to complete the certificate enrollment process (Figure 12-23).
6. Finally, click Install This Certificate to install the newly created certificate to your personal certificate store (Figure 12-24).

7. You will receive a successful installation message upon completion.

**CERTIFICATE REVOCATION**

Creating and issuing certificates is one of the most important functions of a CA. Since certificates are heavily integrated with authentication and encryption, it makes sense to have an efficient and effective method for revoking a certificate when it should no longer be used.
Chapter 12: Enterprise Public Key Infrastructure

For example, if you issue a user certificate to an individual who then leaves the company, you will probably want to prevent that certificate from being used to access any of your systems.

Certificate revocation is a straightforward process. You simply right-click the certificate you want to revoke in the Issued Certificates folder in the CA management console and select Revoke Certificate from the All Tasks pop-up menu.

All revoked certificates are automatically added to the CA's CRL—the list of serial numbers of revoked certificates signed by the CA to ensure its integrity. Although this list is continually updated internally with the CA, it is not published immediately to Active Directory. Instead, the CRL gets published according to its own schedule. CRLs can get fairly large. To manage replication, you can configure the CRL to publish delta CRLs, which contain only changes since the last replication. By default, CRLs are published once...

Figure 12-21. AD Certificate Services Web request Welcome screen

For example, if you issue a user certificate to an individual who then leaves the company, you will probably want to prevent that certificate from being used to access any of your systems.

Certificate revocation is a straightforward process. You simply right-click the certificate you want to revoke in the Issued Certificates folder in the CA management console and select Revoke Certificate from the All Tasks pop-up menu.

All revoked certificates are automatically added to the CA's CRL—the list of serial numbers of revoked certificates signed by the CA to ensure its integrity. Although this list is continually updated internally with the CA, it is not published immediately to Active Directory. Instead, the CRL gets published according to its own schedule. CRLs can get fairly large. To manage replication, you can configure the CRL to publish delta CRLs, which contain only changes since the last replication. By default, CRLs are published once...

Figure 12-21. AD Certificate Services Web request Welcome screen

For example, if you issue a user certificate to an individual who then leaves the company, you will probably want to prevent that certificate from being used to access any of your systems.

Certificate revocation is a straightforward process. You simply right-click the certificate you want to revoke in the Issued Certificates folder in the CA management console and select Revoke Certificate from the All Tasks pop-up menu.

All revoked certificates are automatically added to the CA's CRL—the list of serial numbers of revoked certificates signed by the CA to ensure its integrity. Although this list is continually updated internally with the CA, it is not published immediately to Active Directory. Instead, the CRL gets published according to its own schedule. CRLs can get fairly large. To manage replication, you can configure the CRL to publish delta CRLs, which contain only changes since the last replication. By default, CRLs are published once...
Figure 12-22. Requesting a certificate

Figure 12-23. Completing certificate enrollment
a week, while delta CRLs are published once a day. You can view or change how often the CRLs are published by right-clicking the Revoked Certificates folder in the CA management console for your server and selecting Properties. You can change the CRL publication interval and enable as well as set the publication interval for the delta CRLs from the CRL Publishing Parameters tab, as shown in Figure 12-25. The next update time is also displayed so you know when the next publication is scheduled to take place. You can also view the CRL or the delta CRL from the View CRLs tab shown in Figure 12-26.
Figure 12-25. Revoked Certificates Properties screen

Figure 12-26. View CRLs tab
CHAPTER SUMMARY

This chapter covered the installation, configuration, and management of Active Directory Certification Services, which is the physical implementation of PKI for Windows Server 2008. Certificates can be used for both encryption and identity management. When implemented correctly, certificates can greatly enhance security by adding another layer of protection for securing your data. You can use certificates with IPSec to encrypt communication between two systems. It can be used to encrypt data on an EFS formatted hard drive and even to validate the authenticity of a server’s identity that is required for SSL traffic. For identity management, certificates can be used to sign e-mail messages digitally or to authenticate to the domain using certificates loaded onto smart cards.

CAs can be installed either as Enterprise or Stand-alone CAs. Stand-alone CAs are most appropriate for generating certificates for entities outside of your organization. Generally, you should install an Enterprise CA since it provides the most functionality and greatest overall flexibility. It requires AD because it uses AD as a central store for certificates and rides on its replication capabilities to publish its data to all its clients. You can use certificates generated by an Enterprise CA for authenticating to your domain using smart cards. An Enterprise PKI solution should be part of a larger overall security initiative. While it does help create a more secure environment, its deployment and subsequent use should be built around very good processes that tie in the various aspects of PKI. You should pay careful attention to processes surrounding encryption, identity management, provisioning, and revocation of certificates.
This page intentionally left blank
CHAPTER 13

Windows PowerShell
Windows Server 2008 is the first operating system released by Microsoft that ships with Windows PowerShell. Windows PowerShell is a command shell similar to the traditional command prompt (cmd.exe), except it’s much more powerful. Not only does it include many more built-in commands, called cmdlets (pronounced command-lets), but these cmdlets provide a more structured approach to running command-line tasks and increase flexibility by allowing you to interact with virtually anything in the operating system that can be interfaced with a cmdlet or the .NET Framework. You can run all the commands available in cmd.exe directly from the PowerShell prompt, which helps ease the transition to this new command shell. In short, PowerShell gives Windows administrators more tools for automating routine tasks.

POWERSHELL AT A GLANCE

At first glance, PowerShell looks like nothing more than a version of the command prompt you’ve been using since the good-old MS-DOS days. However, PowerShell completely revolutionizes scripting and automating a Windows environment. PowerShell was designed from the ground up to be a powerful tool that gives administrators more control while interacting with the operating system.

Windows PowerShell requires .NET Framework 2.0 to run since it is built around .NET interfaces. In fact, each cmdlet is actually .NET code that interacts with the .NET Framework. This allows Windows PowerShell to be extended seamlessly. For example, while PowerShell ships with more than 100 cmdlets, more cmdlets can be installed or developed to suit all your needs. You may be wondering how this differs from having additional command-line tools like those found in the Windows resource kits. While cmdlets do extend the functionality of the shell, just like command-line applications do, the fact that cmdlets are written using a standard interface means that each command can interact with other commands without requiring extensive string parsing routines. If you haven’t done much scripting in the past, much of this might be a bit confusing—but don’t put down the book just yet, because we’ll go into this in greater detail in this chapter.

Hands-On Exercise: Installing Windows PowerShell

Although Windows Server 2008 is the first Microsoft operating system to ship with PowerShell, it isn’t installed automatically by default. As with many other features, this is by design, since Windows Server 2008 installs only the minimal number of components by default to minimize security risks.
Follow these steps to install Windows PowerShell on your server.

1. Open Server Manager.
2. Click the Add Features link in the Features Summary section to start the Add Features Wizard.
3. Select Windows PowerShell from the list of Features, and then click Next (Figure 13-1).
4. Confirm the installation selection and click Install.

**Figure 13-1. Selecting the Windows PowerShell feature**

![Add Features Wizard](image)
5. Click Close when the installation completes.

6. Choose Start | All Programs | Windows PowerShell 1.0 | Windows PowerShell to open the Windows PowerShell command window (Figure 13-2).

**GETTING YOUR FEET WET**

Hopefully, you’ve become curious enough to keep on reading. If not, I suggest you keep reading, as the magic of PowerShell becomes crystal clear once you start using it. As shown in Figure 13-2, the interface looks similar to the old familiar command prompt interface, except Windows PowerShell appears in the title bar, the path for the prompt is prefixed with the letters PS, and the background color is blue instead of black. You can type in familiar DOS commands such as CD, DIR, COPY, MOVE, DEL, and so on. But in this interface, you’re not running these old commands. Instead, these commands are *aliases*
of real PowerShell cmdlets. To see a list of all the available commands, type `help` at the prompt and press ENTER. As you can see in Figure 13-3, a list of all available commands is displayed including aliases. Press ENTER again to continue, and you can scroll through all the aliases and cmdlets available. For example, the `DIR` command is an alias to the `Get-ChildItem` cmdlet and `HELP` is actually an alias to the `Get-Help` cmdlet.

If you want to learn more about a particular command, you can use `Get-Help` to display that information. For example, if you want to know more about the `Get-ChildItem` cmdlet, you can run this command:

```
Get-Help Get-ChildItem
```

Typing `Get-Help` will show you how to use this cmdlet. The most important switch is `-full`, which displays the full help file for a particular command. Using `Get-ChildItem` as an example, here’s the command to get more detailed help about `Get-ChildItem` as well as a few examples:

```
Get-Help Get-ChildItem -full
```
With those basics out of the way, let’s delve into some really cool stuff before taking a step back to talk about more technical details. Whenever you are working on a server and troubleshooting an issue, you often go to the Task Manager to see what processes are running, who’s using them, how much memory they are using, and other information. With previous versions of Windows, if you wanted to get more information from the command prompt, you would have to rely on some resource kit or third-party tools. PowerShell, on the other hand, comes with a cmdlet that you can use to show all your running processes:

Get-Process

This command sorts the output by the Process Name by default; however, for example, say you wanted it sorted by process ID. No problem; just run this:

Get-Process | Sort-Object Id

Another common administrative task is managing Windows Services. This is a snap with PowerShell, since it has built-in cmdlets for managing services. To see a list of all the services on the system and its status, you can run this command:

Get-Service

You can also indicate a specific service you want to query by providing the service or display name, and you can use wildcards if you don’t know the exact name. For example, the following command will list the status of all services that contain the string win in the display name:

Get-Service -displayname *win*

Windows PowerShell also includes cmdlets that allow you to interact with Windows Management Interface (WMI) with relative ease. The Get-WmiObject cmdlet provides a direct interface to query any WMI object accessible to you. For example, if you want to know information about your BIOS and system information, you can query the Win32_BIOS and Win32_ComputerSystem WMI classes using the following commands:

Get-WmiObject -class Win32_BIOS
Get-WmiObject -class Win32_ComputerSystem

As you can see in Figure 13-4, the Get-WmiObject command can be useful for retrieving asset-based information using WMI.

You can see that even with just a handful of commands, Windows PowerShell can be a powerful ally. Now read on for more about what cmdlets can do.
Cmdlets follow a standard naming convention of *verb-noun*. The reason for this is quite simple: The command name is descriptive in and of itself. Every cmdlet name intuitively explains what it does and to which objects. For example, `Get-Service` immediately conveys that this particular command can get information about a service. Compared to traditional command-line applications, cmdlets are also designed to separate the tasks for retrieving and setting information as well as separating the data from its presentation. So you’ll typically find a cmdlet to *retrieve* information about an object and a separate cmdlet to *set* information about the same object. For example, the `Get-Date` cmdlet retrieves the current date and time, while the `Set-Date` cmdlet sets the date and time.
When we talk about Windows PowerShell separating data from presentation, we’re saying that data returned by a cmdlet isn’t actually how it ends up being displayed on the screen. So do I have you scratching your head now? This concept is not really that difficult. In traditional command-line tools, data would typically be retrieved by the command, and it would be formatted to look pretty much as it does on the screen. Sometimes a command would have switches that would allow the display to be different. For example, the typical command-line `DIR` command displays the directory listing as a fairly detailed list by default, but you can use the `/W` switch to display it in wide format, where only file names are displayed in columns. While that works fine when you’re dealing with one command, it becomes troublesome when you want to use that output in a different command. Typically, you would have to run some command-line tool or build a script that could parse the output of the previous command and translate it into useful data for the next command.

Working with output as strings brings about many limitations, such as dealing with special characters or trying to parse command output of something that doesn’t generate output where a pattern for parsing out useful information can be clearly defined. Windows PowerShell overcomes this limitation by having cmdlets return objects rather than simple plain text. If you run a command by itself, PowerShell automatically invokes the default formatting to render the output as text. But say, for example, that you want to pipe the output into another command. Rather than having that command output text that you would then have to parse yourself, the command simply returns an object containing that data, so that the next command can work on the data set itself and manipulate directly without having to try to deal with parsing strings.

Consider our good friend `Get-ChildItem` (the old `DIR`). By default, Windows PowerShell renders its output just as the `DIR` command did. Suppose you want to convert this output to HTML. In the past, you would have had to write a Windows Shell Script or even write the whole thing in VBScript to get this type of functionality. In PowerShell, since `Get-ChildItem` returns an object representing the list of files in that folder, you can simply pass it to another cmdlet that can take a list of objects and convert it directly to HTML tables. In practice, all you need to do is run the following command and it will generate an HTML file, as shown in Figure 13-5:

```
Get-ChildItem | ConvertTo-HTML > MyFile.html
```

If that alone didn’t wow you, you probably have never had to write your own HTML conversion routine. Writing an HTML routine is not that difficult, but it’s certainly not a one-liner.
Chapter 13: Windows PowerShell

Windows PowerShell was created out of the need to have a scripting language that could easily interface with .NET managed code. With so much of Microsoft’s own products being developed in .NET managed code, it seemed natural to offer a solution to make it simple for administrators (non-developers) to reap the benefits of having so many of these .NET interfaces available. In fact, if something is .NET enabled, it can almost certainly be managed with Windows PowerShell. This is great news since most, if not all, of Microsoft’s flagship products will be shifting toward leveraging the .NET Framework. The good news for Windows administrators all over the world is that more and more of

Figure 13-5. HTML file created by running Get-ChildItem through ConvertTo-HTML

WINDOWS POWERSHELL AND .NET

Windows PowerShell was created out of the need to have a scripting language that could easily interface with .NET managed code. With so much of Microsoft’s own products being developed in .NET managed code, it seemed natural to offer a solution to make it simple for administrators (non-developers) to reap the benefits of having so many of these .NET interfaces available. In fact, if something is .NET enabled, it can almost certainly be managed with Windows PowerShell. This is great news since most, if not all, of Microsoft’s flagship products will be shifting toward leveraging the .NET Framework. The good news for Windows administrators all over the world is that more and more of
these systems will be completely available to us for automation using Windows Power-Shell. These products will not only ship with .NET interfaces that can easily be used by developers, but they will also supply their own set of cmdlets that allow administrators to use Windows PowerShell for automating management tasks.

**WINDOWS POWERSHELL, SCRIPTING, AND SECURITY**

Naturally, a powerful command-line–based shell supports the ability to be scripted. You can make scripts for Windows PowerShell just as you can make scripts in the traditional command shell, except that Windows PowerShell has a more complete scripting language where looping and various logic branches can be readily implemented. In fact, since it is built on top of the .NET Framework, PowerShell uses much of the same syntax and naming conventions as the .NET programming languages such as C#.

Unfortunately, the ability to automate tasks has been exploited numerous times by virus, worm, and spyware writers all over the world. To address some of these concerns, a few default settings are built into PowerShell:

- **▼** No file is associated with the PowerShell executable. That means, for example, that even if you create a PowerShell script called myps.ps1 (.ps1 is the extension used for PowerShell scripts), you can’t simply run it by double-clicking it. It will, however, open in Notepad instead so you can view the source.

- **■** You can run only scripts that are signed and trusted by your system’s certificate store.

- **▲** When allowed, you can run a script from the PowerShell interface, but you must always explicitly enter the path. So if a malicious hacker places a script name similar to another command in your search path somewhere, you won’t execute that malicious script instead of the intended command.

Obviously, these measures aren’t foolproof, but they certainly help limit the security exposure. You can override these settings if you want. For example, you can allow execution of non-signed scripts by changing the current execution policy. To view your current execution policy, run this command:

```
Get-ExecutionPolicy
```

Four execution policies are available:

- **▼** **Restricted** (Default) No scripts are allowed.
- **■** **AllSigned** Only signed scripts are allowed.
- **■** **RemoteSigned** Locally executed scripts are allowed. Anything else must be signed.
- **▲** **Unrestricted** All scripts are allowed.
To change the execution to RemoteSigned (minimally recommended if you must run unsigned scripts locally), you can change it by running the following command:

```powershell
Set-ExecutionPolicy RemoteSigned
```

You can also change the execution policy using Group Policy if you want to make this change across your organization from a central location.

### Hands-On Exercise: Your First PowerShell Script

Before we go on to the details of the various components used in creating scripts, let’s take a moment to put together a short script to help show how you can create a script and run it within PowerShell—assuming you have changed your execution policy to at least RemoteSigned so that local scripts are allowed to be executed without having to sign them.

1. Open Notepad and enter the following code:

   ```powershell
   $s = "Hello World!"
   write-host $s
   ```

2. Save the file as C:\helloworld.ps1.
3. At the Windows PowerShell prompt, enter

   ```powershell
   powershell C:\helloworld.ps1
   ```

   The string “Hello World!” should be displayed on the screen. Congratulations! You have now created and executed your first, albeit mundane, Windows PowerShell script.

### VARIABLES

The concept of variables exists in every scripting and programming language; they essentially allow you to name placeholders for values that you will use within the script. In Windows PowerShell, you can use any name as a variable, but it must start with the dollar sign ($). You can use any combination of letters, numbers, and symbols. You can even use a space in the variable name, provided that you enclose the entire variable name in curly braces `{}`. The following are valid declarations of variables:

```powershell
$MyName = "Steve"
$x = 5
${Variable with space} = "See the curly braces!"
```
As you can see, defining a variable and assigning it a value is a fairly straightforward endeavor. In fact, if you’ve written batch files, you can do all of the same things above except for the last one, which is using spaces in the variable name, using the `SET` command.

Windows PowerShell also supports typecasted variables. This means that you can tell PowerShell what kind of value the variable is going to store. This is generally considered best practice since it prevents strange bugs from occurring if you write more complicated scripts. For example, look at this piece of code:

```powershell
$a = 2
write-host ($a + 2)
```

We assigned the value 2 to the variable `$a` and then output to the screen the value of `$a + 2`. As expected, this code sequence will result in the value 4 being displayed on the screen.

Now look at this code:

```powershell
$a = 2
$s = "Some string"
...some more code...
$a = "Steve"
...some more code...
write-host ($a + 2)
```

In this example, you create two variables: `$a` contains the value 2 while `$s` contains the value "Some string". Now assume that you accidentally assigned the value "Steve" to the variable `$a` when you meant to assign it to `$s`. (If you have a QWERTY keyboard, the `A` key is right next `S`, so this kind of mistake can easily happen.) This time, the code outputs "Steve2" instead of what we really intended, which was the value 4. If this was in a large script, this error might be hard to find.

To avoid this kind of problem, you can typecast each variable, like so:

```powershell
[int]$a = 2
[string]$s = "Some string"
...some more code...
$a = "Steve"
...some more code...
write-host ($a + 2)
```

You prefixed each variable as you first used it, with `[int]` and `[string]`. I told Windows PowerShell that `$a` would hold an integer while `$s` would hold a string. If you run this code, Windows PowerShell would spit out an error telling me that “Steve” cannot be converted to type System.Int32 (the long name for an integer). The system has enforced the fact that you are trying to assign a non-integer to a variable that is supposed to hold only integers. This way, you can go into the code and immediately and see that the error is caused by the fact that you tried to assign "Steve" to `$a` instead of `$s`. You simply need to correct that mistake and the script will operate as expected.
Interestingly enough, no special naming convention is needed to define an array. An array is generally a simple data structure in which a group of values or objects can be accessed using the same name but using indexes to access each individual element. If you've looked at VBScript code, you have undoubtedly seen something similar to this:

```
Dim myArr(2)
myArr(0) = "first"
myArr(1) = "second"
myArr(2) = "third"
WScript.Echo myArr(1)
```

This isn't a VBScript tutorial, so we won't go into this example in great detail; basically, this code defines an array containing three elements (even though there's a 2 in the parentheses since the 2 signifies the index of the last element starting from 0). You then assign values to each element and then output the value of myArr at index 1, which in this case would be the string second.

The following code snippet shows how arrays are dealt with in Windows PowerShell:

```
$myArr = "first","second","third"
$myArr[1] = "2nd"
write-host $myArr
```
The result of this little code snippet above would be *first 2nd third* being displayed on the screen on one line. Just like many programming languages, the arrays are 0-index-based, so $myArr[0]$ refers to the first element, $myArr[1]$ refers to the second element, and so on. Notice how you implicitly defined $myArr$ as having three data elements; but what if you wanted to add two more? In VBScript, you would have had to use the `ReDim` statement to resize the array. But in PowerShell, this is extremely easy:

```powershell
$myArr = "first","second","third"
$myArr = $myArr + "fourth","fifth"
write-host $myArr[4]
```

This code snippet results in the string *fifth* being displayed on the screen. Notice that all you had to do to extend my existing array was to add the new data elements you wanted using the plus (+) operator. Windows PowerShell automatically handles the memory allocation for me.

### CONDITIONAL STATEMENTS

One of the most important features needed in any scripting environment is the ability to define conditional statements such as *if x equal 2 then do this otherwise do something else*. After all, without conditional statements such as `If/ElseIf/Else` combinations, you can’t really implement any kind of logic in your script. The key to being able to create branches in your code is to combine conditional statements with comparison operators to make decisions based on values of variables within your script. Here’s an example:

```powershell
$a = 5
if ($a -eq 1) {
    write-host "One"
}
elseif ($a -eq 2) {
    write-host "Two"
}
else {
    write-host "Anything but One or Two!"
}
```

Hopefully you can follow this slightly longer code snippet. First, you assign the value of 5 to $a. Then check if the value of $a is equal to 1 and, if it is, you output *One* to the screen. If $a is not equal to 1, check whether it is equal to 2 and output *Two* if it is. If neither condition is met, the string *Anything but One or Two!* is displayed. Based on the value of $a being 5, this script will output *Anything but One or Two!* Try changing the value of $a to a different number to see the output.
In the preceding example, we have used the `-eq` comparison operator to check whether the variable equaled a certain value. You can use seven different comparison operators in Windows PowerShell, and each starts with a hyphen (`-`) followed by a two-letter abbreviation of the comparison it performs:

- `-eq` Equal to
- `-ne` Not equal to
- `-notmatch` Does not match
- `-gt` Greater than
- `-ge` Greater than or equal to
- `-lt` Less than
- `-le` Less than or equal to

Another method for performing a conditional branching within your code is through the use of a `Switch` statement. A `Switch` statement is a more efficient way of handling situations in which you want to test more than two conditions with an `If/Elseif` statement. For example, let’s say you have a variable that can contain the name of one of seven different colors—Red, Blue, Yellow, White, Green, Orange, Black—and you want to perform different actions based on each individual color. If you could only use `If/Elseif` statements, it would take many such statements and would not be easy to read later. Using a `Switch` statement makes the code much neater and intuitive, as in this example:

```powershell
$color = "blue"
switch ($color) {
    red {write-host "Color Red"; break}
    blue {write-host "Color Blue"; break}
    yellow {write-host "Color Yellow"; break}
    white {write-host "Color White"; break}
    green {write-host "Color Green"; break}
    orange {write-host "Color Orange"; break}
    black {write-host "Color Black"; break}
}
```

Notice how easy it is to see which code gets executed based on the value of $color. What you haven’t seen before is the `break` statement. We’ll discuss this in the next section, but essentially it tells Windows PowerShell to stop processing the rest of the potential switch conditions, which makes sense since we’ve already found a match.
GOING LOOPY

One of the main reasons administrators write scripts is to automate repetitive tasks—after all, you have more important things to do than renaming a bunch of files or setting permissions to a folder structure. Computers are excellent for these kinds of tasks, because they don’t get tired, they don’t complain, and in general they can do this around the clock, even while you are sound asleep. Another key construct in any scripting language is the ability to create loops in your script. A typical example would be a script to go through all the files in a folder and rename each file so that it is prefixed by the string backup-.

Loops are quite simple, but they are one of the biggest reasons why scripts “go wild.” In general, the loop has a condition that defines when it should stop doing whatever it is that it’s doing. Sometimes coding or logic errors result in a state in which that condition is never met and your script gets caught up in an endless loop that keeps on going, since the condition to make it stop will never happen. You can implement loops in Windows PowerShell in four ways: For, Foreach, While, and Do…While statements.

The For statement, otherwise known as a For loop, runs a block of code until a condition is found to be true. Normally, you would use a For loop when you want to initialize a variable, run it while the condition is true, and then run some code that is repeated for each execution. FOR loops have the following syntax:

```
For(<init>;<condition>;<repeat>) {
  <code to run in the loop>
}
```

The following code snippet is a For loop that counts from 1 to 100:

```
For($i=1;$i -lt 101;$i++) {
  write-host $i
}
```

The <Init> section is executed only once for the For loop and is used for initialization. In this example, I used this section to initialize $i to the value of 1. The <Condition> section defines what condition must be true before the code in the code block gets executed. In this example, I state that if $i is less than 101, it can execute the code. The <Repeat> section is code that is executed each time the loop executes. In this case, I increment $i by 1 by using the shorthand notation of $i++, which is functionally equivalent to $i = $i + 1. Finally, for each iteration of the loop, I output the value of $i. This effectively makes the script count from 1 to 100 since once $i is incremented to the value of 101, the condition that $i is less than 101 is no longer true and the loop stops executing.

**NOTE** If you have a programming background or have used C, C++, or Java, the ++ operator should be nothing new to you. In fact, many constructs in Windows PowerShell should be familiar to anyone who has worked with the C programming language.

The Foreach statement is used to loop through a collection of items. Unlike the For loop where you define a variable, a stop condition, and repeating code, the Foreach
statement is designed to take a collection as its parameter and run a block of code for each item in that collection (hence the name). Foreach statements have the following syntax:

```
Foreach ($<item> in $<collection>) {
    <command_block>
}
```

This is an extremely useful looping statement. The following code snippet shows how you can use Foreach to display the name of an item in the Windows directory:

```
Foreach ($file in Get-ChildItem C:\Windows) {
    write-host $file
}
```

Hopefully a light bulb just lit up above your head. You can run any PowerShell code in the command block so you can easily convert this Foreach example to do something useful. For example, you might use this code to rename every item in a specific folder. You can use Foreach to iterate through any collection, including arrays.

The While statement, otherwise known as a While loop, is similar to a For loop in that it runs a command block any number of times while a condition is true, except its only parameter is a condition statement. This means that initializing or incrementing any variables to make sure the condition will eventually evaluate to false so that the loop will end has to be done separately. The syntax for a While loop is this:

```
While(<condition>) {
    <code_block>
}
```

Notice how much simpler it is than a For loop. To compare the two, the following code snippet shows how we can use a While loop to have our script count from 1 to 100:

```
$i = 1
while($i -lt 101) {
    write-host $i
    $i++
}
```

The Do...While statement is an interesting variation of the While loop in that just like the While loop, it loops through a code block while a condition is true. The main differentiator is that since the condition is checked at the end of the code block, every Do loop is guaranteed to execute at least once. Consider the following example:

```
$a = 11

do {
    write-host $a
```
Notice how you initialized the $a variable to the value of 11. This is already greater than the condition for the loop, which is set to run only while $a is less than 10. If you run this code snippet, you will see the output of 11 followed by the string Done! As you can see, since the while condition is at the end of the block, it isn’t evaluated until after the block has executed at least once. In this case, the value of $a was already displayed before the while condition was checked and the loop terminated immediately due to the value of $a being too great.

A typical example for a scenario where a Do…While loop would be appropriate is when prompting the user for some information. If the code within the loop is designed to display the prompt, process the input, and then compare it against a certain value (for example, if you are prompting for a password), then a Do…While loop guarantees that the prompt will be displayed at least once. It also makes sense to do this since you obviously have nothing to compare against until the user has entered some information, so all the other looping constructs would be inefficient since they want to evaluate a condition before even getting any information from the user.

Finally, two other statements are very important to loops: break and continue statements. The break statement is a way to completely bypass any other condition for the loop and instruct Windows PowerShell to get out of the loop right away (kind of like the “Go to Jail, Do Not Pass Go” card in your favorite board game). The continue statement is slightly different in that it instructs Windows PowerShell to stop processing the rest of the code in the code block and immediately jump to the next iteration of the loop.

Let’s see these in practice. The following example shows how break and continue statements can be used to perform flow control within a loop:

```powershell
$a = 0
write-host "Starting to count to 10..."
while ($a -lt 11) {
    $a++
    if ($a -eq 3) {
        continue
    }
    if ($a -eq 8) {
        break
    }
    write-host $a
}
write-host "Done!"
```

In this code, we are trying to count from 1 to 10 with a twist. First off, notice how I initialize $a to the value of 0. This is because the first operation we do within the code block is to increment its value by 1. The condition for the loop is to run when $a is less
than 11. The twist is this: if $a$ equals 3, you issue the `continue` statement that forces it to jump through the next iteration of the loop and skip the rest of the code, including the `write-host` cmdlet. If the value of $a$ is equal to 8, you instruct Windows PowerShell to cease processing the code block immediately and jump out of the loop. This results in this output:

```
Starting to count to 10...
1
2
4
5
6
7
```

Notice how it skipped outputting the number 3 since the `continue` statement forced the code to the next cycle and the counting stopped at 7 instead of 10 since the `break` statement took effect when $a$ was equal to 8 but before `write-host` got a chance to output its value.

### POWERSHELL IN ACTION

If you’re like most Windows systems administrators, you want proof that PowerShell can make your life easier before you commit to using it. The good news is that using just the basics covered so far, you can immediately perform a large number of interesting real-world tasks.

Let’s go back to the `Get-Service` cmdlet. `Get-Service` returns a collection of service objects including their various properties. Suppose you want to show a list of services that are currently running based on the data retrieved by `Get-Service`. All you need is this one-liner:

```
Get-Service | ForEach {if ($_.Status -eq "Running") {write-host $_.DisplayName}}
```

A few new constructs here need some explanation. `Get-Service` is straightforward: It runs and its output is piped as the input to the next section, which is the `ForEach` loop. Since `Get-Service` returns a collection of service objects, the `ForEach` statement is perfect for looping through each item `Get-Service` returns. Inside the outer set of curly braces is our block of code. In this case, we are using an `If` statement to check whether the status of the service is equal to "Running", and if it is, it outputs the display name.

Two things must pop out to you. First is the `$` notation and second is the dot (.) notation. The `$` variable is one of the system-defined automatic variables. In a pipe, it holds the current pipeline object. In this example, for each iteration of the loop, `$` would reference each service item in the collection returned by `Get-Service`. The dot (.) notation is a member operator for object. Every object has a set of properties. For example,
a service has a status, name, and displayname among other things. To access each of these individual properties of an object, you use the dot notation to get to that property using the syntax `objectname.propertyname`.

Here’s a practical solution to a common problem. You run a nightly job that dumps a bunch of text files on a certain folder and you want Windows PowerShell to go through all the files in that folder and change them from a .TXT file extension to a .BAK. One line of PowerShell does the trick:

```powershell
Get-ChildItem E:\Logs\* -include *.txt | foreach {move-item $_ ($_ -replace(".txt",".bak"))}
```

You leverage the `Get-ChildItem` cmdlet (alias DIR) against the E:\Logs directory to look for any file ending in .TXT. This data set is then piped to a `Foreach` loop, where the `Move-Item` cmdlet changes the file extension to .BAK. Note how ($_ - replace (".txt",".bak")) is used to generate the new filename, and then the results of the string replacement are used as the destination name for the `Move-Item` cmdlet. If you’ve ever had the pleasure of writing a Windows Shell Script or even VBScript to perform a similar function, you can appreciate how elegant this solution is; it’s where PowerShell really begins to shine.

### Working with the Registry

You cannot be a Windows administrator and not have to deal with the Windows registry at least once a day. After all, it is the central point for almost all configuration data regarding your system and applications. Typically, you would use `Regedit.exe` to edit the

---

**Automatic Variables**

While `_` is a commonly used automatic variable, numerous automatic variables are defined by the system. Here are a few of the most useful ones:

- `_` Contains the current pipeline object
- `?$` Contains True if the last operation succeeded; otherwise False
- `$_` Contains an array of the parameters passed to a function
- `$foreach` Refers to the enumerator in a `Foreach` loop
- `$Home` User’s home directory; equivalent to `%homedrive%\%homepath%`
- `$LASTEXITCODE` Contains the exit code of the last Win32 executable execution
- `$PsHome` Directory where Windows PowerShell is installed
- `$Host` Contains information about the current console host such as version number
registry, or if you wanted to script it, you could use the command-line tool `Reg.exe`. Working with the registry with Windows PowerShell is easy, because PowerShell treats the registry like a file system. If you think about it for a second, it makes sense—after all, the registry is organized just like a directory tree. You can think of registry keys as folders and registry values as files. This natural similarity made it quite easy to have the registry accessible directly through the PowerShell command prompt.

One of the most common registry keys we access is the Run key. This key stores a list of programs to run at startup, or in the case of the Run key in HKEY_CURRENT_USER, when the user logs in. To access the HKEY_LOCAL_MACHINE Run key from the PowerShell console, you would run the following:

```
CD HKLM:
CD software\microsoft\windows\currentversion\run
```

Notice how this is just like going through your folder structure, except a special drive called HKLM: takes you to the registry instead. If you want to get a list of values on the Run key, you might be tempted to use the `DIR` command. If you try running this, you might be surprised to find out that it returns nothing. The `DIR` command lists only registry keys and not registry values. To get the actual registry values in the Run key, you have to use the `Get-ItemProperty` cmdlet since the registry values are treated as properties of registry keys. `Get-ItemProperty` requires that you specify the path of the object for which you want to get a property.

The following example shows how you can query the registry values of the current key or of another key:

```
Get-ItemProperty .
Get-ItemProperty HKLM:\Software\Microsoft\Windows\CurrentVersion
```

You can see the results of running `Get-ItemProperty .`, while in HKLM:\Software\Microsoft\Windows\CurrentVersion\run in Figure 13-6. Since `Get-ItemProperty` is not specifically designed just for registry keys, it shows additional metadata regarding the object you are running it against. In this case, it shows the path to the registry key, the path to its parent, the key name, the “drive” (hive) of the key, the provider, and finally a list of values in the Run key. Since I have only one entry in my Run key, namely MyApp, pointing to C:\Apps\MyApp.exe, it gets displayed after the general object information.

Only HKEY_LOCAL_MACHINE (HKLM) and HKEY_CURRENT_USER (HKCU) are accessible quickly using the `CD HKLM:` or `CD HKCU:` shortcut. To get to other hives such as HKEY_CLASSES_ROOT or HKEY_USERS, you need to connect directly to the PowerShell Registry Provider. It’s not that difficult to do. The following command sequence finds out which application opens up .TXT files:

```
CD REGISTRY::
CD HKEY_CLASSES_ROOT\\.txt\shellnew
Get-ItemProperty .
```
You could also put it all on one line:

Get-ItemProperty REGISTRY::HKEY_CLASSES_ROOT\.txt\shellnew

**TIP** Typing Get-ItemProperty over and over gets old pretty quickly, so Microsoft has predefined an alias for this cmdlet. Instead of typing Get-ItemProperty, all you need to do is type GP. Don’t you love aliases?

**Working with Dates and Times**

Knowing dates and times and performing calculations using dates and times are necessities for every administrator. Time has many uses and is significant in almost all aspects of computing. Fortunately, PowerShell comes with a rich set of date and time–related features to help you tackle these tasks with relative ease. It’s not surprising that the cmdlet that handles data and time is called Get-Date. (Perhaps Get-DateTime might be more appropriate, but just be happy you have less to type.) Running Get-Date by itself
returns the current day, date, and time. If you want to return just the date or time, you can run either of the following commands, respectively:

Get-Date -displayhint date
Get-Date -displayhint time

The date and time are displayed in the current time zone configured on your server. If you are a global company, sometimes it works best when everyone expresses dates and times in terms of Universal Time Code (UTC). This isn’t a problem since the Get-Date cmdlet has a ToUniversalTime method built in to do this for you. This can be displayed by running the following:

(Get-Date).ToUniversalTime()

Perhaps one of the biggest date and time–related events in computers after Y2K was the issue of daylight savings time. In the United States, for example, the beginning and end of daylight savings time was shifted, so any system that was time-sensitive needed to be aware of this change. One of the nice methods included with the Get-Date cmdlet is the IsDaylightSavingTime method. It returns whether the current date and time are adjusted for daylight savings time in the current locale:

(Get-Date).IsDaylightSavingTime()

Whenever I write scripts that generate log files, I typically like to give them names that are based on the current date and time. Not only does this guarantee uniqueness, but it also lets me quickly determine when a log file was created. Let’s say I wanted to generate a string that represents the time so I could later use it in a file name. I could use the following script to get the job done:

$filename = "myfile"
$datestring = Get-Date -uformat %Y%M%d
$newfilename = $filename + "_" + $datestring + "_.txt"
Write-Host $newfilename

You should be able to follow this code snippet. I define a file name and then generate a string that represents the current date. I then combine the file name with an underscore character, the date string, and the file extension to generate a new file name that is output to the screen. The interesting part is the second line. The -uformat switch of the Get-Date cmdlet technically stands for UNIX format. It’s not that we will use this in UNIX, but that you can then define how the date will be presented using a set of modifiers. In this case %Y represents a four-digit year such as 2007, %M represents a two-digit month such as 09, and %d represents a two-digit day such as 21.

**NOTE** For this particular switch cmdlet, the date formatters are case-sensitive. %Y is very different from %y, so be careful to use the correct case.
The following list shows some of the potential values you can use. Remember that you must prefix each character with the percent (%)) sign in order for them to work, and remember that case is very important.

- C (capital C) Century of the year. It uses the first two digits of the year, such as 20 for 2007.
- Y Four-digit year
- y Two-digit year
- b Abbreviated month name
- B Full month name
- M Two-digit month
- W (capital W) Week of the year (00–52)
- V (capital V) Week of the year (01–53)
- a Abbreviated day of the week
- A Full day of the week
- u Day of the week as a number starting with 1 for Monday
- d Two-digit day of the month
- j Day of the year
- r Time in 12-hour format
- R Time in 24-hour format (no seconds)
- T Time in 24-hour format
- p A.M. or p.m.
- Z (capital Z) Timezone offset from UTC
- H Hour in 24-hour format
- I Hour in 12-hour format
- m Minutes
- S (capital S) Seconds

Calculating dates is another one of those useful date and time functions. For example, wouldn’t you like to know what the date and time will be three months from now? How about 145 hours from now? This is a no-brainer with Windows PowerShell. To answer those two questions, you simply need to run these:

```powershell
(Get-Date).AddMonths(3)
(Get-Date).AddHours(145)
```

Get-Date has methods to add seconds, minutes, hours, days, months, and years called AddSeconds, AddMinutes, AddHours, and so on. What happens if you want to find out the date and time of an event before the current time? There is no SubtractSeconds or
MinusSeconds method in Get-Date. But the process is simple, really: Subtraction is nothing more than addition of a negative number, so to find out the date and time 30 hours ago or two years ago, you simply run one of the following:

(Get-Date).AddHours(-30)
(Get-Date).AddYears(-2)

You can set the system time using the Set-Date cmdlet. To specify a specific date and time to set it to, you can use the -date switch and pass in the date and time as a string, like so:

Set-Date -date "9/5/2007 9:00 AM"

If your computer clock is running 2 hours late, you could type in the new date and time, or you could simply rely on your trusty Get-Date cmdlet to help you out:

Set-Date (Get-Date).AddHours(2)

Sometimes you need to calculate the difference between two times—such as if you are timing the execution of a script, maybe even your login script. This is done using the New-Timespan cmdlet. It takes a start time and an end time and stores an object that calculates the timespan in values from milliseconds all the way to days. For example, if you wanted to time a script’s execution, you could use something like this:

$starttime = Get-Date
...Do lots of stuff here...
$endtime = Get-Date
$timediff = New-TimeSpan $starttime $endtime
Write-Host $timediff.milliseconds + " milliseconds!"

Notice how we have used the milliseconds property of a time span to display the number of milliseconds that elapsed. You have the following options:

- Days
- Hours
- Minutes
- Seconds
- Milliseconds
- Ticks
- Total Days
- Total Hours
- Total Minutes
- Total Seconds
- Total Milliseconds
What if you wanted to know the time span between the current date and time and
January 1, 2001? Getting the current date and time is easy, but so is representing January 1,
2001, since you can use the `Get-Date` cmdlet to help you with this:

```powershell
Get-Date -month 1 -day 1 -year 2001
```

To put it all together, you simply need to use `New-TimeSpan` to end up with this:

```powershell
New-TimeSpan $(Get-Date) $(Get-Date -month 1 -day 1 -year 2001)
```

Notice how I had to surround the calls to `Get-Date` with parentheses. This is because I’m instructing Windows PowerShell to run the command within those parentheses first and then use the value returned by it. If I’d used only `$Get-Date`, I would simply define a variable called `Get-Date` rather than call the `Get-Date` cmdlet.

**CHAPTER SUMMARY**

This chapter covered the basics of loading Windows PowerShell onto your Windows Server 2008 server and familiarizing yourself with its command-prompt–like interface. It discussed many of the basic elements needed to use Windows PowerShell, such as defining variables, creating conditional statements, and using loops for repetitive tasks. Finally, you saw firsthand how you can use PowerShell to perform useful tasks that are important to Windows administrators. Needless to say, Windows PowerShell’s uses are nearly endless due to its ability to be extended by making .NET-enabled interfaces available on your server. In fact, the trend at Microsoft is to make all its major Enterprise-based software manageable through Windows PowerShell. We have seen this already with Systems Center Operations Manager 2007 and Exchange 2007.

Hopefully, this chapter has encouraged you to seek out more information about this amazing new shell. In fact, while you’re at it, pick up a book or two on PowerShell. One of best things about Windows PowerShell is that it runs not only in Windows Server 2008 but also in Windows XP SP2, Windows Server 2003, and Windows Vista, so you can reuse this knowledge in managing all your other Windows operating systems. Even if you don’t like writing scripts, but like to manage tasks manually, it’s clear to see that Windows PowerShell does give administrators a leg up when trying to get more done in less time. Even fairly tricky actions such as moving and renaming files in bulk can be done with nothing more than a single line of PowerShell commands.

Using Windows PowerShell, you can enjoy the benefits of a command shell truly designed for Windows administrators that give you the flexibility and control that UNIX admins have had for decades. Graphical interfaces are nice, and you might prefer to use wizards, and that’s all fine. However, when it comes to doing lots of tedious and repetitive tasks quickly and efficiently, nothing beats a well-written script, and Windows PowerShell can get you there faster than ever before.
INDEX

A

a values, Windows PowerShell Dates, 456
Accept/Announce All Routes In the Range Listed option, RIPv2 Properties Security tab, 369
Accept/Announce All Routes option, RIPv2 Properties Security tab, 369
access services, network. See Network Access Protection

Accounting database, WSRM, 215, 231–235
accounts database, NAP, 259
Action field, RIPv2 Properties Security tab, 369
Actions pane, IIS Manager, 190
Actions tab, Task Scheduler, 72
Activate Authentication/Password setting, RIPv2 Properties General tab, 368
Activate Server Wizard, TS Licensing, 297–298
activation
Server Core server, 33–34
Terminal Services Licensing, 295
Active Directory Certificate Services, 409–413
Active Directory Domain Controllers, 132

$? variable, 452
$_ variable, 452
/? switch, 16
++ operators, 448
802.1X enforcement, Network Access Protection, 257

802.1X enforcement, Network Access Protection, 257
Active Directory Domain Services (AD DS). See also Active Directory Domain Services (AD DS) installation options
  application data partitions, 104
  auditing, 133–135
  backup and recovery, 137–141
  and DNS, 105, 332
  domain and forest functional levels, 105–106
  Enterprise CA installation, 405
  Flexible Single Master Operations, 102–103
  Installation Wizard, 107
  installing role in Server Core, 41–46
  migration strategies, 141–142
  organization of, 99–100
  organizational units, 100–102
  overview, 96–98
  read-only domain controller, 135–137
  removing, 126–130
  requirements, 106
  restartable, 132
  sites, 103–104
  trusts, 100
  unattended installation, 130–131
  verifying installation, 126
  WDS installation requirements, 149
Active Directory Domain Services (AD DS) installation options
  overview, 107
  WS 2008 domain controller on existing domain from restored backup media, 122–126
  WS 2008 domain in WS 2008 forest, 108–114
Active Directory domains, 78

Active Directory Services Restore Mode Administrator Password screen, AD DS Installation Wizard, 113
Active Directory sites, 118
active partitions, 338
Active Tasks section, Task Scheduler summary, 68–69
Add Counters dialog boxes
  Performance Monitor, 242–243
  Resource Monitor, 238
Add Features task, 9
Add Features Wizard, 57, 59, 217–218, 435–436
Add Image Wizard, 153–154
Add Role Services Wizard, 57, 296–297, 323–324
Add Roles task, 9
Add Roles Wizard
  AD Certificate Services installation, 410–413
defined, 57
  DHCP installation, 272
  IIS 7.0 installation, 179–180
  Network Policy Server installation, 265
  opening, 59
  Routing and Remote Access installation, 357
  Terminal Services installation, 291–293
  TS Gateway Role installation, 304–305
Add Schedule Item dialog box, Windows System Resource Manager, 231
Added Cost for Routes setting, RIPv2 Properties General tab, 367
Additional Domain Controller Options screen, AD DS Installation Wizard, 111
Additional Options screen, AD DS Installation Wizard, 118, 122, 125
address bar, IIS Manager Console, 190
AD DS. See Active Directory Domain Services
administration, Internet Information Services
delegated, 200–203
remote, 192–194
using APPCMD.EXE, 194–199
administration server, NAP, 258
Administrator certificate template, 407
administrator passwords, 8–9, 31
AdministratorPassword parameter, 43
Adprep /domainprep command, 120–121
Adprep /domainprep /gpprep command, 120
Adprep /forestprep command, 114–115, 120–121
Adprep /rodcprep command, 120
adprep.exe command-line tool, 120
[adsi] variables, 445
Advanced Security MMC snap-ins, 37
Advanced tabs
  RIPv2 Properties, 371
  rule properties boxes, 84
  WDS server Properties dialog box, 152
Advertisement Lifetime setting, IPv4
  Network Interface, 364
agents, Network Access Protection, 258
AIK (Automated Installation Kit), 161–162, 166
aliases
  of application objects, 14
  of inheritable objects, 15
  of PowerShell commands, 436–437
Allow Rules, Windows Firewall, 80–81
AllowDomainControllerReinstall parameter, 45
AllSigned execution policy, 442
Answer File panes
  SIM, 165
  Windows Server, 171
answer files, 131, 172–174
APIs (application programming interfaces), 179
APPCMD.EXE command-line tool, 179, 194–199
Appcmd.exe list SITE command, 195
application class inheritable objects, 16
application data partitions, AD DS, 104
application elements, 16
application objects, 13–15
application pools, 189, 198
application programming interfaces (APIs), 179
application types, 14
ApplicationHost.config files, 199
ApplicationPartitionsToReplicate parameter, 46
applications
  health and performance of in IIS
  automatic failed request tracing, 205–211
  overview, 204
  Runtime Status & Control API, 204–205
Terminal Services
  installing, 318–323
  overview, 317–318
  requirements, 318
archiving Accounting data, WSRM, 235
$Args variable, 452
arrays, Windows PowerShell, 445–446
ASP component, IIS, 183
ASP.NET component, IIS, 183
At Log on trigger option, Task Scheduler, 72
At Startup trigger option, Task Scheduler, 72
At Task Creation/Modification trigger option, Task Scheduler, 72
attributes, logging, 133–134
Audit Directory Service Access policy, 134
auditing Active Directory Domain Services, 133–135
Auditing tab, Certificate Authority Properties, 418
auditpol.exe command-line tool, 135
auditSystem component pass, 167
auditUser component pass, 167
Authenticated Bypass Rules, Windows Firewall, 80
Authenticated Bypass, Windows Firewall, 78
Authenticated Session certificate template, 407
authentication, BitLocker, 337–338
Authentication Exemption Rules, Windows Firewall, 89
Authentication Methods pane, RRAS Security management console, 391
Authentication Rules, Windows Firewall, 88–89
authoritative restores, Active Directory, 140–141
authorization, DHCP Server, 39–40
Authorization Policies Wizard, 309–310, 312
AutoConfigDNS parameter, 43
Automated Installation Kit (AIK), 161–162, 166
automatic failed request tracing, IIS, 205–211
Automatic Updates, 34–36
automatic variables, 452
auto-static update mode, Routing Information Protocol, 357

baseline images, 147
Baseline Performance Metrics Data Collector Set dialog box, Performance Monitor, 244–245
Baseline Performance Metrics Properties dialog box, Reliability and Performance Monitor, 250–251
Basic Authentication component, IIS, 184
Basic EFS certificate template, 407
BCD. See Boot Configuration Data
ced /delete command, 22
BCDEdit tool, Boot Configuration Data, 16–22
BDCs (backup domain controllers), 135
bindings parameter, APPCMD.EXE, 196–197
BIOS-based operating systems, BCD in, 10
BitLocker Drive Encryption architecture, 337–344
initializing, 344–349
installation command, 47
overview, 332, 336
recovery, 350
requirements, 336
turning off or uninstalling, 351
Block All Connections Rule, Windows Firewall, 81
Block Rules, Windows Firewall, 80–81
[boolean] variables, 445
Boot Configuration Data (BCD) elements, 16
modification methods, 18–22
objects, 13–16
overview, 10
stores, 10–13
boot entry, default, 19
boot images, types of, 152–153, 163
boot images, Windows Deployment Services, 153–154
boot loaders, 13
Boot Manager, Windows, 12–14, 18, 20
boot sequences, 18–19

▼ B

B values, Windows PowerShell
Dates, 456
b values, Windows PowerShell
Dates, 456
background zone loading, DNS, 333
backup domain controllers (BDCs), 135
Backup feature installation command, 47
Backup, Windows Server, 91–94, 124, 138
backups
Active Directory Domain Services, 137–141
Boot Configuration Data, 20–21
bare installation, automated, 147
Boot tab, WDS server Properties dialog box, 152
boot.ini files, 10, 14
/bootsequence switch, 18–19
break statements, 450–451
bridgehead servers, 104
built-in commands, PowerShell, 439–441

Public Key Infrastructure, 402, 404, 423–428
Terminal Services Gateway, 305–308
Certification Authorities (CAs), 404–406, 422–424
Certification Authority Backup
Wizard, 422
Certification Authority Management Console (MMC)
backing up CAs, 422–423
overview, 413–421
renewing CA certificates, 423–424
Certification Authority Restore
Wizard, 423
CGI component, IIS, 183
Challenge Handshake Authentication Protocol (CHAP), 382
Change Administrator Account link, Server Summary, 59
Change System Properties link, Server Summary, 58–59
CHAP (Challenge Handshake Authentication Protocol), 382
[char] variables, 445
child domain names, 117
ChildName parameter, 43
ciphertext, 402
clean installation, Server Core, 27
/cleanup switch, 22
clear key authentication, BitLocker, 338
client access licenses (CALs), 294–295, 299–301
Client Certificate Mapping Authentication component, IIS, 184
Client tab, WDS server Properties dialog box, 152
clients, Network Access Protection configuring, 281–283
defined, 261–262
testing, 283
cmdlets, PowerShell, 434, 439–441
Command Prompt option, System Recovery Options screen, 339–340

C values, Windows PowerShell Dates, 456
CA Exchange certificate template, 407
Calculator application, 320–321
Calendar, Windows System Resource Manager, 216, 228–231
Callback Control Protocol (CBCP), 382
CAs. See Certification Authorities
Policies, 302, 309–311
Capture boot images, WDS, 152, 154–155
CAs. See Certification Authorities
case sensitivity, command, 47
CBCP (Callback Control Protocol), 382
CCP (Compression Control Protocol), 382
CDs, burning Discover boot images on, 161–162
CEIP (Customer Experience Improvement Program), 60
CEP Encryption certificate template, 407
Certificate Import Wizard, 306–307, 314
Certificate Managers tab, CA Properties, 420
Certificate Revocation List (CRL), 404
certificate templates, 406–409
Certificate Templates folder, CA management console, 414
certificates
digital, 404
of health, 257–258
command prompts, Server Core installation, 27
command-line switches, 16
command-line tools, 27, 58
commands
  Automatic Updates, 35
to install File Server roles, 40
PowerShell, 436–437, 439–441
Server Core Optional Features installation, 47
committed memory, 226
communications flow, Network Access Protection, 263–264
comparison operators, 446–447
component configuration passes,
  Windows SIM, 165, 167
Compression Control
  Protocol (CCP), 382
Computer certificate template, 407
<condition> section, FOR statements, 448
conditional forwarders, DNS, 333
conditional statements, PowerShell, 446–447
Conditions tab, Task Scheduler, 73
Conditions, Windows System Resource Manager, 235–236
configuration
  Automatic Updates, 34–36
  Dynamic Host Configuration Protocol, 271–281
  Internet Information Services, 197–200
  NAP client, 281–283
  Network Address Translation, 374–377
  network interfaces for routing, 359–361
  Network Policy Server, 266–271
  RIPv2, 362–366
  Routing and Remote Access Services, 358–359, 385–394
  Server Core network interfaces, 31–33
  Server Core roles, 38–46
  Terminal Services, 288–290, 319–329
  TS Gateway, 303, 307–313
  VPN using PPTP, 394–398
  Windows Deployment Services, 148–151
  Windows Firewall, 36–38
  Windows Server 2008, 8–10
Configuration Names, IIS Property Grid layout, 190
configuration passes, Windows SIM, 165, 167
Configuration snap-in, Server Manager
  Local Users and Groups/Device Manager, 90
  overview, 67–68
  Task Scheduler, 68–77
  Windows Firewall with Advanced Security, 77–89
  WMI Control, 89–90
Configuration Wizard, WDS, 148, 150–151
Configure Accounting View Filter dialog box, WSRM, 233
Configure IE ESC link, Server Manager
  Security Information section, 59
Configure Networking task, 9
Configure Remote Desktop link, Server Summary, 59
Configure Updates link, Server Manager
  Security Information section, 59
Configure Windows Firewall task, 9
ConfirmGC parameter, 43
Connection Authorization Policies (CAPs), 302, 309–311
connection profiles, Windows Firewall and Advanced Security summary pane, 80–81
Connection Security Rules, Windows Firewall, 80
Connection Security, Windows Firewall with Advanced Security, 88–89
Connections pane, IIS Manager, 188–190, 206
continue statements, 450–451
Control Panel applets, Server Core, 48
counters, Reliability and Performance Monitor, 240
CPU allocation policies, WSRM, 224–228
CPU utilization, Resource Monitor, 236–239
Create a Self-signed Certificate dialog box, Terminal Services Gateway, 306
Create events, Audit Directory Service Access Policies, 134
Create New Data Collector Set dialog box, Performance Monitor, 243–244
CreatePartition Properties pane, Windows Server, 169
credential caching, RODCs, 136–137
Credential Security Service Provider (CredSSP), 36
credentials delegation, Terminal Services, 289–290
criteria names, WSRM, 220
critical volumes, Window Server Backup, 138
CriticalReplicationOnly parameter, 43
CRL (Certificate Revocation List), 404
Cross-Certification Authority certificate template, 408
cryptographic service providers (CSPs), 406
CScript commands, 36
CSPs (cryptographic service providers), 406
current boot entry, 14
current execution policies, Windows PowerShell, 442
custom event log views, Server Manager Diagnostics snap-in, 63–65
Custom Logging component, IIS, 184
Custom Rules, Windows Firewall, 89
Custom View dialog box, Server Manager, 63–64

Customer Experience Improvement Program (CEIP), 60

D

data collector sets, Reliability and Performance Monitor, 242–246, 250–252
data partitions, Active Directory Domain Services, 104
data protection, 336
Database Path parameter, 43
dates, working with in Windows PowerShell, 454–458
[datetime] variables, 445
daylight savings time, 455
DCOM (Distributed Component Object Model) interface, 215
dcpromo.exe, 41–42, 107
decryption
domain, 127
drive, 351
default boot entry, Boot Configuration Data, 14, 19
Default Document component, IIS, 182
default routes, 356
Default Rules, Windows Firewall, 80
default settings, Windows PowerShell, 442
/default switch, 19
delimited administration, IIS, 200–203
/delete switch, 22
delta CRLs, 427–428
DemoteFSMO parameter, 45
Desktop Experience feature, Terminal Services, 287
device elements, BCD, 16
Device Manager, 63, 90
device objects, BCD, 16
DHCP. See Dynamic Host Configuration Protocol
DHCP tab, WDS server Properties dialog box, 152
Diagnostics snap-in, Server Manager, 62–67
dialog pages, IIS Manager Home pane, 190
dial-up networking (DUN), 381–382
Digest Authentication component, ISS, 184 digests, 403
digital certificates. See certificates
digital signatures, 403
DIR command, 440, 453
directly connected routers, 354–355
Directory Browsing component, IIS, 182
Directory E-mail Replication certificate template, 408
Directory Service event log, 125
Directory Services Restore Mode (DSRM), 132, 139
Directory Services tab, WDS server Properties dialog box, 152
Disable Subnet Summarization property, RIPv2 Properties Advanced tab, 371
DisableCancelForDnsInstall parameter, 43
Discover images, Windows Deployment Services, 153, 159–162
Disk Management snap-in, Server Manager, 90, 94
Disk Properties pane, Windows Server, 169
diskpart command, 339–342
/displayorder switch, 19
Distributed Component Object Model (DCOM) interface, 215
Distributed File System installation command, 40
Distributed File System Replication installation command, 40
Distribution Share pane, SIM, 165
DNS. See Domain Name System
Dnscmd commands, 38–39
DNSDelegation parameter, 45
DNSDelegationPassword parameter, 45
DNSDelegationUserName parameter, 45
DNSSOnNetwork parameter, 43
DO...WHILE statements, 449–450
dollar signs, 443
domain controller audit policies, 133
Domain Controller Authentication certificate template, 408
Domain Controller certificate template, 408
Domain Controller service, AD DS, 132
domain controllers
failure, 137
overview, 102–103
removal of, 126–127
replacement of, 141–142
Domain firewall profile, 78
domain forests, 99–100
domain functional levels, AD DS, 105–106, 117
Domain Name System (DNS)
Active Directory Domain Services, 105
Active Directory reliance on service, 97
enhancements to, 332–335
installing Windows Server domain, 117
RODCs, 137
Server Core Server role, 38–39
servers, 32–33
verifying installation, 126
WDS installation requirements, 149
Domain Naming Master Role, Active Directory, 103
domain profile commands, 37
domain trees, 99–100
DomainLevel parameter, 46
DomainNetBiosName parameter, 43
domains, Server Core, 34
[double] variables, 445
Download and Install Updates task, 9
drive encryption. See BitLocker Drive Encryption
drivers, loading, 48–49
drvload command, 48
DSRM (Directory Services Restore Mode), 132, 139
DUP (dial-up networking), 381–382
DVDs, burning Discover boot images on, 161–162
Dynamic Content Compression component, IIS, 185
Dynamic Host Configuration Protocol (DHCP)
  integration with RRAS, 389
  NAP Enforcement Client, 262–263
  NAP using enforcement by, 265–283
  relay agents, 362, 372–373
  Server Core Server role, 39–40
  WDS installation requirements, 149
dynamic routing, Routing and Remote Access Services, 356–358
dynamic updates, Domain Name System, 333

E
EAPHost NAP Enforcement Client, 262
EFI (Extensible Firmware Interface)–based operating systems, 10
EFS Recovery Agent certificate template, 408
elements
  Boot Configuration Data, 11, 16
  Server Manager, 56–58
Enable Automatic Updating and Feedback task, 9
Enable Fragmentation Checking setting, IPv4 Network Interface, 364
Enable IP Router Management setting, IPv4 Network Interface, 364
Enable Poison-Reverse Processing property, RIPv2 Properties Advanced tab, 371
Enable Remote Desktop task, 9
Enable Router Discovery
  Advertisements setting, IPv4 Network Interface, 364
Enable Triggered Updates property, RIPv2 Properties Advanced tab, 371
encapsulation, Virtual Private Networks, 383–388
encryption, 402–403. See also BitLocker Drive Encryption
encryption keys, BitLocker, 337–338
Enforcement Clients, NAP, 262
Enforcement Servers (ES)
  Network Access Protection, 263
  Windows Access Protection, 262–263
Enhanced Security Configuration (ESC), Internet Explorer, 58
Enrollment Agent certificate template, 408
Enrollment Agents tab, Certificate Authority Properties, 415
Enterprise Certification Authorities, 405
enterprise public key infrastructure certificate revocation, 426–428
  certificate templates, 406–409
  Certification Authorities, 404–406
  Certification Authority MMC, 413–424
cryptographic service providers, 406
digital certificates, 404
digital signatures, 403
issuing certificates, 425–426
overview, 402–403
recovery keys, 409–413
entries, manipulating Boot Configuration Data, 21–22
/enum switch, 17
-eq comparisons, 447
Equal per process management rule, WSRM, 225
Equal per user management rule, WSRM, 225
Equal_Per_IISAppPool resource allocation policy, 223
Equal_Per_Process resource allocation policy, 223
Equal_Per_Session resource allocation policy, 223
Equal_Per_User resource allocation policy, 223
Error Reporting, 60
ES (Enforcement Servers)
  Network Access Protection, 263
  Windows Access Protection, 262–263
ESC (Enhanced Security Configuration), Internet Explorer, 58
event IDs, Audit Directory Service Access Policies, 134
event logs
  Directory Service, 125
  Server Manager Diagnostics snap-in, 63–67
  Windows Security, 134
Event Viewer, Server Manager Diagnostics snap-in, 62–63
events, WSRM Calendar, 228–230
Exchange 2003, 98
Exchange Enrollment Agent certificate template, 408
Exchange Signature Only certificate template, 408
Exchange user certificate template, 408
Execute mode, Terminal Services, 318–319
exit modules, 415
/export switch, 20
Extensible Firmware Interface (EFI)–based operating systems, 10
Extensions tab, CA Properties, 420

F
failed request tracing, IIS, 205–211
Failed Requests folder, CA management console, 414
Features snap-in, Server Manager, 62
Features Summary section, Server Manager, 59–60
Features View, IIS Manager Home pane, 189–190
File Replication event log, 125
File Replication installation command, 40
File Services roles
  Server Core, 40
  Server Manager, 61
Filter After Grouping option, WSRM Accounting, 235
Filter Before Grouping option, WSRM Accounting, 234
filter criteria, WSRM, 220
filters, WSRM Accounting view, 233–234
Firewall Monitoring, Windows Firewall with Advanced Security, 89
Firewall, Windows. See Windows Firewall with Advanced Security
Firmware Boot Manager, 14
Flexible Single Master Operations (FSMO), 102–103
floppy disks, password recovery, 10
FOR statements, 448
FOREACH statements, 448–449
foreach variable, 452
forest functional levels, AD DS, 105–106
ForestLevel parameter, 46
forests, domain, 99–100
formatting, Windows PowerShell, 440
fragmented packets, 364
Friendly Names, 190
FSMO (Flexible Single Master Operations), 102–103
Full server installation, 3

G
Gateway, Terminal Services architecture, 302–303
certificates, 305
configuring to use certificates, 307–308
connecting to terminal server using client through, 313–316
Connection Authorization Policies, 309–311
installing certificates on server, 306–307
installing Role, 304–305
and NAP, 317
overview, 302
Resources Authorization Policies, 311–312
self-signed certificates, 305–306
GC (global catalog), 98
-get comparisons, 447
General option, Routing Information Protocols, 360
General tabs
Certificate Authority Properties, 414
RIPv2 Properties, 366–368
RRAS management console, 391
rule’s Properties dialog box, 84
Task Scheduler, 70–71
WDS server Properties dialog box, 152
generalize component pass, 167
Generic Routing Encapsulation (GRE) protocol, 384
Get-ChildItem cmdlet, 437
Get-Date cmdlet, 454–458
Get-Help cmdlet, 437
Get-ItemProperty cmdlet, 453–454
Get-Process cmdlet, 438
Get-Service cmdlet, 438, 451
Getting Started section, Windows Firewall and Advanced Security summary pane, 82
Get-WmiObject cmdlet, 438
global Audit Policy, AD DS, 134–135
global catalog (GC), 98
global unique identifiers (GUIDs), 13–15
GlobalNames zone, DNS, 334
Go to Manage Roles link, Roles Summary, 59
Go to Windows Firewall link, Server Manager Security Information section, 59
GP alias, 454
GPOs (Group Policy objects), 425
/gpprep switch, 120
graph interface, Resource Monitor, 236
GRE (Generic Routing Encapsulation) protocol, 384
Group Items option, Windows System Resource Manager Accounting, 235
Group Policy Management feature, 134
Group Policy objects (GPOs), 425
-gt comparisons, 447
GUIDs (global unique identifiers), 13–15

\n
H

H values, Windows PowerShell Dates, 456
hard disk space requirements, Server Core, 27
hard limits, WSRM, 225
hardware requirements
optional features, 47
Windows Server 2008, 2
health agents, 261–262
health certificates, 257–258
health policies, NAP, 259
Health Registration Authority (HRA), 256–257, 259
History tab, Task Scheduler, 73
Home pane, IIS Manager, 189–190
$Home variable, 452
$Host variable, 452
HRA (Health Registration Authority), 256–257, 259
HTTP Errors component, IIS, 182
HTTP Logging component, IIS, 183
HTTP Redirection component, IIS, 182
hyphens, 446–447
I values, Windows PowerShell Dates, 456
IAS (Internet Authentication Service), 258, 408
id parameter, APPCMD.EXE, 196
Idx parameter, 32
IE ESC (Internet Explorer Enhanced Security Configuration), 58
IF/ELSEIF statements, 446–447
IGMP (Internet Group Management Protocol), 361–362, 373
Ignore/Do Not Announce All Routes In the Ranges Listed option, RIPv2 Properties Security tab, 369
IgnoreIsLastDcInDomainMismatch parameter, 45
IgnoreIsLastDnSServerForZone parameter, 46
IIS. See Internet Information Services
IISAppPool process matching criteria, 219
Image Capture Wizard, 152, 155
Image Capture Wizard, WDS, 157–161
image types
   BCD object, 14
   WDS, 152
implementation, Network Access Protection, 260
/import switch, 20–21
Inbound Filtering, Windows Firewall with Advanced Security, 78
Inbound Filters setting, IPv4 Network Interface, 364
Inbound Rules, Windows Firewall with Advanced Security, 37, 83–88
Include Default Routes In Sent Announcements property, RIPv2 Properties Advanced tab, 371
Include Host Routes In Sent Announcements property, RIPv2 Properties Advanced tab, 371
Incoming Packet Protocol setting, RIPv2 Properties General tab, 367
Infrastructure Master Role, Active Directory, 103
inheritable objects, 15
/init> section, FOR statements, 448
Initial Configuration Tasks screen, Server Manager, 57
Initialize TPM Security Hardware Wizard, 345
Install From Media screen, AD DS Installation Wizard, 125
Install images, WDS, 152–153, 156–159, 162–164
Install Licenses Wizard, 300
Install mode, Terminal Services, 318–319 installation
   Active Directory Certificate Services, 409–413
   Active Directory Domain Services, 107–126
   BitLocker, 338–344
   Dynamic Host Configuration Protocol, 271–281
   IIS 7.0 features, 179–181
   Network Address Translation, 374–377
   Network Policy Server, 265–266
   remote IIS administration, 192–194
   RIPv2, 362–366
   Routing and Remote Access Services, 357–358
   Server Core, 27–30
   Server Core roles, 38–46
   Terminal Services, 291–293, 296–301, 318–329
   TS Gateway, 303–307
   Windows Deployment Services, 148–151
   Windows PowerShell, 434–436
   Windows Server 2008, 2–10, 163–164
   Windows System Resource Manager, 217–218
installation commands
  for File Server roles, 40
  for Server Core optional features, 47
installation media, 149
Installation Wizard, AD DS
  installing Active Directory from
  restored backup, 124–126
  overview, 107
  removing AD DS from last domain
  controller, 127–130
  WS 2008 domain controller in
  Windows 2000/2003 domain,
  121–122
  WS 2008 domain in Windows
  2000/2003 forest, 116–119
  WS 2008 domain in WS 2008 forest,
  108–114
[int] variables, 445
integration, Domain Name System, 333
interactive installation, Server Core,
  28–30
international settings applet, Server
  Core, 48
Internet Authentication Service (IAS),
  258, 408
Internet Explorer (IE) Enhanced Security
  Configuration (ESC), 58
Internet Group Management Protocol
  (IGMP), 361–362, 373
Internet Information Services (IIS) 7.0
  administration using APPCMD.EXE,
  194–199
deleagated administration, 200–203
management console, 188–192
overview, 178–181
remote administration, 192–194
server and application health and
  performance, 204–211
unattended installation, 181–188
XCOPY deployment, 211
Internet Protocol (IP), Point-to-Point
  Protocol, 382
Internet Protocol Control Protocol
  (IPCP), 382
Internet Protocol Security (IPSec)
  certificate template, 408
  NAP enforcement, 256–257
  NAP Enforcement Client, 262–263
  NAP Enforcement Server, 263
Internet Protocol version 4 (IPv4)
  addresses, 334
  installing and configuring RIPv2
  for, 362–366
  Network Interface General
    Properties, 364
    Properties dialog box, 276
  protocols, 360–361
Internet Protocol version 6 (IPv6)
  address configuration, 31–33
  addresses, 334
  DNS support, 334
  protocols, 360
  support, 78
inter-site replication traffic, 104
intra-site replication traffic, 104
IP and Domain Restrictions component,
  IIS, 185
IPSec Settings tab, Windows Firewall
  and Advanced Security summary
  pane, 81
ISAPI Extensions component, IIS, 183
ISAPI Filters component, IIS, 183
IsDaylightSavingTime method, 455
IsLastDCInDomain parameter, 44
Isolation Rules, Windows Firewall, 89
Issue and Manage Certificates permission,
  Certificate Authority MMC, 418
Issued Certificates folder, Certificate
  Authority MMC, 413
Itanium processors, 351–352
\[ J \]

j values, Windows PowerShell Dates, 456

\[ K \]

Kernel Scheduler, WSRM, 216
Key Recovery Agent certificate template, 408
keys
- BitLocker startup and recovery, 338
- registry, 453

\[ L \]

$LASTEXITCODE variable, 452
Layer 2 Tunneling Protocol/Internet Protocol Security (L2TP/IPSec), 384–385
LCP (Link Control Protocol), 382
LDAP (Lightweight Directory Access Protocol), 98
- le comparisons, 447
Legacy Windows Loader, 14
Level of Preference setting, IPv4 Network Interface, 364
library class inheritable objects, 16
library elements, 16
Library folder, Task Scheduler, 69
Licensing, Terminal Services activating server, 297–299
- installing Client Access Licenses, 299–301
- installing role service, 296–297
license types, 294–295
licensing modes, 292–293
overview, 294–301
life spans, certificate, 407
Lightweight Directory Access Protocol (LDAP), 98
Link Control Protocol (LCP), 382

- link-local multicast name resolution (LLMNR), 335
- List page layout, IIS Manager Home pane, 190
- list SITE command, APPCMD.EXE, 196
- Local Users and Group, Server Manager, 90
- Location For Database, Log Files, and SYSVOL screen, AD DS Installation Wizard, 111, 118, 122
- Location-Aware Profiles, Windows Firewall with Advanced Security, 78
- log files, 455
- Logging tab, RRAS management console, 393
- Logging Tools component, IIS, 183
- logical network zones, NAP, 255
- LogPath parameter, 44
- logs, event, 63
- loops, PowerShell, 448–451
- -lt comparisons, 447

\[ M \]

m values, Windows PowerShell Dates, 456
M values, Windows PowerShell Dates, 456
Machine.config files, 199
malicious users, 259–260
Manage CA permission, Certificate Authority MMC, 418
managed processes, 216
management consoles
- IIS, 188–192
Management Scripts and Tools role service, IIS, 205
Management Service page, IIS, 193–194
managing policies, 222
member servers, 102
Memory Diagnostic tool, 13, 20
memory limits, WSRM, 225–226
Memory Manager, WSRM, 216
memory resource allocation, WSRM, 224
memory tester, Windows, 14
Memory Usage statistics, Resource Monitor, 236–239
menu display order, BCD, 19–20
Messages pane, SIM, 165
metrics gathering tools, 214
Microsoft Failover Clustering installation command, 47
Microsoft Point-to-Point Encryption (MPPE), 384
Microsoft Root Certificate Members Program, 305
migration strategies, AD DS, 141–142
minute values, Windows PowerShell, 456
MMC. See Certification Authority Management Console
modification methods, BCD, 18–22
Modify events, Audit Directory Service Access Policies, 134
ModifyPartition Properties pane, Windows Server, 170
month values, Windows PowerShell, 456
Move events, Audit Directory Service Access Policies, 134
MPPE (Microsoft Point-to-Point Encryption), 384
MSConfig.exe GUI, 17
MSI (Windows Installer), 47–48, 317
MSI files, 47–48, 317
multi-boot screen, Windows Boot Manager, 12
multilinking, 382
multi-master configuration, 103
Multipath IO installation command, 47

N
name parameter, APPCMD.EXE, 196
names
  criteria, 220
  policy, 226

NAP. See Network Access Protection
NAT (Network Address Translation), 362, 373–377
-ne comparisons, 447
Neighbors tab, RIPv2 Properties, 370
.NET Extensibility component, IIS, 183
.NET managed code, 441–442
Net user Administrator P@ssword command prompt, 31
netdom command, 34
netdom join command, 34
Netsh command, 31–33
netsh firewall commands, 37–38
Network Access Protection (NAP)
  architecture, 260–261
  client architecture, 261–262
  communications flow overview, 263–264
  components, 256–259
  DHCP NAP Enforcement Client, 262–263
  myths about, 259–260
  overview, 253–256
  server architecture, 262–263
  and Terminal Services Gateway, 317
  using DHCP enforcement, 264–283
Network Address Translation (NAT), 362, 373–377
Network Credentials screen, AD DS Installation Wizard, 117
Network File System installation commands, 40
network interface cards (NICs), 33
network interfaces
  RRAS, 359–361
  Server Core, 31–33
Network Load Balancing installation command, 47
Network Policy Server (NPS)
  configuring, 266–271
  installing, 265–266
  management console, 267
Network Access Protection, 258
Network Settings tab, WDS server
Properties dialog box, 152
network utilization tracking, 241
New Action dialog box, Task Scheduler, 75–76
New Calendar Event dialog box, WSRM, 229
New Network Policies dialog box, NPS, 269–271
New Process Matching Criteria dialog box, WSRM, 220–221
New Resource Allocation Policy dialog box, WSRM, 226–227
New Schedule dialog box, WSRM, 231
New Scope Wizard, NPS, 275–279
New Technology File System (NTFS) partitions, 149
New Triggers dialog box, Task Scheduler, 74–75
NewDomain parameter, 44
New-Timespan cmdlet, 457–458
NICs (network interface cards), 33
/noCleanup switch, 22
non-authoritative restores, Active Directory, 139–140
-notmatch comparisons, 447
NPS. See Network Policy Server
NT domain model, Windows, 96–97
NT Loader (NTLDR), 10, 13
NTFS (New Technology File System) partitions, 149

Objects

Objects

objectname.propertyname syntax, 452
objects
Boot Configuration Data, 11, 13–16
Windows PowerShell, 440
<object-type> parameter, APPCMD.EXE, 195–196
ocsetup command, 40, 47
ODBC Logging component, IIS, 184
offlineServicing component pass, 167
On a Schedule trigger option, Task Scheduler, 72
On an Event trigger option, Task Scheduler, 72
On Connection to User Session trigger option, Task Scheduler, 72
On Disconnect from User Session trigger option, Task Scheduler, 72
On Idle trigger option, Task Scheduler, 72
On Workstation Lock trigger option, Task Scheduler, 72
On Workstation Unlock trigger option, Task Scheduler, 72
OnDemandAllowed parameter, 45
OnDemandDenied parameter, 45
one-time events, WSRM Calendar, 228
one-way trusts, Active Directory, 100
ongoing compliance, NAP, 255
oobeSystem component pass, 167
operating system images, WDS
boot images, 153–154
Capture images, 154–155
Discover images, 159–162
Install images, 156–159
overview, 152–153
Operation Mode setting, RIPv2
Properties General tab, 366
Optional boot applications BCD object, 13
organizational units (OUs), AD DS, 96–97, 100–102
Outbound Filtering, Windows Firewall with Advanced Security, 78
Outbound Filters setting, IPv4 Network Interface, 364
Outbound Rules, Windows Firewall with Advanced Security, 83–85
Outgoing Packet Protocol setting, RIPv2
Properties General Tab, 367
Out-of-box Experience (OOBE), 167
Overview section, Windows Firewall and Advanced Security summary pane, 80
p values, Windows PowerShell Dates, 456
packets, fragmented, 364
parameters, dcpromo unattended install, 42–46
parent domain names, 117
ParentDomainDNSName parameter, 44
partitions
  BitLocker, 338–342
  hard drive, 6
  network, 255
Password parameter, 44
password recovery disks, 10
passwords
  administrator, 8–10, 31
  recovery, BitLocker, 338, 350
  Restore Mode, 113
  on RODCs, 136
  Server Core, 30
PCRs (Platform Configuration Registers), 336
PDC. See (primary domain controller)
  Emulator Role, Active Directory, 103
Pending Requests folder, CA management console, 414
Per Device license mode, Terminal Services, 294–295
Per User CALs, Terminal Services, 295
Per User license mode, Terminal Services, 294–295
Performance Log Users group, Reliability and Performance Monitor, 240–241
Performance Monitor Users group, Reliability and Performance Monitor, 241
performance monitoring, 213–215. See also Reliability and Performance Monitor
performance, program, Terminal Services, 329–330
performance-metrics gathering, 214
Periodic Announcement Interval (Seconds) property, RIPv2 Properties Advanced tab, 371
periodic update mode, Routing Information Protocol, 357
permissions, security, 418
personal identification numbers (PINs), BitLocker, 337–338
physical path parameter, APPCMD.EXE, 197
PINs (personal identification numbers), BitLocker, 337–338
pkgmgr.exe command-line tool, 181, 186–188
PKI. See public key infrastructure plain text, 402
Platform Configuration Registers (PCRs), 336
Point-to-Point Protocol (PPP), 382, 393
Point-to-Point Tunnel Protocol (PPTP), 384, 394–398
poison-reverse processing, 371
policies, network. See Network Access Protection
policies, resource allocation, WSRM, 222–228
Policy Module tab, Certificate Authority Properties, 415
Policy store, WSRM, 215
Ports Properties dialog box, Routing and Remote Access management console, 394–396
PowerShell
  .NET managed code, 441–442
  basics of, 436–439
  cmdlets, 439–441
  conditional statements, 446–447
  loops, 448–451
  overview, 434–436
  real-world tasks, 451–458
  scripting and security, 442–443
  variables, 443–446
PPP (Point-to-Point Protocol), 382, 393
PPP tab, Routing and Remote Access Services management console, 393
PPTP (Point-to-Point Tunnel Protocol), 384, 394–398
pre-boot execution environment (PXE), 151–152, 163–164
preferred replication partners, 104
primary domain controller (PDC) Emulator Role, Active Directory, 103
primary read-only zones, 334
Print Server role, Server Core, 41
priority-matching algorithm, WSRM service, 216–217
priority-order chains, WSRM, 226
Private domain profile, Windows Firewall with Advanced Security, 78
Process Host Routes In Received Announcements property, RIPv2 Properties Advanced tab, 371
process matching criteria, WSRM, 219–222 processes
managed and unmanaged, 216 viewing in PowerShell, 438
ProductKey Properties, Windows Server, 172
profile-specific properties, firewalls, 37
Programs and Services tabs, rule properties dialog boxes, 84
Property Grid layout, IIS Manager Home pane, 190
protected volumes, BitLocker, 350 protection, network. See Network Access Protection
Protocols and Ports tabs, rule properties dialog box, 84 protocols, routing
DHCP Relay Agent, 372–373 installing and configuring RIPv2 for IP, 362–366
Internet Group Management Protocol, 373
Network Address Translation, 373–377 overview, 361–362 RIPv2 properties, 366–371 Provide Computer Name and Domain task, 9 $PsHome variable, 452
Public domain profile, Windows Firewall with Advanced Security, 78
public key infrastructure (PKI) certificate revocation, 426–428 certificate templates, 406–409 Certification Authorities, 404–406 Certification Authority MMC, 413–424 cryptographic service providers, 406
digital certificates, 404 digital signatures, 403
issuing certificates, 425–426 overview, 402–403
recovery keys, 409–413
PXE (pre-boot execution environment), 151–152, 163–164 PXE Response Settings tab, WDS server Properties dialog box, 152
PXE Server Initial Settings screen, WDS Configuration Wizard, 151

R

r values, Windows PowerShell Dates, 456
R values, Windows PowerShell Dates, 456
RADIUS (Remote Authentication Dial-In User Service), 258
RAPs (Resource Authorization Policies), 302, 311–313
RAS (Remote Access Server), 408
RDP (Remote Desktop Protocol), 36, 302, 317, 320
Read permissions, CA security, 418
read-only DNS zone, 334
read-only domain controllers (RODCs), 96, 107, 135–137
RebootOnCompletion parameter, 45
RebootOnSuccess parameter, 45
record types, DNS server, 39
/recordadd switch, 39
/recorddelete switch, 39
recovery
  Active Directory Domain Services, 137–141
  Windows BitLocker Drive Encryption, 350
recovery agents, 409
Recovery Agents tab, Certificate Authority Properties, 418
Recovery Console, BitLocker Drive Encryption, 350
recovery keys, 338, 409–413
recovery passwords, BitLocker, 338, 350
recurring events, WSRM Calendar, 228
reference system, Windows Server 2008, 156–159
registry, Windows, 452–454
Relative ID (RID) Master Role, Active Directory, 103
Relay Agents, DHCP, 372–373, 389
Reliability and Performance Monitor
data collector sets, 242–246
overview, 213–215, 239–241
Reliability Monitor, 246–248
reports, 248–251
remediation, 256
remediation servers, NAP, 259
RE-MINST file shares, 148
remote access, RRAS
  configuring server properties, 389–394
  configuring VPN using PPTP, 394–398
  DHCP integration, 389
dial-up networking, 381–382
overview, 381
Point-to-Point Protocol, 382
Virtual Private Networks, 383–388
Remote Access Server (RAS), 408
remote activation, Windows Server 2008, 33–34
remote administration
  IIS, 192–194
  Server Core, 36
Remote Authentication Dial-In User Service (RADIUS), 258
Remote Desktop Connection 6.0, 287
Remote Desktop Connection Gateway
  Server settings, 314
Remote Desktop Protocol (RDP), 36, 302, 317, 320
remote desktop, Windows Server, 9
remote domain controllers, 142
Remote Installation Services (RIS), 146
remote management, firewalls, 37
remote programs, Terminal Services
  installing applications, 318–323
  overview, 317–318
  requirements, 318
Remote Shell, Windows, 48
RemoteApp Wizard, 319–320
RemoteSigned execution policy, 443
Removable Storage Management
  installation command, 47
Remove Features link, Features Summary, 59
Remove Features Wizard, 58
Remove Roles link, Roles summary, 59
Remove Role Services Wizard, 57
Remove Roles Wizard, 57
RemoveApplicationPartitions parameter, 44
<repeat> section, 448
ReplicaDomainDNSName parameter, 44
ReplicaOrNewDomain parameter, 44
replication
  on RODCs, 136
  traffic, 104
ReplicationSourceDC parameter, 44
ReplicationSourcePath parameter, 44
reports, Reliability and Performance Monitor, 248–251
Request Certificates permission, CA security, 418
Request Filtering component, IIS, 185
Request Monitor component, IIS, 183
Residual process matching criteria, 219
resolution, Remote Desktop Connection, 287
resolution services, 332
resource allocation policies, WSRM, 222–228
Resource Authorization Policies (RAPs), 302
resource management, 213–215. See also Windows System Resource Manager
Resource Monitor, WSRM, 218–219, 236–239
Resource Overview, Reliability and Performance Monitor, 239–240
Resources and Support section, Server Manager, 60
Resources Authorization Policies (RAPs), 311–313
Resources section, Windows Firewall and Advanced Security summary pane, 82
restartable Active Directory Domain Services, 132
Restore Mode Passwords, Active Directory, 113
restoring Boot Configuration Data, 20–21
Restricted execution policy, PowerShell, 442
resume application, Windows, 14
Revoked Certificates folder, CA management console, 413
RID (Relative ID) Master Role, Active Directory, 103
RIPv2. See Routing Information Protocol version 2
RIS (Remote Installation Services), 146
RODCs. See (read-only domain controllers)
Role management home pages, Server Manager, 58
roles
Server Core, 38–46
Server Manager, 54–56
Terminal Services Gateway, 304–305
Terminal Services Licensing, 296–297
Roles snap-in, Server Manager, 61–62
Roles Summary section, Server Manager, 59
Root CAProve certificate template, 408
root certificate authority, 305, 404
route add command, 378–379
Route print command, 379
router discovery, 356, 364
routes, network communication, 356
Routing and Remote Access Services (RRAS)
overview, 354
remote access, 381–394
routing services, 359–381
Routing Information Protocol version 2 (RIPv2)
installing and configuring for IP, 362–366
overview, 357–358
properties, 366–371
routing tables, 356
RRAS. See Routing and Remote Access Services
RSCA (Runtime Status & Control API), 204–205
rule property boxes, Windows Firewall, 84
rule types, Windows Firewall, 80
Run key, Windows PowerShell, 453
Run Security Configuration Wizard link, Server Manager Security Information section, 59
Runtime Status & Control API (RSCA), 204–205
Send Out Advertisements Within This
Interval setting, IPv4 Network Inter-
facing, 364
Server Authorization screen, DHCP, 274
Server Core
installation, 27–30, 142
management, 47–49
optional features, 46–47
overview, 3, 26–27
post installation tasks, 30–38
role installation and configuration,
38–46
server health and performance, IIS
automatic failed request tracing,
205–211
overview, 204
Runtime Status & Control API,
204–205
Server Manager
console, 58–60
elements, 56–58
overview, 10, 52–56
snap-ins, 60–67, 90–94. See also
Configuration snap-in, Server
Manager
Server Properties dialog box, WDS, 152
Server Side Includes component, IIS, 183
Server Summary section, Server
Manager, 58–59
Server to Server Rules, Windows
Firewall, 89
servers. See also Windows Server 2008
activating Server Core, 33–34
capacity of, 329–330
configuring properties of RRAS,
389–394
connecting to using client through
TS Gateway, 313–316
Network Access Protection,
262–263
Terminal Services Licensing,
297–299
TS Gateway certificate installation
on, 306–307

S

S values, Windows PowerShell Dates, 456
SACL (System Access Control List), 135
SafeModeAdminPassword parameter, 44
SC command, 49
scale-out, application, 332
scale-up, application, 332
schedules, WSRM Calendar, 228, 231
scheduling data collector sets, Reliability
and Performance Monitor, 245–246
Schema Master Role, Active Directory, 103
schemas, Active Directory, 98, 135
Scope Filter option, WSRM
Accounting, 234
Scope tabs, rule properties boxes, 84
scregedit.wsf commands, 35–36
scripting
PowerShell, 442–443
WMI, 22
searchFlags property, attributes, 135
second values, Windows PowerShell, 456
section names, APPCMD.EXE, 200
Secure Socket Tunneling Protocol (SSTP),
384–385
security, 442–443. See also Network
Access Protection
security contexts, task, 70
Security Event Log, Windows, 134
Security Information section, Server
Manager, 59
Security permissions, Certificate
Authority, 418
Security tabs
Certificate Authority Properties, 418
RIPv2 Properties, 368–369
RRAS management console, 391
self-signed certificates, Terminal Services
Gateway, 305–306
Send Clean-Up Updates When Stopping
property, RIPv2 Properties Advanced
tab, 371
Service Hardening, Windows, 78, 80
Service Manager, Diagnostics snap-in, 63
service, WSRM, 215–218
services, managing, 438
Set Accounting Database dialog box, WSRM, 231
Set Administrator Password task, 9
Set Forest Functional Level screen, AD DS Installation Wizard, 111
Set Time Zone task, 9
Set-Date cmdlet, 457
Settings tab, Task Scheduler, 73
SHAs (System Health Agents), NAP, 258, 261–262
SHVs (System Health Validators), NAP, 259
signatures, digital, 403
SIM. See System Image Manager
Simple Network Management Protocol (SNMP) installation command, 47
single sign-on, Terminal Services, 287–290
[single] variables, 445
SiteName parameter, 44
sites, Active Directory, 103–104, 118
slmgr.vbs script, 33
Smartcard Logon certificate template, 409
snap-ins, Server Manager. See also Configuration snap-in, Server Manager
  Diagnostics, 62–67
  Features, 62
  overview, 60–61
  Roles, 61–62
  Storage, 90–94
SNMP (Simple Network Management Protocol) installation command, 47
soft limits, WSRM, 225
Software License Management Tool, Windows, 33
SoHR (Statement of Health Response), 259
SoHs (Statements of Health), 257
Sort Items option, WSRM Accounting, 235
specialize component pass, WDS, 167
Specify Columns option, WSRM Accounting, 235
SSL Certificate tab, TS Gateway server Properties, 308
SSTP (Secure Socket Tunneling Protocol), 384–385
stability index, Reliability Monitor, 246–247
Stand-alone Certification Authorities, 405–406
Standard management rule, WSRM, 225
Started state, Active Directory, 132
startup keys, BitLocker, 338
Statement of Health Response (SoHR), 259
Statements of Health (SoH), 257
Static Content component, IIS, 182
Static Content Compression component, IIS, 185
static routes, RRAS, 356, 377–381
Stopped state, Active Directory, 132
Storage snap-in, Server Manager, 90–94
Storage tab, Certificate Authority Properties, 420
stores, Boot Configuration Data, 10–13
Streaming Media Services role, Server Core, 41
stub zones, DNS, 332
suballocation of processor resources, WSRM, 226
subdirectory names, WSRM, 244
subject names, certificate templates, 406
Subordinate CA certificate template, 409
Subscription Properties dialog box, Server Manager, 66–67
subscriptions, event log, 66–67
Subsystem for UNIX-based applications
installation command, 47
summary page, WSRM, 218
summary, Task Scheduler, 68
SWITCH statements, 447
switches, command, 16–17
synchronization, ADDS, 140
SysKey parameter, 44
System Access Control List (SACL), 135
System Control Panel applet, 17
System Health Agents (SHAs), NAP, 258, 261–262
System Health Validators (SHVs), NAP, 259
System Image Manager (SIM)
attaching answer file to WS 2008 image, 172–174
overview, 164–166
unattended install files for WS 2008, 166–172
system partitions, 338
System Properties dialog box, Server Manager, 58–59
System reports, Reliability and Performance Monitor, 249
System Stability Chart, Reliability Monitor, 246
System State, 138
Systems Health Validator, Windows, 266
SysVolPath parameter, 45

T values, Windows PowerShell Dates, 456
tables, routing, 356
Tag for Announce Routes setting, RIPv2 Properties General tab, 368
Task Scheduler
   Actions tab, 72
   Conditions tab, 73
   creating tasks using, 73–77
   General tab, 70–71
   History tab, 73
   overview, 68–70
   Settings tab, 73
   Triggers tab, 71–72
TCP/IP communication, 354–356
Telnet Client installation command, 47
terminal servers, 313–316
Terminal Servers Properties dialog box, 326
Terminal Services
core functionality, 286–287
Gateway, 302–316
installing, 291–293
Licensing, 294–301
overview, 286
program placement and performance, 329–330
in Remote Administration mode, 36
remote programs, 317–323
single sign-on, 287–290
Web Access, 323–329
TFTP (Trivial File Transfer Protocol)
servers, 148
Time Before Route Is Removed (Seconds) property, RIPv2 Properties Advanced tab, 371
Time Before Routes Expire (Seconds) property, RIPv2 Properties Advanced tab, 371
time values, Windows PowerShell, 456
time, working with in PowerShell, 454–458
time zone applet, Server Core, 48
timeout, boot manager, 20
TLS (Transport Layer Security) 1.0, 305
tools display order, Boot Configuration Data, 20
ToUniversalTime method, 455
TPM. See Trusted Platform Model
tracing, IIS, 184, 205–211
traffic, replication, 104
Transport Layer Security (TLS) 1.0, 305
trees, domain, 99–100
triggered updates, 371
Triggers tab, Task Scheduler, 71–72
Trivial File Transfer Protocol (TFTP)
servers, 148
Trust List Signing User certificate
template, 409
Trusted Platform Model (TPM)
chips, 336
plus PIN, 337–338
plus startup key, 338
TPM only authentication, 337
TrustedInstall.exe command-line tool, 187
trusts, AD DS, 100
Tunnel Rules, Windows Firewall, 89
tunneling, Virtual Private Networks,
383–388
two-way transitive trusts, 100
typecasted variables, 444

\textbf{U}

\begin{itemize}
  \item u values, Windows PowerShell Dates, 456
  \item -uformat switch, 455
  \item uninstalling Windows BitLocker Drive Encryption, 351
  \item universal groups, 136
  \item Universal Time Code (UTC), 455
  \item unmanaged processes, 216
  \item Unrestricted execution policy,
    PowerShell, 442
  \item URL Authorization component, IIS, 184
  \item Use Broadcast or Multicast Only option,
    RIPv2 Properties Neighbors tab, 370
  \item Use Neighbors In Addition to Broadcast
    or Multicast option, RIPv2 Properties
    Neighbors tab, 370
  \item Use Neighbors Instead of Broadcast Or
    Multicast option, RIPv2 Properties
    Neighbors tab, 370
  \item user accounts, RODCs, 137
  \item User Defined reports, Reliability and
    Performance Monitor, 249
  \item UserData Properties screen, Windows
    Server, 171
  \item UserDomain parameter, 45
  \item UserName parameter, 45
  \item Users and Computers tabs, rule
    properties dialog boxes, 84
  \item users groups, Reliability and
    Performance Monitor, 240–241
  \item UTC (Universal Time Code), 455
  \item Utopia installation, 33–34
\end{itemize}

\textbf{V}

\begin{itemize}
  \item /v switch, 17
  \item V values, Windows PowerShell Dates, 456
  \item values, date, 455–456
  \item variables, PowerShell, 443–446, 452
  \item <verb> parameter, APPCMD.EXE, 195
  \item verb-noun naming convention, 439
  \item verbose parameter, 37
\end{itemize}
verbosity levels, IIS, 211
VeriSign, 404
view filters, WSRM, 233–234
View Network Connections link, Server Summary, 59
Virtual Private Networks (VPNs)
  configuring using PPTP, 394–398
  NAP Enforcement Client, 262
  NAP enforcement of, 257
  NAP Enforcement Server, 263
  remote access, 381, 383–388
volume master keys, 337–338
VPNs. See Virtual Private Networks

W

W values, Windows PowerShell
  Dates, 456
WAS (Windows Activation Service), 179
wadmin command, 122–123
WDS. See Windows Deployment Services
Web Access, Terminal Services, 323–329
Web Enrollment Agent, 425–426
Web Server certificate template, 409
Web sites folder, IIS Manager, 189
web.config files, 179, 199, 211
week values, Windows PowerShell, 456
Welcome to the Forgotten Password
  Wizard, 10
WHILE statements, 449
WIM (Windows Imaging Format), 146
Windows Activation Service (WAS), 179
Windows Authentication component, IIS, 184
Windows Automated Installation Kit
  (AIK), 161–162, 166
Windows BitLocker Drive Encryption
  architecture, 337–344
  initializing, 344–349
installation command, 47
overview, 332, 336
recovery, 350
requirements, 336
turning off or uninstalling, 351
Windows Boot Loader, 13
Windows Boot Manager, 12–14, 18, 20
Windows Calculator application, 320–321
Windows Deployment Services (WDS)
  components, 148
  creating operating system image
    for, 152–162
  installation, 148–151
  loading Install image using,
    162–164
  overview, 146–147
  properties, 151–152
  scenarios for, 147–148
  Windows SIM and unattended
    installs, 164–174
Windows Domain Name System.
  See Domain Name System
Windows Error Reporting, 60
Windows Firewall with Advanced
  Security
    Computer Connection Security,
      88–89
    configuration, 36–38
    creating new Inbound Rules, 85–88
    Firewall Monitoring, 89
    Inbound and Outbound Rules,
      83–85
    overview, 67, 77–83
Windows Image pane, SIM, 165
Windows Imaging Format (WIM), 146
Windows Installer (MSI), 47–48, 317
Windows Internet Name Service
  installation command, 47
Windows link-local multicast name resolution (LLMNR), 335
Windows Management Interface (WMI), 17, 22, 34, 89–90
Windows Memory Diagnostic tool, 13, 20
Windows memory tester, 14
Windows NT domain model, 96–97
Windows NT Loader (NTLDR), 10, 13
Windows PowerShell
  .NET managed code, 441–442
  basics of, 436–439
  cmdlets, 439–441
  conditional statements, 446–447
  loops, 448–451
  overview, 434–436
  real-world tasks, 451–458
  scripting and security, 442–443
  variables, 443–446
Windows registry, 452–454
Windows Remote Management (WRM), 64
Windows Remote Shell, 48
Windows resume application, 14
Windows Security Event Log, 134
Windows Security Health Validator dialog box, 267–271
Windows Server 2008
  Base Image, 172–173
  installation, 2–10
  reference system, 156–159
  WDS installation requirements, 149
Windows Server Backup, 91–94, 124, 138
Windows Service Hardening, 78, 80
Windows Software License Management Tool, 33
Windows System Image Manager (SIM)
  attaching answer file to WS 2008 image, 172–174
  overview, 164–166
  unattended install files for WS 2008, 166–172
Windows System Resource Manager (WSRM)
  Accounting, 231–235
  architecture, 215–216
  Calendar, 228–231
  Conditions, 235–236
  managed and unmanaged processes, 216
  management interface, 218–219
  overview, 213–215
  process matching criteria, 219–222
  resource allocation policies, 222–228
  Resource Monitor, 236–239
  service, 216–218
Windows Systems Health Validator, 266
windowsPE component pass, 167
winrs commands, 48
wizards. See individual wizards by name
WMI (Windows Management Interface), 17, 22, 34, 89–90
WMI Command-line (WMIC), 34
WMI Control Properties dialog box, Server Manager, 89
[wmi] variables, 445
WMIC (Windows Management Interface Command-line), 34
[wmiclass] variables, 445
worker processes, 205
working set memory, 226
Workstation Authentication certificate template, 409
WRM (Windows Remote Management), 64
WSRM. See Windows System Resource Manager
**X**

- x.509 certificates, 305
- XCOPY deployment, IIS, 211

**Y**

- year values, Windows PowerShell, 456

**Z**

- Z values, Windows PowerShell Dates, 456
- zero-touch deployment strategy, WDS, 147
- zone transfers, DNS, 333
- /zoneprint switch, 39
- zones
  - DNS, 38
  - NAP protected, 255
ESSENTIAL SKILLS for Network Professionals


VISIT MHPROFESSIONAL.COM TO READ SAMPLE CHAPTERS AND LEARN MORE.
Stop Hackers in Their Tracks
THE BEST

in Microsoft Certification Prep

VISIT MHPROFESSIONAL.COM TO READ SAMPLE CHAPTERS AND LEARN MORE.
CATCH THE LATEST WAVE
OF WEB 2.0 TECHNOLOGIES

available everywhere books are sold.