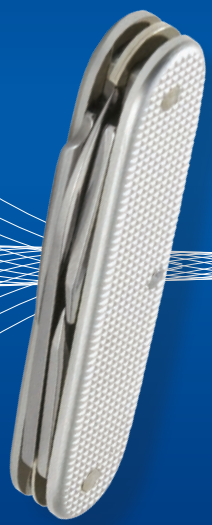
Microsoft

Windows Server 2012 R2 Essentials & Configuration

William R. Stanek
Author and Series Editor



Pocket Consultant

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

Library of Congress Control Number: 2013955479
ISBN: 978-0-7356-8257-3

Printed and bound in the United States of America.

First Printing

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at mspinput@microsoft.com. Please tell us what you think of this book at http://www.microsoft.com/learning/booksurvey.

Acquisitions Editor: Anne Hamilton
Developmental Editor: Karen Szall
Project Editor: Rosemary Caperton
Editorial Production: Online Training Solutions, Inc. (OTSI)
Technical Reviewer: Bob Hogan; Technical Review services provided by Content Master, a member of CM Group, Ltd.
Copyeditor: Victoria Thulman (OTSI)
Indexer: Krista Wall (OTSI)
Cover: Best & Company Design

Contents

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

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

microsoft.com/learning/booksurvey

Acknowledgments

To my readers—thank you for being there with me through many books and many years. It has been an honor and a privilege to be your pocket consultant.

To my wife—for many years, through many books, many millions of words, and many thousands of pages she's been there, providing support and encouragement and making every place we've lived a home.

To my kids—for helping me see the world in new ways, for having exceptional patience and boundless love, and for making every day an adventure.

To Anne, Karen, Martin, Lucinda, Juliana, and many others who've helped out in ways both large and small.

Special thanks to my son Will for not only installing and managing my extensive dev lab for all my books since *Windows 8 Pocket Consultant* but also for performing check reads of all those books.

—William R. Stanek

Introduction

Welcome to *Windows Server 2012 R2 Pocket Consultant: Essentials & Configuration*. Over the years, I've written about many different server technologies and products, but the one product I like writing about the most is Windows Server. Anyone transitioning to Windows Server 2012 R2 from Windows Server 2012 might be surprised at just how much has been updated, because changes both subtle and substantial have been made throughout the operating system. For anyone transitioning to Windows Server 2012 R2 from Windows Server 2008 R2 or an earlier release of Windows Server, I'll let you know right up front that I believe Windows Server 2012 and Windows Server 2012 R2 are the most significant updates to Windows Server since the introduction of Windows 2000 Server.

The good news is Windows Server 2012 R2 builds off the same code base as Windows 8.1. This means that you can apply much of what you know about Windows 8.1 to Windows Server 2012 R2, including how Windows works with touchscreen devices. Although you might not install Windows Server 2012 R2 on touchscreen computers, you can manage Windows Server 2012 R2 from your touchscreen computers. If you do end up managing it this way, understanding the touchscreen options and the revised interface options will be crucial to your success. For this reason, I discuss both the touchscreen options and the traditional mouse and keyboard techniques throughout this book.

When you are working with touchscreen computers, you can manipulate on-screen elements in ways that weren't possible previously. You can enter text by using the on-screen keyboard and also in the following ways:

- **Tap** Tap an item by touching it with your finger. A tap or double-tap of elements on the screen generally is the equivalent of a mouse click or double-click.
- **Press and hold** Press your finger down and leave it there for a few seconds. Pressing and holding elements on the screen generally is the equivalent of a right-click.
- **Swipe to select** Slide an item a short distance in the opposite direction compared to how the page scrolls. This selects the items and also might bring up related commands. If pressing and holding doesn't display commands and options for an item, try using swipe to select instead.
- **Swipe from edge (slide in from edge)** Starting from the edge of the screen, swipe or slide in. Sliding in from the right edge displays the charms. Sliding in from the left edge shows open apps and allows you to easily switch between them. Sliding in from the top or bottom edge shows commands for the active element.
- **Pinch** Touch an item by using two or more fingers, and then move the fingers toward each other. Pinching zooms out.
- **Stretch** Touch an item by using two or more fingers, and then move the fingers away from each other. Stretching zooms in.

Because I've written many top-selling Windows Server books, I was able to bring a unique perspective to this book—the kind of perspective you gain only after working with technologies for many years. As you've probably noticed, a great deal of information about Windows Server 2012 R2 is available on the web and in other printed books. You can find tutorials, reference sites, discussion groups, and more to make using Windows Server 2012 R2 easier. However, the advantage of reading this book is that much of the information you need to learn about Windows Server 2012 R2 is organized in one place and presented in a straightforward and orderly fashion. This book has everything you need to customize Windows Server 2012 R2 installations, master Windows Server 2012 R2 configurations, and maintain Windows Server 2012 R2 servers.

In this book, I teach you how features work, why they work the way they do, and how to customize them to meet your needs. I also offer specific examples of how certain features can meet your needs, and how you can use other features to troubleshoot and resolve issues you might have. In addition, this book provides tips, best practices, and examples of how to optimize Windows Server 2012 R2. This book won't just teach you how to configure Windows Server 2012 R2, it will teach you how to squeeze every last bit of power out of it and make the most from the features and options it includes.

Unlike many other books about administering Windows Server 2012 R2, this book doesn't focus on a specific user level. This isn't a lightweight beginner book. Regardless of whether you are a beginning administrator or a seasoned professional, many of the concepts in this book will be valuable to you, and you can apply them to your Windows Server 2012 R2 installations.

Who is this book for?

Windows Server 2012 R2 Pocket Consultant: Essentials & Configuration covers all editions of Windows Server 2012 R2. The book is designed for the following readers:

- Current Windows system administrators
- Accomplished users who have some administrator responsibilities
- Administrators upgrading to Windows Server 2012 R2 from previous versions
- Administrators transferring from other platforms

To pack in as much information as possible, I had to assume that you have basic networking skills and a basic understanding of Windows Server. With this in mind, I don't devote entire chapters to explaining Windows Server architecture, Windows Server startup and shutdown, or why you want to use Windows Server. I do, however, cover configuration, remote management, directory services, security, and much more.

I also assume that you are fairly familiar with Windows commands and procedures and the Windows user interface. If you need help learning Windows basics, you should read other resources (many of which are available from Microsoft Press).

How this book is organized

Rome wasn't built in a day, and this book wasn't intended to be read in a day, in a week, or even in a month. Ideally, you'll read this book at your own pace, a little each day as you work your way through all the features Windows Server 2012 R2 has to offer. This book is organized into 10 chapters. The chapters are arranged in a logical order, taking you from planning and deployment tasks to configuration and maintenance tasks.

Ease of reference is an essential part of this hands-on guide. This book has an expanded table of contents and an extensive index for finding answers to problems quickly. Many other quick-reference features have also been added to the book, including quick step-by-step procedures, lists, tables with fast facts, and extensive cross references.

As with all *Pocket Consultants*, *Windows Server 2012 R2 Pocket Consultant: Essentials & Configuration* is designed to be a concise and easy-to-use resource for managing Windows servers. This is the readable resource guide that you'll want on your desktop at all times. The book covers everything you need to perform the essential administrative tasks for Windows servers. Because the focus is on giving you maximum value in a pocket-size guide, you don't have to wade through hundreds of pages of extraneous information to find what you're looking for. Instead, you'll find exactly what you need to get the job done, and you'll find it quickly.

In short, the book is designed to be the one resource you turn to whenever you have questions regarding Windows Server administration. To this end, the book zeroes in on daily administration procedures, frequently performed tasks, documented examples, and options that are representative though not necessarily inclusive. One of my goals is to keep the content so concise that the book remains compact and easy to navigate while at the same time ensuring that it is packed with as much information as possible.

Conventions used in this book

I've used a variety of elements to help keep the text clear and easy to follow. You'll find code listings in `monospace` type. When I tell you to actually enter a command or text, the command or text appear in **bold** type. When I introduce and define a new term in a paragraph of text, I put it in *italics*.

NOTE **Group Policy now includes both policies and preferences. Under the Computer Configuration and User Configuration nodes, you find two nodes: Policies and Preferences. Settings for general policies are listed under the Policies node. Settings for general preferences are listed under the Preferences node. When referencing settings under the Policies node, I sometimes use shortcut references, such as User Configuration \Administrative Templates\Windows Components, or specify that the policies are found in the Administrative Templates for User Configuration under Windows Components. Both references tell you that the policy setting being discussed is under User Configuration rather than Computer Configuration and can be found under Administrative Templates\Windows Components.**

Other conventions include the following:

- **Best Practices** To examine the best technique to use when working with advanced configuration and maintenance concepts
- **Caution** To warn you about potential problems you should look out for
- **Important** To highlight important concepts and issues
- **More Info** To provide more information on a subject
- **Note** To provide additional details on a particular point that needs emphasis
- **Real World** To provide real-world advice when discussing advanced topics
- **Security Alert** To point out important security issues
- **Tip** To offer helpful hints or additional information

I truly hope you find that *Windows Server 2012 R2 Pocket Consultant: Essentials & Configuration* provides everything you need to perform the essential administrative tasks on Windows servers as quickly and efficiently as possible. You are welcome to send your thoughts to me at *williamstanek@aol.com*. Follow me on Twitter at WilliamStanek and on Facebook at *www.facebook.com/William.Stanek.Author*.

Other resources

No single magic bullet for learning everything you'll ever need to know about Windows Server 2012 R2 exists. Even though some books are offered as all-in-one guides, there's simply no way one book can do it all. With this in mind, I hope you use this book as it is intended to be used—as a concise and easy-to-use resource. It covers everything you need to perform core administration tasks for Windows servers, but it is by no means exhaustive.

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

I recommend that you regularly visit the Microsoft website for Windows Server (*microsoft.com/windowsserver/*) and *support.microsoft.com* to stay current with the latest changes. To help you get the most out of this book, you can visit my corresponding website at *williamstanek.com/windows*. This site contains information about Windows Server 2012 R2 and updates to the book.

Errata and book support

We've made every effort to ensure the accuracy of this book and its companion content. Any errors that have been reported since this book was published are listed at:

http://aka.ms/wsR2pc_EC/errata

If you find an error that is not already listed, you can report it to us through the same page.

If you need additional support, email Microsoft Press Book Support at:

mspinput@microsoft.com

Please note that product support for Microsoft software is not offered through the addresses above.

We want to hear from you

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

http://aka.ms/tellpress

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in touch

Let's keep the conversation going! We're on Twitter: *http://twitter.com/MicrosoftPress*.

CHAPTER 1

Windows Server 2012 R2 administration

Windows Server 2012 R2 is a powerful, versatile, full-featured server operating system that builds on the enhancements that Microsoft provided in Windows Server 2012. Windows Server 2012 R2 and Windows 8.1 share a number of common features because they were part of a single development project. These features share a common code base and extend across many areas of the operating systems, including management, security, networking, and storage. Because of this, you can apply much of what you know about Windows 8.1 to Windows Server 2012 R2.

This chapter covers getting started with Windows Server 2012 R2 and explores the extent to which the architectural changes affect how you work with and manage Windows Server 2012 R2. Throughout this chapter and the other chapters of this book, you'll also find discussions of the many security features and enhancements. These discussions explore all aspects of computer security, including physical security, information security, and network security. Although this book focuses on Windows Server 2012 R2 administration, the tips and techniques it presents can help anyone who supports, develops for, or works with the Windows Server 2012 R2 operating system.

Windows Server 2012 R2 and Windows 8.1

Before you deploy Windows Server 2012 R2, you should carefully plan the server architecture. As part of your implementation planning, you need to look closely at the software configuration that will be used and modify the hardware configuration on a per-server basis to meet related requirements. For additional flexibility in server deployments, you can deploy servers by using one of three installation types:

- **Server With A GUI installation** An installation option that provides full functionality—also referred to as a *full-server installation*. You can configure a server to have any allowed combination of roles, role services, and features, and a full user interface is provided for managing the server. This installation option provides the most dynamic solution and is recommended for deployments of Windows Server 2012 R2 in which the server role might change over time.
- **Server Core installation** A minimal installation option that provides a fixed subset of roles but does not include the Server Graphical Shell, Microsoft Management Console, or Desktop Experience. You can configure a Server Core installation with a limited set of roles. A limited user interface is provided for managing the server, and most management is done locally at a command prompt or remotely by using management tools. This installation option is ideally suited to situations in which you want to dedicate servers to a specific server role or combination of roles. Because additional functionality is not installed, the overhead caused by other services is reduced, providing more resources for the dedicated role or roles. Generally, the limited interface also is inherently more secure than other installation types.
- **Server With Minimal Interface installation** An intermediate installation option where you perform a full-server installation and then remove the Server Graphical Shell. This leaves a minimal user interface, Microsoft Management Console, Server Manager, and a subset of Control Panel for local management. This installation option is ideally suited to situations in which you want to carefully control the tasks that can be performed on a server and the roles and features installed, but you still want the convenience of the graphical interface.

You choose the installation type during installation of the operating system. In a significant change from earlier releases of Windows Server, you can change the installation type after you've installed a server. A key difference between the installation types relates to the presence of the graphical management tools and the graphical shell. A Server Core installation has neither, a full-server installation has both, and a minimal-interface installation has only the graphical management tools.

MORE INFO **Several server features and roles require the graphical shell. They include Fax Server, Remote Desktop Session Host, Windows Deployment Services, and the Internet Printing user interface. Additionally, in Event Viewer, the Details view requires the graphical shell, as does the graphical interface for Windows Firewall.**

Like Windows 8.1, Windows Server 2012 R2 has the following features:

- **Modularization for language independence and disk imaging for hardware independence** Each component of the operating system is designed as an independent module you can easily add or remove. This functionality provides the basis for the configuration architecture in Windows Server 2012 R2. Microsoft distributes Windows Server 2012 R2 on media with Windows Imaging (WIM) format disk images that use compression and single-instance storage to dramatically reduce the size of image files.
- **Preinstallation and preboot environments** The Windows Preinstallation Environment (Windows PE) replaces MS-DOS as the preinstallation environment and provides a bootable startup environment for installation, deployment, recovery, and troubleshooting. The Windows Preboot Environment provides a startup environment with a boot manager that lets you choose which boot application to run to load the operating system. On systems with multiple operating systems, you access pre–Windows 7 operating systems in the boot environment by using the legacy operating system entry.
- **User account controls and elevation of privileges** User Account Control (UAC) enhances computer security by ensuring true separation of standard user and administrator user accounts. Through UAC, all applications are run by using either standard user or administrator user privileges, and a security prompt is displayed by default whenever you run an application that requires administrator privileges. The way the security prompt works depends on Group Policy settings. Additionally, if you log on by using the built-in Administrator account, typically elevation prompts are not provided.

In Windows 8.1 and Windows Server 2012 R2, features with common code bases have identical management interfaces. In fact, just about every Control Panel utility that is available in Windows Server 2012 R2 is identical to or nearly identical to its Windows 8.1 counterpart. Of course, exceptions exist in some cases for standard default settings. Because Windows Server 2012 R2 does not use performance ratings, Windows servers do not have Windows Experience Index scores. Because Windows Server 2012 R2 does not use Sleep or related states, Windows servers do not have sleep, hibernate, or resume functionality. Because you typically do not want to use extended power management options on Windows servers, Windows Server 2012 R2 has a limited set of power options.

Windows Server 2012 R2 does not include the Windows Aero enhancements, Windows Sidebar, or other user-interface enhancements, because Windows Server 2012 R2 is designed to provide optimal performance for server-related tasks and is not designed for extensive personalization of the desktop appearance. That said, when you are working with a full-server installation, you can add the Desktop Experience feature and then enable some Windows 8.1 features on your server.

The Desktop Experience provides Windows desktop functionality on the server. Windows features added include Windows Media Player, desktop themes, Video

for Windows (AVI support), Windows Defender, Disk Cleanup, Sync Center, Sound Recorder, Character Map, and Snipping Tool. Although these features allow a server to be used like a desktop computer, they can reduce the server's overall performance.

> ***NOTE*** **Windows Defender for Windows Server 2012 R2 has been upgraded to a more fully featured program. Windows Defender now protects against viruses, spyware, rootkits, and other types of malware. Rootkit detection helps to safeguard computers from malware that inserts itself into non-Microsoft drivers. If Windows Defender detects that a non-Microsoft driver has been infected, it prevents the driver from starting. Microsoft drivers are protected at startup as part of other security features. Note also that Windows Defender is available on Server Core installations, though without the user interface. If you add Windows Defender as an option on a Server Core installation, the program is enabled by default.**

Because the common features of Windows 8.1 and Windows Server 2012 R2 have so many similarities, I will not cover changes in the interface from previous operating system releases, discuss how UAC works, and so on. You can find extensive coverage of these features in *Windows 8.1 Pocket Consultant: Essentials & Configuration* (Microsoft Press, 2013), which I encourage you to use in conjunction with this book. In addition to its coverage of broad administration tasks, *Windows 8.1 Pocket Consultant: Essentials & Configuration* examines how to customize the operating system and Windows environment, configure hardware devices, manage user access and global settings, troubleshoot system problems, and much more. This book, alternatively, focuses on directory services administration, user administration, and server management.

Getting to know Windows Server 2012 R2

The Windows Server 2012 R2 operating system includes several different editions. All Windows Server 2012 R2 editions support multiple processor cores. It is important to point out that although an edition might support only one discrete-socketed processor (also referred to as a *physical processor*), that one processor could have eight processor cores (also referred to as *logical processors*).

Introducing Windows Server 2012 R2

Windows Server 2012 R2 is a 64-bit–only operating system. In this book, I refer to 64-bit systems designed for the x64 architecture as *64-bit* systems. Because the various server editions support the same core features and administration tools, you can use the techniques discussed in this book regardless of which Windows Server 2012 R2 edition you're using.

When you install a Windows Server 2012 R2 system, you configure the system according to its role on the network, as the following guidelines describe:

- Servers are generally assigned to be part of a workgroup or a domain.
- *Workgroups* are loose associations of computers in which each individual computer is managed separately.

- *Domains* are collections of computers you can manage collectively by means of *domain controllers*, which are Windows Server 2012 R2 systems that manage access to the network, to the directory database, and to shared resources.

NOTE **In this book, *Windows Server 2012 R2* and *Windows Server 2012 R2 family* refer to all editions of Windows Server 2012 R2. The various server editions support the same core features and administration tools.**

Windows 8.1 and Windows Server 2012 R2 also support a workplace configuration. A *workplace* is a loose association of computers that grants access to certain internal network resources and business apps. Workplaces have specific benefits:

- If users have Windows 8.1 devices from which they want to access corporate resources, those devices can use a workplace configuration to remotely connect to a computer at work. Users can then change their network passwords and connect to internal websites. Users also can use workplaces as an alternative way to access Outlook Web Access (OWA). Here, users connect to the workplace and then access OWA by using the internal URL (rather than an external URL).
- If administrators have Windows 8.1 devices from which they want to access corporate resources, those devices can use a workplace configuration to remotely connect to servers and perform assigned tasks by using administration links.

You implement workplaces by installing the Windows Server Essentials Experience role on servers running Windows Server 2012 R2. Because the Windows Server Essentials Experience role is designed to be used in single-domain deployments of Active Directory Domain Services (AD DS), you should not deploy this role in AD DS implementations with multiple domains.

Windows 8.1 and Windows Server 2012 R2 also support Work Folders, a similar but distinctly different feature. Work Folders allow users to synchronize their corporate data to their devices and vice versa. Those devices can be joined to the corporate domain or a workplace. To deploy Work Folders, the administrator adds the File And Storage Services\File and iSCSI Services\Work Folders role service to a server and then configures Work Folders by using Server Manager. Unlike workplaces, Work Folders can be used in multidomain environments.

Working with Windows Server 2012 R2

Windows Server 2012 R2 uses a Start screen. Start is a window, not a menu. Programs can have Tiles on the Start screen. Tapping or clicking a Tile runs the program. When you press and hold or right-click on a program, an options panel usually is displayed. The charms bar is an options panel for Start, Desktop, and PC Settings. With a touch UI, you can display the charms by sliding in from the right side of the screen. With a mouse and keyboard, you can display the charms by pointing to the hidden button in the upper-right or lower-right corner of the Start, Desktop, or PC Settings screen; or by pressing Windows key+C.

Tap or click the Search charm to display the Search panel. Any text entered while on the Start screen is entered into the Search box in the Search panel. The search box can be focused on Everywhere, Settings, or Files. When focused on Everywhere, you can use Search to quickly find installed programs, settings, and files. When focused on Settings, you can use Search to quickly find settings and options in Control Panel. When focused on Files, you can use Search to quickly find files.

One way to quickly open a program is by typing the file name of the program, and then pressing Enter. This shortcut works as long as the Everywhere search box is in focus (which it typically is by default). In an Everywhere search, any matches for programs are listed first, followed by any matches for settings and finally, any matches for files.

Pressing the Windows key toggles between the Start screen and the desktop (or, if you are working with PC Settings, between Start and PC Settings). On Start, there's a Desktop Tile that you can tap or click to display the desktop. You also can display the desktop by pressing Windows key+D or, to peek at the desktop, press and hold Windows key+Comma. From Start, you access Control Panel by tapping or clicking the Control Panel Tile. From either Start or the desktop, you can display Control Panel by pressing Windows key+I and then clicking the Control Panel option. Additionally, because File Explorer is pinned to the desktop taskbar by default, you typically can access Control Panel on the desktop by following these steps:

1. Open File Explorer by tapping or clicking the taskbar icon.
2. Tap or click the leftmost arrow button in the address list.
3. Tap or click Control Panel.

Start and Desktop have a convenient menu that you can display by right-clicking the lower-left corner of the Start screen or the desktop. Alternatively, you can press Windows key+X. Options on the menu include Computer Management, Device Manager, Event Viewer, System, Task Manager, Windows Command Prompt, and Command Prompt (Admin). On the Start screen, the button in the lower-left corner shows a Windows icon, and tapping or clicking the thumbnail opens the desktop. On the desktop, tapping or clicking this button opens Start. Right-clicking the button is what displays the shortcut menu.

REAL WORLD **By default with Windows Server 2012 R2, the command prompt and the administrator command prompt are options on the shortcut menu that is displayed when you right-click in the lower-left corner or press Windows key+X. The alternative is for the Windows PowerShell command prompt and the administrator Windows PowerShell command prompt to be displayed on this menu. To configure which options are available, on the desktop, press and hold or right-click the taskbar, and then click Properties. In the Taskbar And Navigation Properties dialog box, on the Navigation tab, select or clear the Replace Command Prompt With Windows PowerShell... check box as appropriate.**

Shut Down and Restart are options of Power settings now. This means to shut down or restart a server, you follow these steps:

1. Display Start options by sliding in from the right side of the screen or moving the mouse pointer to the lower-right or upper-right corner of the screen.
2. Tap or click Settings, and then tap or click Power.
3. Tap or click Shut Down or Restart as appropriate.

Alternatively, if configured as a power option, you can press the server's physical power button to initiate an orderly shutdown by logging off and then shutting down. If you are using a desktop-class system and the computer has a sleep button, the sleep button is disabled by default, as are closing the lid options for portable computers. Additionally, servers are configured to turn off the display after 10 minutes of inactivity.

Windows 8.1 and Windows Server 2012 R2 support the Advanced Configuration and Power Interface (ACPI) 5.0 specification. Windows uses ACPI to control system and device power state transitions, putting devices in and out of full-power (working), low-power, and off states to reduce power consumption.

The power settings for a computer come from the active power plan. You can access power plans in Control Panel by tapping or clicking System And Security and then tapping or clicking Power Options. Windows Server 2012 R2 includes the Power Configuration (Powercfg.exe) utility for managing power options from the command prompt. At a command prompt, you can view the configured power plans by entering **powercfg /l**. The active power plan is marked with an asterisk.

The default, active power plan in Windows Server 2012 R2 is called *Balanced*. The Balanced plan is configured to do the following:

- Never turn off hard disks (as opposed to turning off hard disks after a specified amount of idle time).
- Disable timed events to wake the computer (as opposed to enabling wake on timed events).
- Enable USB selective suspend (as opposed to disabling selective suspend).
- Use moderate power savings for idle PCI Express links (as opposed to maximum power savings being on or off).
- Use active system cooling by increasing the fan speed before slowing processors (as opposed to using passive system cooling to slow the processors before increasing fan speed).
- Use minimum processor and maximum processor states if supported (as opposed to using a fixed state).

NOTE **Power consumption is an important issue, especially as organizations try to become more earth friendly. Saving power also can save your organization money and, in some cases, allow you to install more servers in your data centers. If you install Windows Server 2012 R2 on a laptop—for testing or for your personal computer, for example—your power settings will be slightly different, and you'll also have settings for when the laptop is running on battery.**

Power management options

When working with power management, important characteristics to focus on include the following:

- Cooling modes
- Device states
- Processor states

ACPI defines active and passive cooling modes. These cooling modes are inversely related to each other:

- *Passive cooling* reduces system performance but is quieter because there's less fan noise. With passive cooling, Windows lessens power consumption to reduce the operating temperature of the computer but at the cost of system performance. Here, Windows reduces the processor speed in an attempt to cool the computer before increasing fan speed, which would increase power consumption.
- *Active cooling* allows maximum system performance. With active cooling, Windows increases power consumption to reduce the temperature of the machine. Here, Windows increases fan speed to cool the computer before attempting to reduce processor speed.

Power policy includes an upper and lower limit for the processor state, referred to as the *maximum processor state* and the *minimum processor state*, respectively. These states are implemented by making use of a feature of Advanced Configuration and Power Interface (ACPI) 3.0 and later versions called *processor throttling*, and they determine the range of currently available processor performance states that Windows can use. By setting the maximum and minimum values, you define the bounds for the allowed performance states, or you can use the same value for each to force the system to remain in a specific performance state. Windows reduces power consumption by throttling the processor speed. For example, if the upper bound is 100 percent and the lower bound is 5 percent, Windows can throttle the processor within this range as workloads permit to reduce power consumption. In a computer with a 3-gigahertz (GHz) processor, Windows would adjust the operating frequency of the processor between .15 GHz and 3.0 GHz.

Processor throttling and related performance states were introduced with Windows XP and are not new, but these early implementations were designed for computers with discrete-socketed processors and not for computers with processor cores. As a result, they are not effective in reducing the power consumption of computers with logical processors. Windows 7 and later releases of Windows reduce power consumption in computers with multicore processors by taking advantage of a feature of ACPI 4.0 called *logical processor idling* and by updating processor throttling features to work with processor cores.

Logical processor idling is designed to ensure that Windows uses the fewest number of processor cores for a given workload. Windows accomplishes this by consolidating workloads onto the fewest cores possible and suspending inactive

processor cores. As additional processing power is required, Windows activates inactive processor cores. This idling functionality works in conjunction with management of process performance states at the core level.

ACPI defines processor performance states, referred to as *p-states*, and processor idle sleep states, referred to as *c-states*. Processor performance states include P0 (the processor/core uses its maximum performance capability and can consume maximum power), P1 (the processor/core is limited below its maximum and consumes less than maximum power), and P*n* (where state *n* is a maximum number that is processor-dependent, and the processor/core is at its minimal level and consumes minimal power while remaining in an active state).

Processor idle sleep states include C0 (the processor/core can execute instructions), C1 (the processor/core has the lowest latency and is in a nonexecuting power state), C2 (the processor/core has longer latency to improve power savings over the C1 state), and C3 (the processor/core has the longest latency to improve power savings over the C1 and C2 states).

> ***MORE INFO*** **ACPI 4.0 was finalized in June 2009, and ACPI 5.0 was finalized in December 2011. Computers manufactured prior to this time will likely not have firmware that is fully compliant, and you will probably need to update the firmware when a compatible revision becomes available. In some cases, and especially with older hardware, you might not be able to update a computer's firmware to make it fully compliant with ACPI 4.0 or ACPI 5.0. For example, if you are configuring the power options and you don't have minimum and maximum processor state options, the computer's firmware isn't fully compatible with ACPI 3.0 and likely will not fully support ACPI 4.0 or ACPI 5.0 either. Still, you should check the hardware manufacturer's website for firmware updates.**
>
> **Windows switches processors/cores between any p-state and from the C1 state to the C0 state nearly instantaneously (fractions of milliseconds) and tends not to use the deep sleep states, so you don't need to worry about performance impact to throttle or wake up processors/cores. The processors/cores are available when they are needed. That said, the easiest way to limit processor power management is to modify the active power plan and set the minimum and maximum processor states to 100 percent.**
>
> **Logical processor idling is used to reduce power consumption by removing a logical processor from the operating system's list of nonprocessor-affinitized work. However, because processor-affinitized work reduces the effectiveness of this feature, you'll want to plan carefully prior to configuring processing affinity settings for applications. Windows System Resource Manager allows you to manage processor resources through percent processor usage targets and processor affinity rules. Both techniques reduce the effectiveness of logical processor idling.**

Windows saves power by putting processor cores in and out of appropriate p-states and c-states. On a computer with four logical processors, Windows might use p-states 0 through 5, where P0 allows 100 percent usage, P1 allows 90 percent usage, P2 allows 80 percent usage, P3 allows 70 percent usage, P4 allows 60 percent usage, and P5 allows 50 percent usage. When the computer is active, logical processor 0 would likely be active with a p-state of 0 through 5, and the other processors would

likely be at an appropriate p-state or in a sleep state. Figure 1-1 shows an example. Here, logical processor 1 is running at 90 percent, logical processor 2 is running at 80 percent, logical processor 3 is running at 50 percent, and logical processor 4 is in the sleep state.

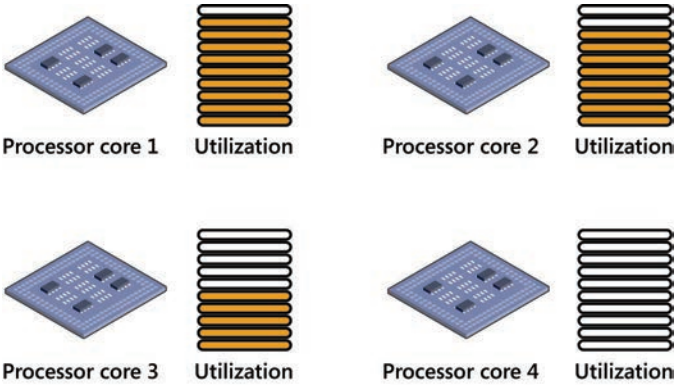


FIGURE 1-1 Processor cores working at different p-states have different usage patterns.

REAL WORLD **ACPI 4.0 and ACPI 5.0 define four global power states. In G0, the working state in which software runs, power consumption is at its highest and latency is at its lowest. In G1, the sleeping state, in which software doesn't run, latency varies with sleep state and power consumption is less than the G0 state. In G2 (also referred to as S5 sleep state), the soft off state where the operating system doesn't run, latency is long and power consumption is very near zero. In G3, the mechanical off state, where the operating system doesn't run, latency is long, and power consumption is zero. There's also a special global state, known as S4 nonvolatile sleep, in which the operating system writes all system context to a file on nonvolatile storage media, allowing system context to be saved and restored.**

Within the global sleeping state, G1, are sleep-state variations. S1 is a sleeping state where all system context is maintained. S2 is a sleeping state similar to S1 except that the CPU and system-cache contexts are lost and control starts from a reset. S3 is a sleeping state where all CPU, cache, and chip-set context are lost and hardware maintains memory context and restores some CPU and L2 cache configuration context. S4 is a sleeping state in which it is assumed that the hardware has powered off all devices to reduce power usage to a minimum and only the platform context is maintained. S5 is a sleeping state in which it is assumed that the hardware is in a soft off state, where no context is maintained and a complete boot is required when the system wakes.

Devices also have power states. D0, the fully on state, consumes the highest level of power. D1 and D2 are intermediate states that many devices do not use. D3hot is a power-saving state, where the device is software enumerable and can optionally preserve device context. D3 is the off state, where the device context is lost and the operating system must reinitialize the device to turn it back on.

Networking tools and protocols

Windows Server 2012 R2 has a suite of networking tools that includes Network Explorer, Network And Sharing Center, and Network Diagnostics. Figure 1-2 shows Network And Sharing Center.

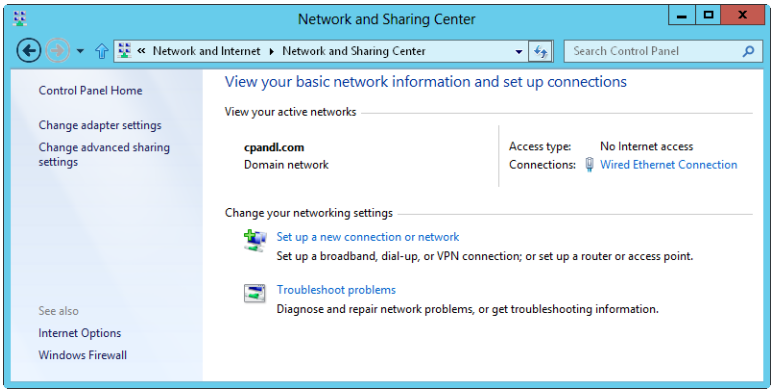


FIGURE 1-2 Network And Sharing Center provides quick access to sharing, discovery, and networking options.

Understanding networking options

The sharing and discovery configuration in Network And Sharing Center controls basic network settings. When network discovery settings are turned on and a server is connected to a network, the server can see other network computers and devices and is visible on the network. When sharing settings are turned on or off, the various sharing options are allowed or restricted.

In Windows 8.1 and Windows Server 2012 R2, networks are identified as one of the following network types:

- **Domain** A network in which computers are connected to the corporate domain to which they are joined
- **Work** A private network in which computers are configured as members of a workgroup and are not connected directly to the public Internet
- **Workplace** A private network in which computers are configured as members of a workplace to which devices can connect over the public Internet (Windows 8.1 only)
- **Home** A private network in which computers are configured as members of a homegroup and are not connected directly to the public Internet
- **Public** A public network in which computers are connected to a network in a public place, such as a coffee shop or an airport, rather than an internal network

These network types are organized into three categories: private, domain, and public. Each network category has an associated network profile. Because a computer saves sharing and firewall settings separately for each network category, you can use different block and allow settings for each network category. When you connect to a network, a dialog box is displayed in which you can specify the network category. If you select Private, and the computer determines that it is connected to the corporate domain to which it is joined, the network category is set as Domain Network.

Based on the network category, Windows Server configures settings that turn discovery on or off. The On (enabled) state means that the computer can discover other computers and devices on the network and that other computers on the network can discover the computer. The Off (disabled) state means that the computer cannot discover other computers and devices on the network and that other computers on the network cannot discover the computer.

Using Advanced Sharing Settings in Network And Sharing Center, you can enable discovery and file sharing. However, discovery and file sharing are blocked by default on a public network, which enhances security by preventing computers on the public network from discovering other computers and devices on that network. When discovery and file sharing are disabled, files and printers you have shared from a computer cannot be accessed from the network. Additionally, some programs might not be able to access the network.

Working with networking protocols

To allow a server to access a network, you must install TCP/IP networking and a network adapter. Windows Server uses TCP/IP as the default wide area network (WAN) protocol. Normally, networking is installed during installation of the operating system. You can also install TCP/IP networking through local area connection properties.

The TCP and IP protocols make it possible for computers to communicate across various networks and the Internet by using network adapters. Windows 7 and later releases of Windows have a dual IP-layer architecture in which both Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) are implemented and share common transport and network layers. IPv4 has 32-bit addresses and is the primary version of IP used on most networks, including the Internet. IPv6, alternatively, has 128-bit addresses and is the next-generation version of IP.

NOTE **DirectAccess clients send only IPv6 traffic across the DirectAccess connection to the DirectAccess server. Thanks to the NAT64/DNS64 support, DirectAccess clients can initiate communications with IPv4-only hosts on the corporate intranet. NAT64/DNS64 work together to translate incoming connection traffic from an IPv6 node to IPv4 traffic. The NAT64 translates the incoming IPv6 traffic to IPv4 traffic and performs the reverse translation for response traffic. The DNS64 resolves the name of an IPv4-only host to a translated IPv6 address.**

REAL WORLD **The TCP Chimney Offload enables the networking subsystem to offload the processing of a TCP/IP connection from the computer's processors to its network adapter as long as the network adapter supports TCP/IP offload processing. Both TCP/IPv4 connections and TCP/IPv6 connections can be offloaded. For Windows 7 and later releases of Windows, TCP connections are offloaded by default on 10 gigabits per second (Gbps) network adapters, but they are not offloaded by default on 1-Gbps network adapters. To offload TCP connections on a 1-Gbps or 10-Gbps network adapter, you must enable TCP offloading by entering the following command at an elevated, administrator command prompt: netsh int tcp set global chimney=enabled. You can check the status of TCP offloading by entering netsh int tcp show global. Although TCP offloading works with Windows Firewall, TCP offloading won't be used with IPsec, Windows virtualization (Hyper-V), network load balancing, or the Network Address Translation (NAT) service. To determine whether TCP offloading is working, enter netstat-t and check the offload state. The offload state is listed as *offloaded* or *inhost*.**

Windows also uses receive-side scaling and network direct memory access (NetDMA). You can enable or disable receive-side scaling by entering netsh int tcp set global rss=enabled or netsh int tcp set global rss=disabled, respectively. To check the status of receive-side scaling, enter netsh int tcp show global. You can enable or disable NetDMA by setting a DWord value under the EnableTCPA registry entry to 1 or 0, respectively. This registry entry is found under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters.

IPv4's 32-bit addresses are commonly expressed as four separate decimal values, such as 127.0.0.1 or 192.168.10.52. The four decimal values are referred to as *octets* because each represents 8 bits of the 32-bit number. With standard unicast IPv4 addresses, a variable part of the IP address represents the network ID, and a variable part of the IP address represents the host ID. A host's IPv4 address and its media access control (MAC) address used by the host's network adapter have no correlation.

IPv6's 128-bit addresses are divided into eight 16-bit blocks delimited by colons. Each 16-bit block is expressed in hexadecimal form, such as FEC0:0:0:02BC:FF:BECB:FE4F:961D. With standard unicast IPv6 addresses, the first 64 bits represent the network ID, and the last 64 bits represent the network interface. Because many IPv6 address blocks are set to 0, a contiguous set of 0 blocks can be expressed as "::", a notation referred to as *double-colon notation*. Using double-colon notation, the two 0 blocks in the previous address can be compressed as FEC0::02BC:FF:BECB:FE4F:961D. Three or more 0 blocks would be compressed in the same way. For example, FFE8:0:0:0:0:0:0:1 becomes FFE8::1.

When networking hardware is detected during installation of the operating system, both IPv4 and IPv6 are enabled by default; you don't need to install a separate component to enable support for IPv6. The modified IP architecture in Windows 7 and later releases of Windows is referred to as the *Next Generation TCP/IP stack*, and it includes many enhancements that improve the way IPv4 and IPv6 are used.

For Windows 8.1 and Windows Server 2012 R2, Group Policy Preferences have been updated to support IPv6 addresses for TCP/I Printers and VPN connections. Item-level Targeting options also allow you to configure IPv6 address ranges.

For traffic routing between virtual and physical networks, Windows Server 2012 R2 includes Windows Server Gateway, which is integrated with Hyper-V Network Virtualization. You can use Windows Server Gateway to route network traffic regardless of where resources are located, allowing you to support integration of public and private cloud services with your internal networks in addition to multitenant implementations with Network Address Translation (NAT) and virtual private network (VPN) connections.

Domain controllers, member servers, and domain services

When you install Windows Server 2012 R2 on a new system, you can configure the server to be a member server, a domain controller, or a stand-alone server. The differences between these types of servers are extremely important. Member servers are part of a domain but don't store directory information. Domain controllers are distinguished from member servers because they store directory information and provide authentication and directory services for the domain. Stand-alone servers aren't part of a domain. Because stand-alone servers have their own user databases, they authenticate logon requests independently.

Working with Active Directory

Windows Server 2012 R2 supports a multimaster replication model. In this model, any domain controller can process directory changes and then replicate those changes to other domain controllers automatically. Windows Server distributes an entire directory of information, called a *data store*. Inside the data store are sets of objects representing user, group, and computer accounts in addition to shared resources such as servers, files, and printers.

Domains that use Active Directory are referred to as *Active Directory domains*. Although Active Directory domains can function with only one domain controller, you can and should configure multiple domain controllers in the domain. This way, if one domain controller fails, you can rely on the other domain controllers to handle authentication and other critical tasks.

Microsoft changed Active Directory in several fundamental ways for the original release of Windows Server 2008. As a result, Microsoft realigned the directory functionality and created a family of related services, including the following:

- **Active Directory Certificate Services (AD CS)** AD CS provides functions necessary for issuing and revoking digital certificates for users, client computers, and servers. AD CS uses certificate authorities (CAs), which are responsible for confirming the identity of users and computers and then issuing certificates to confirm these identities. Domains have enterprise root CAs,

which are the certificate servers at the root of certificate hierarchies for domains and the most trusted certificate servers in the enterprise, and subordinate CAs, which are members of a particular enterprise certificate hierarchy. Workgroups have stand-alone root CAs, which are the certificate servers at the root of nonenterprise certificate hierarchies, and stand-alone subordinate CAs, which are members of a particular nonenterprise certificate hierarchy.

- **Active Directory Domain Services (AD DS)** AD DS provides the essential directory services necessary for establishing a domain, including the data store that stores information about objects on the network and makes that information available to users. AD DS uses domain controllers to manage access to network resources. After users authenticate themselves by logging on to a domain, their stored credentials can be used to access resources on the network. Because AD DS is the heart of Active Directory and is required for directory-enabled applications and technologies, I typically refer to it simply as *Active Directory* rather than Active Directory Domain Services or AD DS.
- **Active Directory Federation Services (AD FS)** AD FS complements the authentication and access-management features of AD DS by extending them to the World Wide Web. AD FS uses web agents to provide users with access to internally hosted web applications and proxies to manage client access. After AD FS is configured, users can use their digital identities to authenticate themselves over the web and access internally hosted web applications with a web browser such as Internet Explorer.
- **Active Directory Lightweight Directory Services (AD LDS)** AD LDS provides a data store for directory-enabled applications that do not require AD DS and do not need to be deployed on domain controllers. AD LDS does not run as an operating system service and can be used in both domain and workgroup environments. Each application that runs on a server can have its own data store implemented through AD LDS.
- **Active Directory Rights Management Services (AD RMS)** AD RMS provides a layer of protection for an organization's information that can extend beyond the enterprise, allowing email messages, documents, intranet webpages, and more to be protected from unauthorized access. AD RMS uses a certificate service to issue rights account certificates that identify trusted users, groups, and services; a licensing service that provides authorized users, groups, and services with access to protected information; and a logging service to monitor and maintain the rights management service. After trust is established, users with a rights account certificate can assign rights to information. These rights control which users can access the information and what they can do with it. Users with rights account certificates can also access protected content to which they've been granted access. Encryption ensures that access to protected information is controlled both inside and outside the enterprise.

Microsoft introduced additional changes in Windows Server 2012 R2. These changes include a new domain functional level, called *Windows Server 2012 R2 domain functional level*, and a new forest functional level, called *Windows Server 2012 R2 forest functional level*. The many other changes are discussed in Chapter 7, "Using Active Directory."

Using read-only domain controllers

Windows Server 2008 and later releases support read-only domain controllers (RODCs) and restartable AD DS. An RODC is an additional domain controller that hosts a read-only replica of a domain's Active Directory data store. RODCs are ideally suited to the needs of branch offices, where a domain controller's physical security cannot be guaranteed. Except for passwords, RODCs store the same objects and attributes as writable domain controllers. These objects and attributes are replicated to RODCs through unidirectional replication from a writable domain controller that acts as a replication partner.

Because RODCs by default do not store passwords or credentials other than for their own computer account and the Kerberos Target (Krbtgt) account, RODCs pull user and computer credentials from a writable domain controller that is running Windows Server 2008 or later. If allowed by a password replication policy that is enforced on the writable domain controller, an RODC retrieves and then caches credentials as necessary until the credentials change. Because only a subset of credentials is stored on an RODC, this limits the number of credentials that can possibly be compromised.

IMPORTANT **Any domain user can be delegated as a local administrator of an RODC without granting any other rights in the domain. An RODC can act in the role of a global catalog but cannot act in the role of an operations master. Although RODCs can pull information from domain controllers running Windows Server 2003, RODCs can pull updates of the domain partition only from a writable domain controller running Windows Server 2008 or later in the same domain.**

Using restartable Active Directory Domain Services

Restartable AD DS is a feature that allows an administrator to start and stop AD DS. In the Services console, the Active Directory Domain Services service is available on domain controllers, allowing you to easily stop and restart AD DS in the same way as for any other service that is running locally on the server. While AD DS is stopped, you can perform maintenance tasks that would otherwise require restarting the server, such as performing offline defragmentation of the Active Directory database, applying updates to the operating system, or initiating an authoritative restore. While AD DS is stopped on a server, other domain controllers can handle authentication and logon tasks. Cached credentials, smart cards, and biometric logon methods continue to be supported. If no other domain controller is available and none of these logon methods applies, you can still log on to the server by using the Directory Services Restore Mode account and password.

All domain controllers running Windows Server 2008 or later support restartable AD DS—even RODCs. As an administrator, you can start or stop AD DS by using the Domain Controller entry in the Services utility. Because of restartable AD DS, domain controllers running Windows Server 2008 or later have three possible states:

- **Active Directory Started** Active Directory is started, and the domain controller has the same running state as a domain controller running Windows 2000 Server or Windows Server 2003. This allows the domain controller to provide authentication and logon services for a domain.
- **Active Directory Stopped** Active Directory is stopped, and the domain controller can no longer provide authentication and logon services for a domain. This mode shares some characteristics of both a member server and a domain controller in Directory Services Restore Mode. As with a member server, the server is joined to the domain. Users can log on interactively by using cached credentials, smart cards, and biometric logon methods. Users can also log on over the network by using another domain controller for domain logon. As with Directory Services Restore Mode, the Active Directory database (Ntds.dit) on the local domain controller is offline. This means you can perform offline AD DS operations, such as defragmentation of the database and application of security updates, without having to restart the domain controller.
- **Directory Services Restore Mode** Active Directory is in restore mode. The domain controller has the same restore state as a domain controller running Windows Server 2003. This mode allows you to perform an authoritative or nonauthoritative restore of the Active Directory database.

When working with AD DS in the Stopped state, you should keep in mind that dependent services are also stopped when you stop AD DS. This means that File Replication Service (FRS), Kerberos Key Distribution Center (KDC), and Intersite Messaging are stopped before Active Directory is stopped, and that even if they are running, these dependent services are restarted when Active Directory restarts. Further, you can restart a domain controller in Directory Services Restore Mode, but you cannot start a domain controller in the Active Directory Stopped state. To get to the Stopped state, you must first start the domain controller in the customary way and then stop AD DS.

Name-resolution services

Windows operating systems use name resolution to make it easier to communicate with other computers on a network. Name resolution associates computer names with the numerical IP addresses that are used for network communications. Thus, rather than using long strings of digits, users can access a computer on the network by using a friendly name.

Current Windows operating systems natively support three name-resolution systems:

- Domain Name System (DNS)
- Windows Internet Name Service (WINS)
- Link-Local Multicast Name Resolution (LLMNR)

The sections that follow examine these services.

Using Domain Name System

DNS is a name-resolution service that resolves computer names to IP addresses. Using DNS, the fully qualified host name computer84.cpandl.com, for example, can be resolved to an IP address, which allows it and other computers to find one another. DNS operates over the TCP/IP protocol stack and can be integrated with WINS, Dynamic Host Configuration Protocol (DHCP), and Active Directory Domain Services.

DNS organizes groups of computers into domains. These domains are organized into a hierarchical structure, which can be defined on an Internet-wide basis for public networks or on an enterprise-wide basis for private networks (also known as *intranets* and *extranets*). The various levels within the hierarchy identify individual computers, organizational domains, and top-level domains. For the fully qualified host name computer84.cpandl.com, *computer84* represents the host name for an individual computer, *cpandl* is the organizational domain, and *com* is the top-level domain.

Top-level domains are at the root of the DNS hierarchy; they are also called *root domains*. These domains are organized geographically, by organization type, and by function. Normal domains, such as cpandl.com, are also referred to as *parent domains*. They're called parent domains because they're the parents of an organizational structure. Parent domains can be divided into subdomains that can be used for groups or departments within an organization.

Subdomains are often referred to as *child domains*. For example, the fully qualified domain name (FQDN) for a computer within a human resources group could be jacob.hr.cpandl.com. Here, *jacob* is the host name, *hr* is the child domain, and *cpandl.com* is the parent domain.

Active Directory domains use DNS to implement their naming structure and hierarchy. Active Directory and DNS are tightly integrated, so much so that you should install DNS on the network before you install domain controllers that use Active Directory. During installation of the first domain controller on an Active Directory network, you're given the opportunity to install DNS automatically if a DNS server can't be found on the network. You are also able to specify whether DNS and Active Directory should be fully integrated. In most cases, you should respond affirmatively to both requests. With full integration, DNS information is stored directly in Active Directory. This allows you to take advantage of Active Directory's capabilities.

The difference between partial integration and full integration is very important:

- **Partial integration** With partial integration, the domain uses standard file storage. DNS information is stored in text-based files that end with the .dns extension, and the default location of these files is %SystemRoot%\System32\Dns. Updates to DNS are handled through a single authoritative DNS server. This server is designated as the primary DNS server for the particular domain or an area within a domain called a *zone*. Clients that use dynamic DNS updates through DHCP must be configured to use the primary DNS server in the zone. If they aren't, their DNS information won't be updated. Likewise, dynamic updates through DHCP can't be made if the primary DNS server is offline.
- **Full integration** With full integration, the domain uses directory-integrated storage. DNS information is stored directly in Active Directory and is available through the container for the *dnsZone* object. Because the information is part of Active Directory, any domain controller can access the data, and a multimaster approach can be used for dynamic updates through DHCP. This allows any domain controller running the DNS Server service to handle dynamic updates. Furthermore, clients that use dynamic DNS updates through DHCP can use any DNS server within the zone. An added benefit of directory integration is the ability to use directory security to control access to DNS information.

REAL WORLD **Windows Server 2012 R2 allows DNS clients to register both address (A) and pointer (PTR) records or only A records. A records are used for name-to-IP address lookups, also known as forward lookups; PTR records are used for IP address-to-name lookups, also known as reverse lookups. Being able to register only A records is useful when reverse lookups haven't been configured and you don't want DNS clients to repeatedly try to register PTR records.**

If you look at the way DNS information is replicated throughout the network, you can see more advantages to full integration with Active Directory. With partial integration, DNS information is stored and replicated separately from Active Directory. Having two separate structures reduces the effectiveness of both DNS and Active Directory and makes administration more complex. Because DNS is less efficient than Active Directory at replicating changes, you might also increase network traffic and the amount of time it takes to replicate DNS changes throughout the network.

To enable DNS on the network, you need to configure DNS clients and servers. When you configure DNS clients, you tell the clients the IP addresses of DNS servers on the network. Using these addresses, clients can communicate with DNS servers anywhere on the network, even if the servers are on different subnets.

When the network uses DHCP, you should configure DHCP to work with DNS. To do this, you need to set the DHCP scope options 006 DNS Servers and 015 DNS Domain Name. Additionally, if computers on the network need to be accessible from other Active Directory domains, you need to create records for them in DNS.

DNS records are organized into zones; as mentioned earlier in this chapter, a zone is simply an area within a domain.

When you install the DNS Server service on an RODC, the RODC is able to pull a read-only replica of all application directory partitions that are used by DNS, including *ForestDNSZones* and *DomainDNSZones*. Clients can then query the RODC for name resolution as they would query any other DNS server. However, as with directory updates, the DNS server on an RODC does not support direct updates. This means that the RODC does not register name server (NS) resource records for any Active Directory–integrated zone that it hosts. When a client attempts to update its DNS records against an RODC, the server returns a referral to a DNS server that the client can use for the update. The DNS server on the RODC should receive the updated record from the DNS server that receives details about the update by using a special replicate-single-object request that runs as a background process.

Windows 7 and later releases add support for DNS Security Extensions (DNSSEC). The DNS client running on these operating systems can send queries that indicate support for DNSSEC, process related records, and determine whether a DNS server has validated records on its behalf. On Windows servers, this allows your DNS servers to securely sign zones and to host DNSSEC-signed zones. It also allows DNS servers to process related records and perform both validation and authentication.

Using Windows Internet Name Service

WINS is a service that resolves computer names to IP addresses. Using WINS, the computer name COMPUTER84, for example, can be resolved to an IP address that enables computers on a Microsoft network to find one another and transfer information. WINS is needed to support pre–Windows 2000 systems and earlier applications that use NetBIOS over TCP/IP, such as the .NET command-line utilities. If you don't have pre–Windows 2000 systems or applications on the network, you don't need to use WINS.

WINS works best in client/server environments in which WINS clients send single-label (host) name queries to WINS servers for name resolution and WINS servers resolve the query and respond. When all your DNS servers are running Windows Server 2008 or later, deploying a Global Names zone creates static, global records with single-label names that do not rely on WINS. This allows users to access hosts by using single-label names rather than FQDNs and removes the dependency on WINS. To transmit WINS queries and other information, computers use NetBIOS. NetBIOS provides an application programming interface (API) that allows computers on a network to communicate. NetBIOS applications rely on WINS or the local LMHOSTS file to resolve computer names to IP addresses. On pre–Windows 2000 networks, WINS is the primary name resolution service available. On Windows 2000 and later networks, DNS is the primary name resolution service and WINS has a different function. This function is to allow pre–Windows 2000 systems to browse

lists of resources on the network and to allow Windows 2000 and later systems to locate NetBIOS resources.

To enable WINS name resolution on a network, you need to configure WINS clients and servers. When you configure WINS clients, you tell the clients the IP addresses for WINS servers on the network. Using the IP addresses, clients can communicate with WINS servers anywhere on the network, even if the servers are on different subnets. WINS clients can also communicate by using a broadcast method through which clients broadcast messages to other computers on the local network segment that are requesting their IP addresses. Because messages are broadcast, the WINS server isn't used. Any non-WINS clients that support this type of message broadcasting can also use this method to resolve computer names to IP addresses.

When clients communicate with WINS servers, they establish sessions that have the following three key parts:

- **Name registration** During name registration, the client gives the server its computer name and its IP address and asks to be added to the WINS database. If the specified computer name and IP address aren't already in use on the network, the WINS server accepts the request and registers the client in the WINS database.
- **Name renewal** Name registration isn't permanent. Instead, the client can use the name for a specified period known as a *lease*. The client is also given a time period within which the lease must be renewed, which is known as the *renewal interval*. The client must reregister with the WINS server during the renewal interval.
- **Name release** If the client can't renew the lease, the name registration is released, allowing another system on the network to use the computer name, IP address, or both. The names are also released when you shut down a WINS client.

After a client establishes a session with a WINS server, the client can request name-resolution services. The method used to resolve computer names to IP addresses depends on how the network is configured. The following four name-resolution methods are available:

- **B-node (broadcast)** Uses broadcast messages to resolve computer names to IP addresses. Computers that need to resolve a name broadcast a message to every host on the local network, requesting the IP address for a computer name. On a large network with hundreds or thousands of computers, these broadcast messages can use up valuable network bandwidth.
- **P-node (peer-to-peer)** Uses WINS servers to resolve computer names to IP addresses. As explained earlier, client sessions have three parts: name registration, name renewal, and name release. In this mode, when a client

needs to resolve a computer name to an IP address, the client sends a query message to the server and the server responds with an answer.

- **M-node (mixed)** Combines b-node and p-node. With m-node, a WINS client first tries to use b-node for name resolution. If the attempt fails, the client then tries to use p-node. Because b-node is used first, this method has the same problems with network bandwidth usage as b-node.
- **H-node (hybrid)** Also combines b-node and p-node. With h-node, a WINS client first tries to use p-node for peer-to-peer name resolution. If the attempt fails, the client then tries to use broadcast messages with b-node. Because peer-to-peer is the primary method, h-node offers the best performance on most networks. H-node is also the default method for WINS name resolution.

If WINS servers are available on the network, Windows clients use the p-node method for name resolution. If no WINS servers are available on the network, Windows clients use the b-node method for name resolution. Windows computers can also use DNS and the local files LMHOSTS and HOSTS to resolve network names.

When you use DHCP to assign IP addresses dynamically, you should set the name resolution method for DHCP clients. To do this, you need to set DHCP scope options for the 046 WINS/NBT Node Type. The best method to use is h-node. You'll get the best performance and reduce traffic on the network.

Using Link-Local Multicast Name Resolution

LLMNR fills a need for peer-to-peer name-resolution services for devices with an IPv4 address, an IPv6 address, or both, allowing IPv4 and IPv6 devices on a single subnet without a WINS or DNS server to resolve each other's names—a service that neither WINS nor DNS can fully provide. Although WINS can provide both client/server and peer-to-peer name-resolution services for IPv4, it does not support IPv6 addresses. DNS, alternatively, supports IPv4 and IPv6 addresses, but it depends on designated servers to provide name-resolution services.

Windows 7 and later releases support LLMNR. LLMNR is designed for both IPv4 and IPv6 clients in configurations where other name-resolution systems are not available, such as the following:

- Home or small office networks
- Ad hoc networks
- Corporate networks where DNS services are not available

LLMNR is designed to complement DNS by enabling name resolution in scenarios in which conventional DNS name resolution is not possible. Although LLMNR can replace the need for WINS in cases where NetBIOS is not required, LLMNR is not a substitute for DNS because it operates only on the local subnet. Because LLMNR

traffic is prevented from propagating across routers, it cannot accidentally flood the network.

As with WINS, you use LLMNR to resolve a host name, such as COMPUTER84, to an IP address. By default, LLMNR is enabled on all computers running Windows 7 and later releases, and these computers use LLMNR only when all attempts to look up a host name through DNS fail. As a result, name resolution works like the following for Windows 7 and later releases:

1. A host computer sends a query to its configured primary DNS server. If the host computer does not receive a response or receives an error, it tries each configured alternate DNS server in turn. If the host has no configured DNS servers or fails to connect to a DNS server without errors, name resolution fails over to LLMNR.
2. The host computer sends a multicast query over User Datagram Protocol (UDP) requesting the IP address for the name being looked up. This query is restricted to the local subnet (also referred to as the *local link*).
3. Each computer on the local link that supports LLMNR and is configured to respond to incoming queries receives the query and compares the name to its own host name. If the host name is not a match, the computer discards the query. If the host name is a match, the computer transmits a unicast message containing its IP address to the originating host.

You can also use LLMNR for reverse mapping. With a *reverse mapping*, a computer sends a unicast query to a specific IP address, requesting the host name of the target computer. An LLMNR-enabled computer that receives the request sends a unicast reply containing its host name to the originating host.

LLMNR-enabled computers are required to ensure that their names are unique on the local subnet. In most cases, a computer checks for uniqueness when it starts, when it resumes from a suspended state, and when you change its network interface settings. If a computer has not yet determined that its name is unique, it must indicate this condition when responding to a name query.

REAL WORLD **By default, LLMNR is automatically enabled on computers running Windows 7 and later releases. You can disable LLMNR through registry settings. To disable LLMNR for all network interfaces, create and set the following registry value to 0: HKLM/SYSTEM/CurrentControlSet/Services/Dnscache/Parameters/EnableMulticast.**

To disable LLMNR for a specific network interface, create and set the following registry value to 0: HKLM/SYSTEM/CurrentControlSet/Services/Tcpip/Parameters/AdapterGUID /EnableMulticast.

Here, *AdapterGUID* is the globally unique identifier (GUID) of the network-interface adapter for which you want to disable LLMNR. You can enable LLMNR again at any time by setting these registry values to 1. You also can manage LLMNR through Group Policy.

Frequently used tools

Many utilities are available for administrating Windows Server 2012 R2 systems. The tools you use the most include the following:

- **Control Panel** A collection of tools for managing system configuration. You can organize Control Panel in different ways according to the view you're using. A *view* is simply a way of organizing and presenting options. You change the view by using the View By list. Category view is the default view, and it provides access to tools by category, tool, and key tasks. The Large Icons and Small Icons views are alternative views that list each tool separately by name.
- **Graphical administrative tools** The key tools for managing network computers and their resources. You can access these tools by selecting them individually from the Administrative Tools program group.
- **Administrative wizards** Tools designed to automate key administrative tasks. You can access many administrative wizards in Server Manager—the central administration console for Windows Server 2012 R2.
- **Command-line utilities** You can start most administrative utilities from the command prompt. In addition to these utilities, Windows Server 2012 R2 provides others that are useful for working with Windows Server 2012 R2 systems.

To learn how to use any of the .NET command-line tools, type **NET HELP** at a command prompt followed by the command name, such as **NET HELP SHARE**. Windows Server 2012 R2 then provides an overview of how the command is used.

Windows PowerShell

For additional flexibility in your command-line scripting, you might want to use Windows PowerShell. Windows PowerShell is a full-featured command shell that can use built-in commands called *cmdlets*, built-in programming features, and standard command-line utilities. A command console and a graphical environment are available.

Although the Windows PowerShell console and the graphical scripting environment are installed by default, several other Windows PowerShell features are not installed by default. They include the Windows PowerShell 2.0 engine, which is provided for backward compatibility with existing Windows PowerShell host applications, and Windows PowerShell Web Access, which lets a server act as a web gateway for managing the server remotely by using Windows PowerShell and a web client.

REAL WORLD **You can install these additional Windows PowerShell features by using the Add Roles And Features Wizard. On the desktop, tap or click the Server Manager button on the taskbar. This option is included by default. In Server Manager, tap or click Manage, and then tap or click Add Roles And Features. This runs the Add Roles And Features Wizard, which you use to add these features. Note, however, that with Windows Server 2012 R2, not only can you disable a role or feature, but you also can remove the binaries needed for that role or feature. Binaries needed to install roles and features are referred to as *payloads*.**

The Windows PowerShell console (Powershell.exe) is a 32-bit or 64-bit environment for working with Windows PowerShell at the command line. On 32-bit versions of Windows, you'll find the 32-bit executable in the %SystemRoot%\System32\WindowsPowerShell\v1.0 directory. On 64-bit versions of Windows, you'll find the 32-bit executable in the %SystemRoot%\SysWow64\WindowsPowerShell\v1.0 directory, and the 64-bit executable in the %SystemRoot%\System32\WindowsPowerShell\v1.0 directory.

On the desktop, you can open the Windows PowerShell console by tapping or clicking the Windows PowerShell button on the taskbar. This option is included by default. On 64-bit systems, the 64-bit version of Windows PowerShell is started by default. If you want to use the 32-bit Windows PowerShell console on a 64-bit system, you must select the Windows PowerShell (x86) option.

You can start Windows PowerShell from a Windows command shell (Cmd.exe) by entering the following:

```
powershell
```

> ***NOTE*** **The directory path for Windows PowerShell should be in your command path by default. This ensures that you can start Windows PowerShell from a command prompt without first having to change to the related directory.**

After starting Windows PowerShell, you can enter the name of a cmdlet at the prompt, and the cmdlet will run in much the same way as a command-line command. You can also execute cmdlets in scripts. Cmdlets are named by using verb-noun pairs. The verb tells you what the cmdlet does in general. The noun tells you what specifically the cmdlet works with. For example, the Get-Variable cmdlet gets all Windows PowerShell environment variables and returns their values, or it gets a specifically named environment variable and returns its value. The common verbs associated with cmdlets are as follows:

- **Get-** Queries a specific object or a subset of a type of object, such as a specified performance counter or all performance counters
- **Set-** Modifies specific settings of an object
- **Enable-** Enables an option or a feature
- **Disable-** Disables an option or a feature
- **New-** Creates a new instance of an item, such as a new event or service
- **Remove-** Removes an instance of an item, such as an event or event log

At the Windows PowerShell prompt, you can get a complete list of cmdlets by entering **get-help *-***. To get Help documentation on a specific cmdlet, enter **get-help** followed by the cmdlet name, such as **get-help get-variable**. Windows PowerShell V3 and later use online and updatable Help files. Because of this, you might see only basic syntax for cmdlets and functions. To get full Help details, you'll have to either use online Help or download the Help files to your computer. For online Help, add the -online option to your get-help command, as shown here:

```
get-help get-variable -online
```

Use the Update-Help cmdlet to download and install the current Help files from the Internet. Without parameters, Update-Help updates the Help files for all modules installed on the computer. However, Update-Help does the following:

- Downloads files only once a day
- Installs files only when they are newer than the ones on the computer
- Limits the total size of uncompressed Help files to 1 GB

You can override these restrictions by using the -Force parameter. You can save Help files to the local computer by using Save-Help.

All cmdlets also have configurable aliases that act as shortcuts for executing a cmdlet. To list all aliases available, enter **get-alias** at the Windows PowerShell prompt. You can create an alias that invokes any command by using the following syntax:

```
new-item –path alias:AliasName –value:FullCommandPath
```

Here *AliasName* is the name of the alias to create, and *FullCommandPath* is the full path to the command to run, such as the following:

```
new-item –path alias:sm –value:c:\windows\system32\compmgmtlauncher.exe
```

This example creates the alias *sm* for starting Server Manager. To use this alias, you simply type **sm** and then press Enter when you are working with Windows PowerShell. It's important to note that Windows PowerShell 3 and later versions automatically import required modules the first time you use a related command. With Windows PowerShell 2, you needed to explicitly import a module before you could run any of its commands.

REAL WORLD **Generally speaking, anything you can enter at a command prompt also can be entered at the Windows PowerShell prompt. This is possible because Windows PowerShell looks for external commands and utilities as part of its usual processing. As long as the external command or utility is found in a directory specified by the PATH environment variable, the command or utility is run as appropriate. However, keep in mind that the Windows PowerShell execution order could affect whether a command runs as expected. For Windows PowerShell, the execution order is 1) alternate built-in or profile-defined aliases, 2) built-i Windows n or profile-defined functions, 3) cmdlets or language keywords, 4) scripts with the .ps1 extension, and 5) external commands, utilities, and files. Thus, if any element in 1 through 4 of the execution order has the same name as a command, that element will run instead of the expected command.**

Windows Remote Management

The Windows PowerShell remoting features are supported by the WS-Management protocol and the Windows Remote Management (WinRM) service that implements WS-Management in Windows. Computers running Windows 8 and later, and also Windows Server 2012 or later, include WinRM 3.0 or later. If you want to manage a Windows server from a workstation, you need to be sure that WinRM 3.0 and Windows PowerShell are installed and that the server has a WinRM listener enabled.

A Microsoft Internet Information Services (IIS) extension, installable as a Windows feature called *WinRM IIS Extension*, lets a server act as a web gateway for managing the server remotely by using WinRM and a web client.

Enabling and using WinRM

You can verify the availability of WinRM 3.0 and configure Windows PowerShell for remoting by following these steps:

1. Tap or click Start, and then point to Windows PowerShell. Start Windows PowerShell as an administrator by pressing and holding or right-clicking the Windows PowerShell shortcut and then selecting Run As Administrator.
2. The WinRM service is configured for manual start by default. You must change the startup type to Automatic and start the service on each computer you want to work with. At the Windows PowerShell prompt, you can verify that the WinRM service is running by using the following command:

   ```
   get-service winrm
   ```

   As shown in the following example, the value of the *Status* property in the output should be *Running*:

   ```
   Status   Name               DisplayName
   ------   ----               -----------
   Running  WinRM              Windows Remote Management
   ```

   If the service is stopped, enter the following command to start the service, and then configure it to start automatically in the future:

   ```
   set-service –name winrm –startuptype automatic –status running
   ```

3. To configure Windows PowerShell for remoting, enter the following command:

   ```
   Enable-PSRemoting –force
   ```

   You can enable remoting only when your computer is connected to a domain or a private network. If your computer is connected to a public network, you need to disconnect from the public network and connect to a domain or private network and then repeat this step. If one or more of your computer's connections has the Public Network connection type but you are actually connected to a domain or private network, you need to change the network connection type in Network And Sharing Center and then repeat this step.

In many cases, you are able to work with remote computers in other domains. However, if the remote computer is not in a trusted domain, the remote computer might not be able to authenticate your credentials. To enable authentication, you need to add the remote computer to the list of trusted hosts for the local computer in WinRM. To do so, enter the following:

```
winrm set winrm/config/client '@{TrustedHosts="RemoteComputer"}'
```

Here *RemoteComputer* is the name of the remote computer, such as the following:

```
winrm set winrm/config/client '@{TrustedHosts="CorpServer56"}'
```

When you are working with computers in workgroups or homegroups, you must use HTTPS as the transport or add the remote machine to the TrustedHosts configuration settings. If you cannot connect to a remote host, verify that the service on the remote host is running and is accepting requests by running the following command on the remote host:

```
winrm quickconfig
```

This command analyzes and configures the WinRM service. If the WinRM service is set up correctly, you'll get output similar to the following:

```
WinRM already is set up to receive requests on this machine.
WinRM already is set up for remote management on this machine.
```

If the WinRM service is not set up correctly, you receive errors and need to respond affirmatively to several prompts that allow you to automatically configure remote management. When this process is complete, WinRM should be set up correctly.

Whenever you use Windows PowerShell remoting features, you must start Windows PowerShell as an administrator by pressing and holding or right-clicking the Windows PowerShell shortcut and then selecting Run As Administrator. When starting Windows PowerShell from another program, such as the command prompt, you must start that program as an administrator.

Configuring WinRM

When you are working with an elevated, administrator command prompt, you can use the WinRM command-line utility to view and manage the remote management configuration. Enter **winrm get winrm/config** to display detailed information about the remote management configuration.

If you examine the configuration listing, you'll notice there is a hierarchy of information. The base of this hierarchy, the Config level, is referenced with the path winrm/config. Then there are sublevels for client, service, and WinRS, referenced as winrm/config/client, winrm/config/service, and winrm/config/winrs, respectively. You can change the value of most configuration parameters by using the following command:

```
winrm set ConfigPath @{ParameterName="Value"}
```

Here *ConfigPath* is the configuration path, *ParameterName* is the name of the parameter you want to work with, and *Value* sets the value for the parameter, as shown in the following example:

```
winrm set winrm/config/winrs @{MaxShellsPerUser="10"}
```

Here, you set the *MaxShellsPerUser* parameter under winrm/config/winrs. This parameter controls the maximum number of connections to a remote computer that can be active per user. (By default, each user can have only five active connections.) Keep in mind that some parameters are read-only and cannot be set in this way.

WinRM requires at least one listener to indicate the transports and IP addresses on which management requests can be accepted. The transport must be HTTP, HTTPS, or both. With HTTP, messages can be encrypted by using NTLM or Kerberos encryption. With HTTPS, Secure Sockets Layer (SSL) is used for encryption. You can examine the configured listeners by entering **winrm enumerate winrm/config /listener**. As Listing 1-1 shows, this command displays the configuration details for configured listeners.

LISTING 1-1 Sample configuration for listeners

```
Listener
    Address = *
    Transport = HTTP
    Port = 80
    Hostname
    Enabled = true
    URLPrefix = wsman
    CertificateThumbprint
    ListeningOn = 127.0.0.1, 192.168.1.225
```

By default, your computer is probably configured to listen on any IP address. If so, no output will be displayed. To limit WinRM to specific IP addresses, the computer's local loopback address (127.0.0.1) and assigned IPv4 and IPv6 addresses can be explicitly configured for listening. You can configure a computer to listen for requests over HTTP on all configured IP addresses by entering the following:

```
winrm create winrm/config/listener?Address=*+Transport=HTTP
```

You can listen for requests over HTTPS on all IP addresses configured on the computer by entering the following:

```
winrm create winrm/config/listener?Address=*+Transport=HTTPS
```

Here, the asterisk (*) indicates all configured IP addresses. Note that the *CertificateThumbprint* property must be empty to share the SSL configuration with another service.

You can enable or disable a listener for a specific IP address by entering the following:

```
winrm set winrm/config/listener?Address=IP:192.168.1.225+Transport=HTTP @
{Enabled="true"}
```

or

```
winrm set winrm/config/listener?Address=IP:192.168.1.225+Transport=HTTP @
{Enabled="false"}
```

You can enable or disable basic authentication on the client by entering the following:

```
winrm set winrm/config/client/auth @{Basic="true"}
```

or

```
winrm set winrm/config/client/auth @{Basic="false"}
```

You can enable or disable Windows authentication using either NTLM or Kerberos (as appropriate) by entering the following:

```
winrm set winrm/config/client @{TrustedHosts="<local>"}
```

or

```
winrm set winrm/config/client @{TrustedHosts=""}
```

In addition to managing WinRM at the command line, you can manage the service by using Group Policy. As a result, Group Policy settings might override any settings you enter.

CHAPTER 7

Using Active Directory

Active Directory Domain Services (AD DS) is an extensible and scalable directory service you can use to efficiently manage network resources. As an administrator, you need to be deeply familiar with how Active Directory technology works, and that's exactly what this chapter is about. If you haven't worked with Active Directory technology before, you'll notice immediately that the technology is fairly advanced and has many features.

Introducing Active Directory

For many years now, Active Directory has been the heart of Windows-based domains. Just about every administrative task you perform affects Active Directory in some way. Active Directory technology is based on standard Internet protocols and is designed to help you clearly define your network's structure.

Active Directory and DNS

Active Directory uses Domain Name System (DNS). *DNS* is a standard Internet service that organizes groups of computers into domains. DNS domains are organized into a hierarchical structure. The DNS domain hierarchy is defined on an Internet-wide basis, and the different levels within the hierarchy identify computers, organizational domains, and top-level domains. DNS is also used to map host names to numeric TCP/IP addresses. Through DNS, an Active Directory domain hierarchy can also be defined on an Internet-wide basis, or the domain hierarchy can be separate from the Internet and private.

When you refer to computer resources in a DNS domain, you use a fully qualified domain name (FQDN), such as zeta.microsoft.com. Here, *zeta* represents the name of an individual computer, *microsoft* represents the organizational domain, and *com* is the top-level domain. *Top-level domains (TLDs)* are at the base of the DNS hierarchy. TLDs are organized geographically by using two-letter country codes, such as *CA* for Canada; by organization type, such as *com* for commercial organizations; and by function, such as *mil* for US military installations.

Normal domains, such as microsoft.com, are also referred to as *parent domains* because they're the parents of an organizational structure. You can divide parent domains into subdomains, which you can then use for different offices, divisions, or geographic locations. For example, the FQDN for a computer at the Seattle office for Microsoft could be designated as jacob.seattle.microsoft.com. Here, *jacob* is the computer name, *seattle* is the subdomain, and *microsoft.com* is the parent domain. Another term for a subdomain is a *child domain*.

DNS is an integral part of Active Directory technology—so much so that you must configure DNS on the network before you can install Active Directory. With Windows Server 2012 R2, you install Active Directory in a two-part process. First, you start the process in Server Manager by tapping or clicking Manage and then clicking Add Roles And Features. This runs the Add Roles And Features Wizard, which you use to specify that you want to add the AD DS role to the server. The wizard installs binaries needed for the role, and the progress of this process is shown on the Installation Progress page.

REAL WORLD **Binaries needed to install roles and features are referred to as *payloads*. With Windows Server 2012 R2, not only can you uninstall a role or feature, but you also can uninstall and remove the payload for that feature or role by using the –Remove parameter of the Uninstall-WindowsFeature cmdlet.**

You can restore a removed payload by using the Install-WindowsFeature cmdlet. By default, payloads are restored via Windows Update. Use the –Source parameter to restore a payload from a WIM mount point. In the following example, you restore the AD DS binaries and all related subfeatures via Windows Update:

```
install-windowsfeature -name ad-domain-services
-includeallsubfeature
```

When the installation completes, you start the Active Directory Domain Services Configuration Wizard by tapping or clicking the Promote This Server To A Domain Controller link on the Installation Progress page, and then you use this wizard to configure the role. This wizard replaces Dcpromo.exe, which was used previously for promoting domain controllers. The wizard also will run Adprep.exe to prepare schema as appropriate. If you have not run Adprep.exe separately previously, and you are installing the first domain controller that runs Windows Server 2012 R2 in an existing domain or forest, the wizard will prompt you to supply credentials to run Adprep commands. To prepare a forest, you need to provide credentials for a member of the Enterprise Admins group, the Schema Admins group, and the Domain Admins group in the domain that hosts the schema master. To prepare a domain, you need to provide credentials for a member of the Domain Admins

group. If you are installing the first read-only domain controller (RODC) in a forest, you need to provide credentials for a member of the Enterprise Admins group.

If DNS isn't already installed, you are prompted to install it. If no domain exists, the wizard helps you create a domain and configure Active Directory in the new domain. The wizard can also help you add child domains to existing domain structures. To verify that a domain controller is installed correctly, do the following:

- Check the Directory Service event log for errors.
- Ensure that the SYSVOL folder is accessible to clients.
- Verify that name resolution is working through DNS.
- Verify the replication of changes to Active Directory.

NOTE **In the rest of this chapter, I'll use the terms *directory* and *domains* to refer to Active Directory and Active Directory domains, respectively, except when I need to distinguish Active Directory structures from DNS or other types of directories.**

Keep in mind that when you use Server Manager for Windows Server 2012 R2 and the forest functional level is Windows Server 2003 or higher, any necessary preparations are done automatically when you deploy a domain controller. This means the Configuration Wizard automatically updates the Active Directory schema for the forest and domain so that the schema is compatible with Windows Server 2012 R2 as necessary.

Read-only domain controller deployment

When the domain and forest are operating at the Windows Server 2003 functional level or higher, and your primary domain controller (PDC) emulator for a domain is running Windows Server 2008 or later, you can deploy read-only domain controllers (RODCs). Any domain controller running Windows Server 2008 R2 or later can be configured as an RODC. When you install the DNS Server service on an RODC, the RODC can act as a read-only DNS (RODNS) server. In this configuration, the following conditions are true:

- The RODC replicates the application directory partitions that DNS uses, including the ForestDNSZones and DomainDNSZones partitions. Clients can query an RODNS server for name resolution. However, the RODNS server does not support client updates directly, because the RODNS server does not register resource records for any Active Directory–integrated zone that it hosts.
- When a client attempts to update its DNS records, the server returns a referral. The client can then attempt to update against the DNS server that is provided in the referral. Through replication in the background, the RODNS server then attempts to retrieve the updated record from the DNS server that made the update. This replication request is only for the changed DNS record. The entire list of data changed in the zone or domain is not replicated during this special request.

The first Windows Server 2008 R2 or later domain controller installed in a forest or domain cannot be an RODC. However, you can configure subsequent domain controllers as read-only.

MORE INFO **The domain and forest must have the correct schema level to support RODCs and must also be prepared to work with RODCs. Previously, in some cases, this required that you prepare the forest and domain schemas for Windows Server and then update the forest schema again for RODCs. When you use Server Manager for Windows Server 2012 R2, and the Windows Server 2003 or higher forest functional level, any necessary preparations are done automatically as part of domain controller (DC) and RODC deployment.**

Features introduced with Windows Server 2008 R2

When you are using Windows Server 2008 R2 and later in your enterprise and have deployed these operating systems on all domain controllers throughout the domains in your Active Directory forest, your domains can operate at the Windows Server 2008 R2 or higher domain functional level, and the forest can operate at the Windows Server 2008 R2 or higher forest functional level. These operating levels allow you to take advantage of the many Active Directory enhancements that improve manageability, performance, and supportability, including the following:

- **Active Directory Recycle Bin** Allows administrators to undo the accidental deletion of Active Directory objects in much the same way as they can recover deleted files from the Windows Recycle Bin. For more information, see the "Using the Active Directory Recycle Bin" section later in this chapter.
- **Managed service accounts** Introduces a special type of domain user account for managed services that reduces service outages and other issues by having Windows manage the account password and related service principal names (SPNs) automatically. For more information, see the "Implementing managed accounts" section in Chapter 9, "Creating user and group accounts."
- **Managed virtual accounts** Introduces a special type of local computer account for managed services that provides the ability to access the network by using a computer identity in a domain environment. For more information, see the "Using virtual accounts" section in Chapter 9.

REAL WORLD **Technically, you can use managed service accounts and managed virtual accounts in a mixed-mode domain environment. However, you have to manually manage SPNs for managed service accounts, and the Active Directory schema must be compatible with Windows Server 2008 R2 and higher.**

- **Authentication Mechanism Assurance** Improves the authentication process by allowing administrators to control resource access based on whether a user logs on by using a certificate-based logon method. Thus, an administrator can specify that a user has one set of access permissions when logged on by using a smart card and a different set of access permissions when not logged on by using a smart card.

Other improvements don't require that you raise domain or forest functional levels, but they do require that you use Windows Server 2008 or later. These improvements include the following:

- **Offline domain join** Allows administrators to preprovision computer accounts in the domain to prepare operating systems for deployment. This allows computers to join a domain without having to contact a domain controller.
- **Active Directory module for Windows PowerShell** Provides cmdlets for managing Active Directory when you are working with Windows PowerShell.
- **Active Directory Administrative Center** Provides a task-orientated interface for managing Active Directory. In Server Manager, tap or click Tools, and then tap or click Active Directory Administrative Center.
- **Active Directory Web Services** Introduces a web service interface for Active Directory domains.

These features are discussed in more detail in Chapter 8, "Core Active Directory administration."

Features introduced with Windows Server 2012

Active Directory Domain Service in Windows Server 2012 has many additional features that give administrators more options for implementing and managing Active Directory. Table 7-1 lists key features. At the least, these features require that you update the Active Directory schema in your forests and domains for Windows Server 2012 or later. You also might need to update the domain, forest, or both functional levels to the Windows Server 2012 or Windows Server 20212 R2 operating level.

TABLE 7-1 Key Active Directory Features for Windows Server 2012

FEATURE	BENEFITS	REQUIREMENTS
Active Directory–based activation	Allows you to use Active Directory to automatically activate clients running Windows 8, Windows 8.1, Windows Server 2012, and Windows Server 2012 R2. Any client connected to the service is activated.	Volume Licensing; Active Directory schema must be updated for at least Windows Server 2012; key is set using Volume Activation server role or command line.
Authentication policies	Allow you to specify access control conditions that restrict the devices that can request Kerberos tickets for user, service, and computer accounts. Applies only to users, computers, managed services accounts, and group managed service accounts.	Accounts must be members of the Protected Users group. Domain controllers must run Windows Server 2012 R2. Windows Server 2012 R2 domain functional level.
Claims-based policy controls	Allow access and audit policies to be defined flexibly.	Claims policy must be enabled for Default Domain Controllers Policy; file servers must run at least Windows Server 2012; domain must have at least one domain controller running Windows Server 2012 or later.
Deferred index creation	Allows deferring of index creation within the directory until UpdateSchemaNow is received or the domain controller is rebooted.	Domain controller must run at least Windows Server 2012.
Enhanced Fine-Grained Password Policy	Allows administrators to use Active Directory Administrative Center for Windows Server 2012 RTM or R2 to create and manage Password Settings objects (PSOs).	Windows Server 2008 or higher domain functional level.

FEATURE	BENEFITS	REQUIREMENTS
Enhanced Recycle Bin	Allows administrators to recover deleted objects by using Active Directory Administrative Center for Windows Server 2012 RTM or R2.	Domain must have Recycle Bin enabled and Windows Server 2008 R2 or higher forest functional level.
Group Managed Service Accounts	Allow multiple services to share a single managed service account.	Active Directory schema must be updated for at least Windows Server 2012; must have at least one domain controller running Windows Server 2012 or later; services must run on Windows Server 2012 RTM or R2.
Kerberos constrained delegation across domains	Allows managed service accounts to act on behalf of users across domains and forests.	Each affected domain must have at least one domain controller running Windows Server 2012 RTM or R2; front-end server must run Windows Server 2012 RTM or R2; back-end server must run Windows Server 2003 or later; and other additional requirements.
Kerberos with Armoring	Improves domain security; allows a domain-joined client and domain controller to communicate over a protected channel.	Domain controllers must be running Windows Server 2012 RTM or Windows Server 2012 R2; Windows Server 2012 RTM or R2 domain functional level; on clients, enable Require FAST policy; on domain controllers, enable Support CBAC And Kerberos Armoring policy.
Off-premises domain join	Allows a computer to be domain-joined over the Internet.	Domain must be Direct Access–enabled, and domain controllers must run Windows Server 2012 RTM or R2.

FEATURE	BENEFITS	REQUIREMENTS
Protected Users security group	Provides additional protections against authentication threats by requiring accounts that are members of this group to use only Kerberos for authentication.	Domain controllers must run Windows Server 2012 R2. Windows Server 2012 R2 domain functional level.
Relative ID (RID) soft ceiling and warnings	Adds warnings as global RID space is used up. Adds a soft ceiling of 900 million RIDs used that prevents RIDs from being issued until administrator overrides.	A domain controller with RID role must run Windows Server 2012 RTM or R2, and domain controllers must run Windows Server 2012 or later.
Server Manager integration	Allows you to perform all the steps required to deploy local and remote domain controllers.	Windows Server 2012 or later; forest functional level of Windows Server 2003 or higher.
Virtual domain controller cloning	Allows you to safely deploy virtualized replicas of domain controllers. Also helps maintain domain controller state.	A domain controller with the PDC emulator role must run Windows Server 2012 or later, and virtual domain controllers must also run Windows Server 2012 or later.

Working with domain structures

Active Directory provides both logical and physical structures for network components. *Logical structures* help you organize directory objects and manage network accounts and shared resources. Logical structures include the following:

- **Organizational units** A subgroup of domains that often mirrors the organization's business or functional structure
- **Domains** A group of computers that share a common directory database
- **Domain trees** One or more domains that share a contiguous namespace
- **Domain forests** One or more domain trees that share common directory information

Physical structures serve to facilitate network communication and to set physical boundaries around network resources. Physical structures that help you map the physical network structure include the following:

- **Subnets** A network group with a specific IP address range and network mask.
- **Sites** One or more subnets. Sites are used to configure directory access and replication.

Understanding domains

An Active Directory domain is simply a group of computers that share a common directory database. Active Directory domain names must be unique. For example, you can't have two microsoft.com domains, but you can have a parent domain microsoft.com, with the child domains seattle.microsoft.com and ny.microsoft.com. If the domain is part of a private network, the name assigned to a new domain must not conflict with any existing domain name on the private network. If the domain is part of the Internet, the name assigned to a new domain must not conflict with any existing domain name throughout the Internet. To ensure uniqueness on the Internet, you must register the parent domain name before using it. You can register a domain through any designated registrar. You can find a current list of designated registrars at InterNIC (*www.internic.net*).

Each domain has its own security policies and trust relationships with other domains. Domains can also span more than one physical location, which means that a domain can consist of multiple sites and those sites can have multiple subnets, as shown in Figure 7-1. Within a domain's directory database, you'll find objects defining accounts for users, groups, computers, and shared resources such as printers and folders.

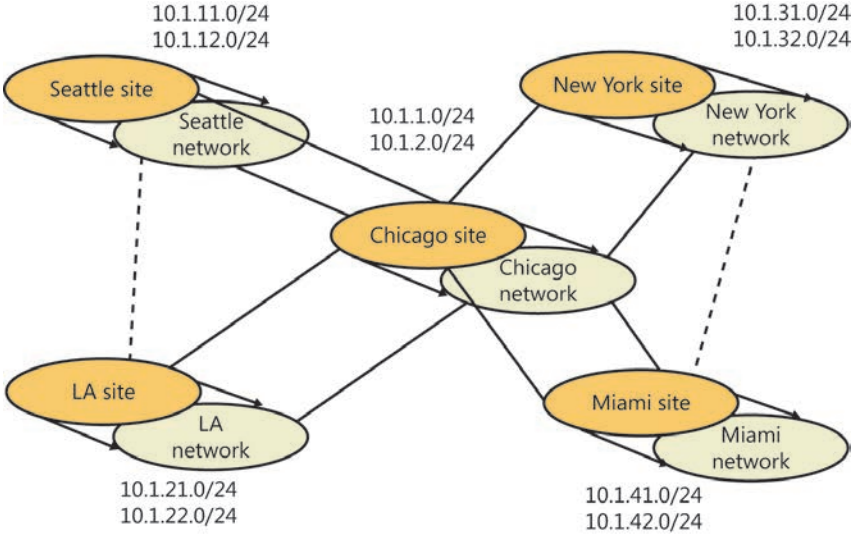


FIGURE 7-1 This network diagram depicts a wide area network (WAN) with multiple sites and subnets.

NOTE **User and group accounts are discussed in Chapter 9. Computer accounts and the various types of computers used in Windows Server domains are discussed in the "Working with Active Directory domains" section later in this chapter.**

Domain functions are limited and controlled by the domain functional level. Several domain functional levels are available, including the following:

- **Windows Server 2003** Supports domain controllers running Windows Server 2003 and later
- **Windows Server 2008** Supports domain controllers running Windows Server 2008 and later
- **Windows Server 2008 R2** Supports domain controllers running Windows Server 2008 R2 and later
- **Windows Server 2012** Supports domain controllers running Windows Server 2012 and later
- **Windows Server 2012 R2** Supports domain controllers running Windows Server 2012 R2

For further discussion of domain functional levels, see the "Working with domain functional levels" section later in this chapter.

Understanding domain forests and domain trees

Each Active Directory domain has a DNS domain name, such as microsoft.com. One or more domains sharing the same directory data are referred to as a *forest*. The domain names within this forest can be noncontiguous or contiguous in the DNS naming hierarchy.

When domains have a contiguous naming structure, they're said to be in the same *domain tree*. Figure 7-2 shows an example of a domain tree. In this example, the root domain adatum.com has two child domains: seattle.adatumcom and ny.adatum.com. These domains, in turn, have subdomains. All the domains are part of the same tree because they have the same root domain.

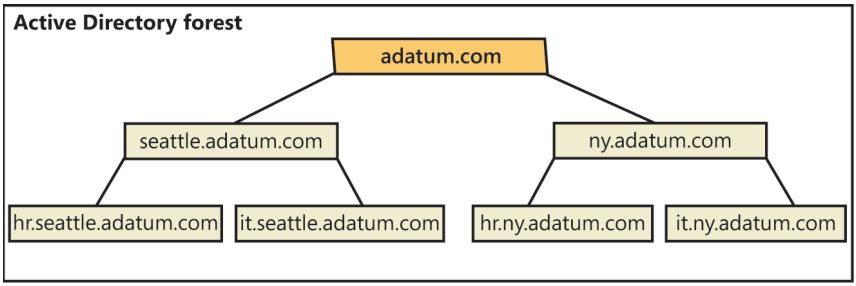


FIGURE 7-2 Domains in the same tree share a contiguous naming structure.

If the domains in a forest have noncontiguous DNS names, they form separate domain trees within the forest. As shown in Figure 7-3, a domain forest can have one or more domain trees. In this example, the adatum.com and microsoft.com domains form the roots of separate domain trees in the same forest.

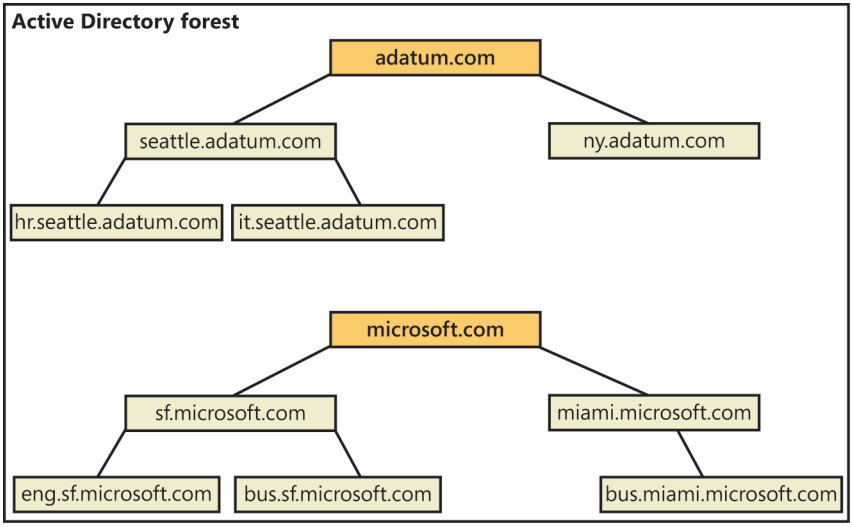


FIGURE 7-3 Multiple trees in a forest with noncontiguous naming structures.

You can access domain structures by using Active Directory Domains And Trusts, shown in Figure 7-4. Active Directory Domains And Trusts is a snap-in for the Microsoft Management Console (MMC). You can also start it from the Tools menu in Server Manager. You'll find separate entries for each root domain. In Figure 7-4, the active domain is cpandl.com.

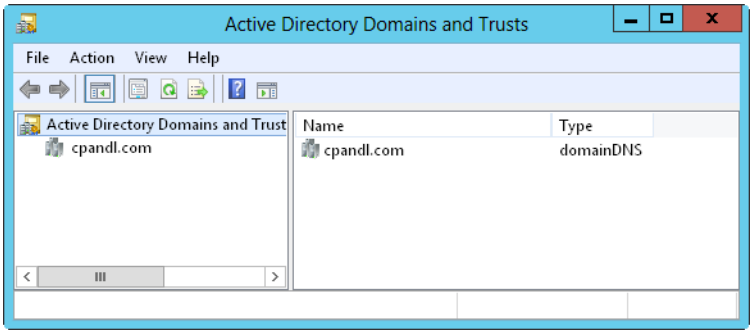


FIGURE 7-4 Use Active Directory Domains And Trusts to work with domains, domain trees, and domain forests.

Forest functions are limited and controlled by the forest functional level. Several forest functional levels are available, including the ones listed here:

- **Windows Server 2003** Supports domain controllers running Windows Server 2003 and later
- **Windows Server 2008** Supports domain controllers running Windows Server 2008 and later

- **Windows Server 2008 R2** Supports domain controllers running Windows Server 2008 R2 and later
- **Windows Server 2012** Supports domain controllers running Windows Server 2012 and later
- **Windows Server 2012 R2** Supports domain controllers running Windows Server 2012 R2

When all domains within a forest are operating in Windows Server 2003 forest functional level, you'll notice improvements over earlier implementations in global catalog replication and replication efficiency. Because link values are replicated, you might also notice improved intersite replication. You can deactivate schema class objects and attributes; use dynamic auxiliary classes; rename domains; and create one-way, two-way, and transitive forest trusts.

The Windows Server 2008 forest functional level offers incremental improvements over the Windows Server 2003 forest functional level in Active Directory performance and features. When all domains within a forest are operating in this mode, you'll notice improvements in both intersite and intrasite replication throughout the organization. Domain controllers can also use Distributed File System (DFS) replication rather than file replication service (FRS) replication. In addition, Windows Server 2008 security principals are not created until the PDC emulator operations master in the forest root domain is running Windows Server 2008.

The Windows Server 2008 R2 forest functional level has several additional features. These features include the Active Directory Recycle Bin, managed service accounts, and Authentication Mechanism Assurance.

Although Active Directory for Windows Server 2012 has many enhancements, most of these enhancements require using only Windows Server 2012 domain controllers and schema. The main exception is for Kerberos with Armoring, which requires the Windows Server 2012 domain functional level. Because the key Active Directory enhancements introduced with Windows Server 2012 R2 require the Windows Server 2012 R2 domain functional level, you'll need to use only Windows Server 2012 R2 domain controllers and schema if you want to take advantage of authentication policies and protected users.

Generally, you cannot lower the forest functional level after you raise it. However, when you raise the forest functional level to Windows Server 2012 or higher, you can lower it. If you are using Windows Server 2012 forest functional level, you can lower it to Windows Server 2008 R2. If you are using Windows Server 2012 R2 forest functional level, you can lower it to Windows Server 2012 or Windows Server 2008 R2. Additionally, with either scenario, if Active Directory Recycle Bin has not been enabled, you can lower the forest functional level to Windows Server 2008. You cannot roll the domain functional level back to Windows Server 2003 or lower.

Understanding organizational units

Organizational units (OUs) are subgroups within domains that often mirror an organization's functional or business structure. You can also think of OUs as logical containers into which you place accounts, shared resources, and other OUs. For

example, you could create OUs named HumanResources, IT, Engineering, and Marketing for the microsoft.com domain. You could later expand this scheme to include child units. Child OUs for Marketing could include OnlineSales, ChannelSales, and PrintSales.

Objects placed in an OU can come only from the parent domain. For example, OUs associated with seattle.microsoft.com can contain objects for this domain only. You can't add objects from ny.microsoft.com to these containers, but you could create separate OUs to mirror the business structure of seattle.microsoft.com.

OUs are helpful in organizing objects to reflect a business or functional structure. Still, this isn't the only reason to use OUs. Other reasons include the following:

- OUs allow you to assign group policies to a small set of resources in a domain without applying the policies to the entire domain. This helps you set and manage group policies at the appropriate level in the enterprise.
- OUs create smaller, more manageable views of directory objects in a domain. This helps you manage resources more efficiently.
- OUs allow you to delegate authority and to easily control administrative access to domain resources. This helps you control the scope of administrator privileges in the domain. You could grant user A administrative authority for one OU and not for others. Meanwhile, you could grant user B administrative authority for all OUs in the domain.

OUs are represented as folders in Active Directory Users And Computers, as shown in Figure 7-5. This utility is a snap-in for the MMC, and you can also start it from the Tools menu in Server Manager.

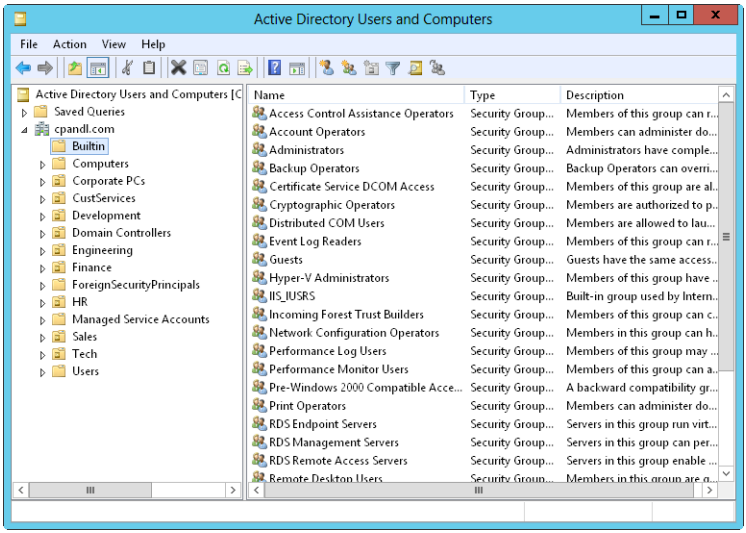


FIGURE 7-5 Use Active Directory Users And Computers to manage users, groups, computers, and organizational units.

Understanding sites and subnets

A *site* is a group of computers in one or more IP subnets. You use sites to map your network's physical structure. Site mappings are independent of logical domain structures, so there's no necessary relationship between a network's physical structure and its logical domain structure. With Active Directory, you can create multiple sites within a single domain or create a single site that serves multiple domains. The IP address ranges used by a site and the domain namespace also have no connection.

You can think of a subnet as a group of network addresses. Unlike sites, which can have multiple IP address ranges, subnets have a specific IP address range and network mask. Subnet names are shown in the form *network/bits-masked*, such as 192.168.19.0/24. In this case, the network address 192.168.19.9 and network mask 255.255.255.0 are combined to create the subnet name 192.168.19.0/24.

> ***NOTE*** **Don't worry—you don't need to know how to create a subnet name. In most cases, you enter the network address and the network mask, and then Windows Server generates the subnet name for you.**

Computers are assigned to sites based on their location in a subnet or a set of subnets. If computers in subnets can communicate efficiently with one another over the network, they're said to be *well connected*. Ideally, sites consist of subnets and computers that are all well connected. If the subnets and computers aren't well connected, you might need to set up multiple sites. Being well connected gives sites several advantages:

- When clients log on to a domain, the authentication process first searches for domain controllers that are in the same site as the client. This means that local domain controllers are used first, if possible, which localizes network traffic and can speed up the authentication process.
- Directory information is replicated more frequently within sites than between sites. This reduces the network traffic load caused by replication while ensuring that local domain controllers get up-to-date information quickly. You can also use site links to customize how directory information is replicated between sites. A domain controller that is designated to perform intersite replication is called a *bridgehead server*. By designating a bridgehead server to handle replication between sites, you place the bulk of the intersite replication burden on a specific server rather than on any available server in a site.

You access sites and subnets through Active Directory Sites And Services, shown in Figure 7-6. Because this is a snap-in for the MMC, you can add it to any updateable console. You can also open Active Directory Sites And Services from the Tools menu in Server Manager.

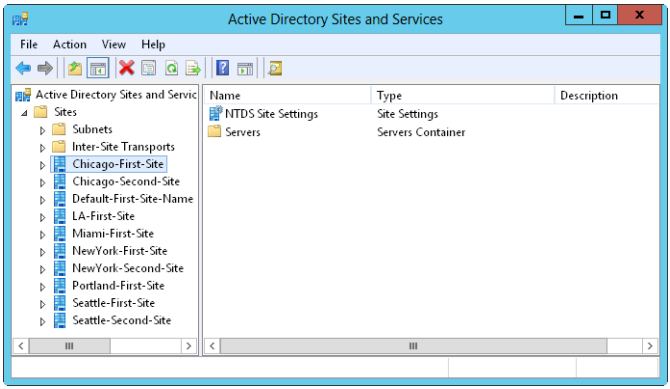


FIGURE 7-6 Use Active Directory Sites And Services to manage sites and subnets.

Working with Active Directory domains

Although you must configure both Active Directory and DNS on a Windows Server network, Active Directory domains and DNS domains have different purposes. Active Directory domains help you manage accounts, resources, and security. DNS domains establish a domain hierarchy that is primarily used for name resolution. Windows Server uses DNS to map host names, such as zeta.microsoft.com, to numeric TCP/IP addresses, such as 172.16.18.8.

Using computers with Active Directory

User computers running professional or business editions of Windows can make full use of Active Directory. These computers access the network as Active Directory clients and have full use of Active Directory features. As clients, these systems can use transitive trust relationships that exist within the domain tree or forest. A transitive trust is one that isn't established explicitly. Rather, the trust is established automatically based on the forest structure and permissions set in the forest. These relationships allow authorized users to access resources in any domain in the forest.

Server computers provide services to other systems and can act as domain controllers or member servers. A domain controller is distinguished from a member server because it runs Active Directory Domain Services. You promote member servers to domain controllers by installing Active Directory Domain Services. You demote domain controllers to member servers by uninstalling Active Directory Domain Services. You use the Add Role And Features and Remove Role And Features wizards to add or remove Active Directory Domain Services. You promote or demote a server through the Active Directory Installation Wizard (Dcpromo.exe).

Domains can have one or more domain controllers. When a domain has multiple domain controllers, the controllers automatically replicate directory data with one another by using a multimaster replication model. This model allows any domain controller to process directory changes and then replicate those changes to other domain controllers.

Because of the multimaster domain structure, all domain controllers have equal responsibility by default. You can, however, give some domain controllers precedence over others for certain tasks, such as specifying a bridgehead server that has priority in replicating directory information to other sites. In addition, some tasks are best performed by a single server. A server that handles this type of task is called an *operations master*. There are five flexible single master operations (FSMO) roles, and you can assign each to a different domain controller. For more information, see the "Understanding operations master roles" section later in this chapter.

Every Windows-based computer that joins a domain has a computer account. Like other resources, computer accounts are stored in Active Directory as objects. You use computer accounts to control access to the network and its resources. A computer accesses a domain by using its account, which is authenticated before the computer can access the network.

REAL WORLD **Domain controllers use Active Directory's global catalog to authenticate both computer and user logons. If the global catalog is unavailable, only members of the Domain Admins group can log on to the domain because the universal group membership information is stored in the global catalog, and this information is required for authentication. In Windows Server 2003 and later servers, you have the option of caching universal group membership locally, which solves this problem. For more information, see the "Understanding the directory structure" section later in this chapter.**

Working with domain functional levels

To support domain structures, Active Directory includes support for the following domain functional levels:

- **Windows Server 2003 mode** When the domain is operating in Windows Server 2003 mode, the directory supports domain controllers running Windows Server 2003 or later. A domain operating in Windows Server 2003 mode can use universal groups, group nesting, group type conversion, easy domain controller renaming, update logon time stamps, and Kerberos KDC key version numbers.
- **Windows Server 2008 mode** When the domain is operating in Windows Server 2008 mode, the directory supports domain controllers running Windows Server 2008 or later. Domain controllers running Windows Server 2003 are no longer supported. A domain operating in Windows Server 2008 mode can use additional Active Directory features, including the DFS replication service for enhanced intersite and intrasite replication.

- **Windows Server 2008 R2 mode** When the domain is operating in Windows Server 2008 R2 mode, the directory supports domain controllers running Windows Server 2008 R2 or later. Domain controllers running Windows Server 2003 and Windows Server 2008 are no longer supported. A domain operating in Windows Server 2008 R2 mode can use Active Directory Recycle Bin, managed service accounts, Authentication Mechanism Assurance, and other important Active Directory enhancements.
- **Windows Server 2012 mode** When the domain is operating in Windows Server 2012 mode, the directory supports domain controllers running Windows Server 2012 or later. Domain controllers running Windows Server 2003, Windows Server 2008, and Windows Server 2008 R2 are no longer supported. Active Directory schema for Windows Server 2012 includes many enhancements, but only the Kerberos with Armoring feature requires this mode.
- **Windows Server 2012 R2 mode** When the domain is operating in Windows Server 2012 R2 mode, the directory supports domain controllers running Windows Server 2012 R2. Domain controllers running Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, and Windows Server 2012 are no longer supported. Active Directory schema for Windows Server 2012 R2 includes enhancements, but only authentication policies and protected users require this mode.

Generally, you cannot lower the domain functional level after you raise it. However, when you raise the domain functional level to Windows Server 2008 R2 or higher, and the forest functional level is Windows Server 2008 or lower, you have the option of rolling the domain functional level back to Windows Server 2008 or Windows Server 2008 R2. You cannot roll the domain functional level back to Windows Server 2003 or lower.

Using the Windows Server 2003 functional level

Every domain in your enterprise should be operating at the Windows Server 2003 functional level or higher, if possible, which will ensure computers in your domains can take advantage of many of the most recent enhancements to Active Directory. After you decommission Windows NT structures and upgrade the Windows 2000 structures in your organization, you can change the functional level to Windows Server 2003 mode operations.

Before updating Windows 2000 domain controllers, you should prepare the domain for upgrade. To do this, you need to update the forest and the domain schema so that they are compatible with Windows Server 2003 domains. A tool called Adprep.exe is provided to automatically perform the update for you. All you need to do is run the tool on the schema operations master in the forest and then on the infrastructure operations master for each domain in the forest. As always, you should test any procedure in a lab before performing it in a production environment.

On the Windows Server 2003 installation media, you'll find Adprep and related files in the i386 subfolder. Follow these steps to perform the upgrade:

1. On the schema operations master in the forest, run **<*cdrom*>:\support\adprep\adprep.exe /forestprep**. You need to use an administrator account that is a member of Enterprise Admins, Schema Admins, or Domain Admins in the forest root domain.
2. On the infrastructure operations master for each domain in the forest, run **<*cdrom*>:\support\adprep\adprep.exe /domainprep**. You need to use an account that is a member of the Domain Admins group in an applicable domain.

NOTE **To determine which server is the current schema operations master for the domain, open a command prompt and enter dsquery server -hasfsmo schema. A directory service path string is returned containing the name of the server, such as "CN =CORPSERVER01,CN=Servers,CN=Default-First-Site-Name,CN=Sites, CN=Configuration,DC=microsoft,DC=com." This string tells you that the schema operations master is CORPSERVER01 in the microsoft.com domain.**

NOTE **To determine which server is the current infrastructure operations master for the domain, start a command prompt and enter dsquery server -hasfsmo infr.**

REAL WORLD **Generally, anything you can enter at a command prompt can be entered at the Windows PowerShell prompt, too. This is possible because Windows PowerShell looks for external commands and utilities as part of its normal processing. As long as the external command or utility is found in a directory specified by the PATH environment variable, the command or utility is run as appropriate. However, keep in mind that the Windows PowerShell execution order could affect whether a command runs as expected. For Windows PowerShell, the execution order is 1) alternate built-in or profile-defined aliases, 2) built-in or profile-defined functions, 3) cmdlets or language keywords, 4) scripts with the .ps1 extension, and 5) external commands, utilities, and files. Thus, if any element in steps 1 through 4 of the execution order has the same name as a command, that element will run instead of the expected command.**

After upgrading your servers, you can raise the domain and forest functionality to take advantage of the additional Active Directory features of the Windows Server 2003 functional level. Keep in mind that after you upgrade, you can use only Windows Server 2003 and later resources in the domain, and you can't go back to any other mode. You should use Windows Server 2003 mode only when you're certain that you don't need Windows NT domain structures, backup domain controllers (BDCs) running Windows NT, or Windows 2000 domain structures.

Using the Windows Server 2008 functional level

After you upgrade the Windows 2000 and Windows Server 2003 structures in your organization, you can change the functional level to Windows Server 2008 mode operations.

Before updating Windows Server 2003 domain controllers, you should prepare the domain for Windows Server 2008. To do this, you need to use Adprep.exe to update the forest and the domain schema so that they are compatible with Windows Server 2008 domains. Follow these steps:

1. On the schema operations master in the forest, copy the contents of the Sources\Adprep folder from the Windows Server 2008 installation media to a local folder, and then run **adprep /forestprep**. If you plan to install any read-only domain controllers, you should also run **adprep /rodcprep**. You need to use an administrator account that is a member of Enterprise Admins, Schema Admins, or Domain Admins in the forest root domain.
2. On the infrastructure operations master for each domain in the forest, copy the contents of the Sources\Adprep folder from the Windows Server 2008 installation media to a local folder, and then run **adprep /domainprep**. You need to use an account that is a member of the Domain Admins group in an applicable domain.
3. If you haven't previously run adprep /domainprep /gpprep in each domain, you need to manually perform this task. Server Manager for Windows Server 2012 R2 will not prepare Group Policy for you. Note that Group Policy needs to be prepared only the first time you deploy domain controllers running Windows Server 2003 SP1 or later. Running adprep /gpprep modifies the access control entries (ACEs) for all Group Policy Object (GPO) folders in the SYSVOL directory to grant read access to all enterprise domain controllers. This level of access is required to support Resultant Set of Policy (RSoP) for site-based policy and causes the NT File Replication Service (NTFRS) to resend all GPOs to all domain controllers.

As always, you should test any procedure in a lab before performing it in a production environment.

NOTE **To determine which server is the current schema operations master for the domain, start a command prompt and enter dsquery server -hasfsmo schema. To determine which server is the current infrastructure operations master for the domain, start a command prompt and enter dsquery server -hasfsmo infr.**

After upgrading all domain controllers to Windows Server 2008, you can raise the domain and forest level functionality to take advantage of additional Active Directory features. If you do this, you can use only Windows Server 2008 or later resources in the domain and you can't go back to any other mode. You should use Windows Server 2008 mode only when you're certain that you don't need Windows NT domain structures, Windows NT BDCs, or Windows 2000 or Windows Server 2003 domain structures.

Using the Windows Server 2008 R2 functional level

Windows Server 2008 R2 and later run only on 64-bit hardware. You'll likely need to install Windows Server 2008 R2 and later on new hardware rather than hardware designed for earlier releases of Windows Server.

Before updating domain controllers running Windows Server 2008, you should prepare the domain for Windows Server 2008 R2. To do this, you need to use Adprep.exe to update the forest and the domain schema so that they are compatible with Windows Server 2008 R2 domains. Follow these steps:

1. On the schema operations master in the forest, copy the contents of the Support\Adprep folder from the Windows Server 2008 R2 installation media to a local folder, and then run **adprep /forestprep**. If you plan to install any read-only domain controllers, you should also run **adprep /rodcprep**. You need to use an administrator account that is a member of Enterprise Admins, Schema Admins, or Domain Admins in the forest root domain.
2. On the infrastructure operations master for each domain in the forest, copy the contents of the Support\Adprep folder from the Windows Server 2008 R2 installation media to a local folder, and then run **adprep /domainprep**. You need to use an account that is a member of the Domain Admins group in an applicable domain.

As always, you should test any procedure in a lab before performing it in a production environment.

NOTE **To determine which server is the current schema operations master for the domain, start a command prompt and enter dsquery server -hasfsmo schema. To determine which server is the current infrastructure operations master for the domain, start a command prompt and enter dsquery server -hasfsmo infr.**

After upgrading all domain controllers to Windows Server 2008 R2, you can raise the domain and forest level functionality to take advantage of the latest Active Directory features. If you do this, you can use only Windows Server 2008 R2 resources in the domain. You should use Windows Server 2008 R2 mode only when you're certain that you don't need old Windows NT domain structures; Windows NT BDCs; or Windows 2000, Windows Server 2003, or Windows Server 2008 domain structures.

Using the Windows Server 2012 functional level

Windows Server 2012 runs only on 64-bit hardware. Unlike earlier releases of Windows Server, the domain and forest preparations required for updating Active Directory schema don't need to be performed manually. Instead, when you use Server Manager for Windows Server 2012, and the forest functional level is Windows Server 2003 or higher, any necessary preparations are done automatically when you deploy a domain controller running Windows Server 2012. This means the Configuration Wizard automatically updates forest and domain schema.

You also have the option of manually preparing for Windows Server 2012. To do this, you can use Adprep.exe to update the forest and the domain schema so that they are compatible with Windows Server 2012 domains. The steps are similar to those discussed in the previous section.

After upgrading all domain controllers to Windows Server 2012, you can raise the domain and forest level functionality to take advantage of the latest Active Directory features. If you do this, you can use only Windows Server 2012 resources in the domain.

Using the Windows Server 2012 R2 functional level

Windows Server 2012 R2 runs only on 64-bit hardware. As with Windows Server 2012 RTM, the domain and forest preparations required for updating Active Directory schema don't need to be performed manually. Instead, when you use Server Manager for Windows Server 2012 R2, and the forest functional level is Windows Server 2003 or higher, any necessary preparations are done automatically when you deploy a domain controller running Windows Server 2012 R2. This means the Configuration Wizard automatically updates forest and domain schema.

As with Windows Server 2012 RTM, you also have the option of manually preparing for Windows Server 2012 R2. After upgrading all domain controllers to Windows Server 2012 R2, you can raise the domain and forest level functionality to take advantage of the latest Active Directory features. If you do this, you can use only Windows Server 2012 R2 resources in the domain.

Raising or lowering domain and forest functionality

Domains operating in a Windows Server 2003 or higher functional level can use universal groups, group nesting, group type conversion, update logon time stamps, and Kerberos KDC key version numbers. In this mode or higher, administrators can do the following:

- Rename domain controllers without having to demote them first.
- Rename domains running on Windows Server 2003 or higher domain controllers.
- Create extended two-way trusts between two forests.
- Restructure domains in the domain hierarchy by renaming them and putting them at different levels.
- Take advantage of replication enhancements for individual group members and global catalogs.

As compared to earlier implementations, forests operating in a Windows Server 2003 or higher functional level have better global catalog replication and intrasite and intersite replication efficiency, in addition to the ability to establish one-way, two-way, and transitive forest trusts.

REAL WORLD **The domain and forest upgrade process can generate a lot of network traffic while information is being replicated around the network. Sometimes the entire upgrade process can take 15 minutes or longer. During this time, you might experience delayed responsiveness when communicating with servers and higher latency on the network, so you might want to schedule the upgrade outside normal business hours. It's also a good idea to thoroughly test compatibility with existing applications (especially legacy applications) before performing this operation.**

You can raise the domain level functionality by following these steps:

1. Open Active Directory Domains And Trusts. In the console tree, press and hold or right-click the domain you want to work with, and then tap or click Raise Domain Functional Level. The current domain name and functional level are displayed in the Raise Domain Functional Level dialog box.
2. To change the domain functionality, select the new domain functional level from the list provided, and then tap or click Raise.
3. Tap or click OK. The new domain functional level is replicated to each domain controller in the domain. This operation can take some time in a large organization.

You can raise the forest level functionality by following these steps:

1. Open Active Directory Domains And Trusts. In the console tree, press and hold or right-click the Active Directory Domains And Trusts node, and then tap or click Raise Forest Functional Level. The current forest name and functional level are displayed in the Raise Forest Functional Level dialog box.
2. To change the forest functionality, select the new forest functional level by using the list provided, and then tap or click Raise.
3. Tap or click OK. The new forest functional level is replicated to each domain controller in each domain in the forest. This operation can take some time in a large organization.

Another way to raise domain or forest functional level is to use Active Directory Administrative Center. This tool is available as an option on the Tools menu in Server Manager. Follow these steps to raise the domain functional level:

1. In Active Directory Administrative Center, the local domain is opened for management by default. If you want to work with a different domain, tap or click Manage, and then tap or click Add Navigation Nodes. In the Add Navigation Nodes dialog box, select the domain you want to work with, and then tap or click OK.
2. Select the domain you want to work with by tapping or clicking it in the left pane. In the Tasks pane, tap or click Raise The Domain Functional Level. The current domain name and functional level are displayed in the Raise Domain Functional Level dialog box.
3. To change the domain functionality, select the new domain functional level by using the list provided, and then tap or click Raise.
4. Tap or click OK. The new domain functional level is replicated to each domain controller in the domain. This operation can take some time in a large organization.

Follow these steps to raise the forest functional level:

1. In Active Directory Administrative Center, select the domain you want to work with by tapping or clicking it in the left pane. In the Tasks pane, tap or click Raise The Forest Functional Level. The current forest name and functional level are displayed in the Raise Forest Functional Level dialog box.

2. To change the forest functionality, select the new forest functional level by using the list provided, and then tap or click Raise.
3. Tap or click OK. The new forest functional level is replicated to each domain controller in each domain in the forest. This operation can take some time in a large organization.

Generally, you cannot lower the forest or domain functional level after you raise it. However, there are specific exceptions as discussed previously in this chapter. Keep in mind that if you enabled the Active Directory Recycle Bin, you won't be able to lower the forest functional level.

Understanding the directory structure

Active Directory has many components and is built on many technologies. Directory data is made available to users and computers through data stores and global catalogs. Although most Active Directory tasks affect the data store, global catalogs are equally important because they're used during logon and for information searches. In fact, if the global catalog is unavailable, standard users can't log on to the domain. The only way to change this behavior is to cache universal group membership locally. As you might expect, caching universal group membership has advantages and disadvantages, which I'll discuss in a moment.

You access and distribute Active Directory data by using directory access protocols and replication. *Directory access protocols* allow clients to communicate with computers running Active Directory. *Replication* is necessary to ensure that updates to data are distributed to domain controllers. Although multimaster replication is the primary technique you use to distribute updates, some changes to data can be handled only by individual domain controllers called *operations masters*. A feature of Windows Server 2008 or later called *application directory partitions* also changes the way multimaster replication works.

With application directory partitions, enterprise administrators (those belonging to the Enterprise Admins group) can create replication partitions in the domain forest. These partitions are logical structures used to control the replication of data within a domain forest. For example, you could create a partition to strictly control the replication of DNS information within a domain, thereby preventing other systems in the domain from replicating DNS information.

An application directory partition can appear as a child of a domain, a child of another application partition, or a new tree in the domain forest. Replicas of the application directory partition can be made available on any Active Directory domain controller running Windows Server 2008 or later, including global catalog servers. Although application directory partitions are useful in large domains and forests, they add overhead in terms of planning, administration, and maintenance.

Exploring the data store

The data store contains information about objects, such as accounts, shared resources, OUs, and group policies. Another name for the data store is the *directory*, which refers to Active Directory itself.

Domain controllers store the directory in a file called Ntds.dit. This file's location is set when Active Directory is installed, and it should be on an NTFS file system drive formatted for use with Windows Server 2008 or later. You can also save directory data separately from the main data store. This is true for group policies, scripts, and other types of public information stored on the shared system volume (SYSVOL).

Sharing directory information is called *publishing*. For example, you publish information about a printer by sharing the printer over the network. Similarly, you publish information about a folder by sharing the folder over the network.

Domain controllers replicate most changes to the data store in multimaster fashion. Administrators for small or medium-size organizations rarely need to manage replication of the data store. Replication is handled automatically, but you can customize it to meet the needs of large organizations or organizations with special requirements.

Not all directory data is replicated. Instead, only public information that falls into one of the following three categories is replicated:

- **Domain data** Contains information about objects within a domain. This includes objects for accounts, shared resources, organizational units, and group policies.
- **Configuration data** Describes the directory's topology. This includes a list of all domains, domain trees, and forests, and also the locations of the domain controllers and global catalog servers.
- **Schema data** Describes all objects and data types that can be stored in the directory. The default schema provided with Windows Server describes account objects, shared resource objects, and more. You can extend the default schema by defining new objects and attributes or by adding attributes to existing objects.

Exploring global catalogs

When universal group membership isn't cached locally, global catalogs enable network logon by providing universal group membership information when a logon process is initiated. Global catalogs also enable directory searches throughout the domains in a forest. A domain controller designated as a global catalog stores a full replica of all objects in the directory for its host domain and a partial replica for all other domains in the domain forest.

> ***NOTE*** **Partial replicas are used because only certain object properties are needed for logon and search operations. Partial replication also means that less information needs to be circulated on the network, reducing the amount of network traffic.**

By default, the first domain controller installed on a domain is designated as the global catalog. If only one domain controller is in the domain, the domain controller and the global catalog are the same server. Otherwise, the global catalog is on the domain controller you've configured as such. You can also add global catalogs to a domain to help improve response time for logon and search requests. The recommended technique is to have one global catalog per site within a domain.

Domain controllers hosting the global catalog should be well connected to domain controllers acting as infrastructure masters. The role of infrastructure master is one of the five operations master roles you can assign to a domain controller. In a domain, the infrastructure master is responsible for updating object references. The infrastructure master does this by comparing its data with that of a global catalog. If the infrastructure master finds outdated data, it requests updated data from a global catalog. The infrastructure master then replicates the changes to the other domain controllers in the domain. For more information about operations master roles, see the "Understanding operations master roles" section later in this chapter.

When only one domain controller is in a domain, you can assign the infrastructure master role and the global catalog to the same domain controller. When two or more domain controllers are in the domain, however, the global catalog and the infrastructure master must be on separate domain controllers. If they aren't, the infrastructure master won't find out-of-date data and will never replicate changes. The only exception is when all domain controllers in the domain host the global catalog. In this case, it doesn't matter which domain controller serves as the infrastructure master.

One of the key reasons to configure additional global catalogs in a domain is to ensure that a catalog is available to service logon and directory search requests. Again, if the domain has only one global catalog and the catalog isn't available, and there's no local caching of universal group membership, standard users can't log on, and those who are logged on can't search the directory. In this scenario, the only users who can log on to the domain when the global catalog is unavailable are members of the Domain Admins group.

Searches in the global catalog are very efficient. The catalog contains information about objects in all domains in the forest. This allows directory search requests to be resolved in a local domain rather than in a domain in another part of the network. Resolving queries locally reduces the network load and allows for quicker responses in most cases.

TIP **If you notice slow logon or query response times, you might want to configure additional global catalogs. But more global catalogs usually means more replication data being transferred over the network.**

Universal group membership caching

In a large organization, having global catalogs at every office location might not be practical. Not having global catalogs at every office location presents a problem, however, if a remote office loses connectivity with the main office or a designated

branch office where global catalog servers reside. If this occurs, standard users won't be able to log on; only members of Domain Admins will be able to log on. This happens because logon requests must be routed over the network to a global catalog server at a different office, and this isn't possible with no connectivity.

As you might expect, you can resolve this problem in many ways. You can make one of the domain controllers at the remote office a global catalog server by following the procedure discussed in the "Configuring global catalogs" section in Chapter 8. The disadvantage of this approach is that the designated server or servers will have an additional burden placed on them and might require additional resources. You also have to manage more carefully the up time of the global catalog server.

Another way to resolve this problem is to cache universal group membership locally. Here, any domain controller can resolve logon requests locally without having to go through a global catalog server. This allows for faster logons and makes managing server outages much easier because your domain isn't relying on a single server or a group of servers for logons. This solution also reduces replication traffic. Instead of replicating the entire global catalog periodically over the network, only the universal group membership information in the cache is refreshed. By default, a refresh occurs every eight hours on each domain controller that's caching membership locally.

Universal group membership caching is site-specific. Remember, a site is a physical directory structure consisting of one or more subnets with a specific IP address range and network mask. The domain controllers running Windows Server and the global catalog they're contacting must be in the same site. If you have multiple sites, you need to configure local caching in each site. Additionally, users in the site must be part of a Windows domain running in a Windows Server 2003 or higher functional mode. To learn how to configure caching, see the "Configuring universal group membership caching" section in Chapter 8.

Replication and Active Directory

Regardless of whether you use FRS or DFS replication, the three types of information stored in the directory are domain data, schema data, and configuration data.

Domain data is replicated to all domain controllers within a particular domain. Schema and configuration data are replicated to all domains in the domain tree or forest. In addition, all objects in an individual domain and a subset of object properties in the domain forest are replicated to global catalogs.

This means that domain controllers store and replicate the following:

- Schema information for the domain tree or forest
- Configuration information for all domains in the domain tree or forest
- All directory objects and properties for their respective domains

However, domain controllers hosting a global catalog store and replicate schema information for the forest and configuration information for all domains in the forest. They also store and replicate a subset of the properties for all directory

objects in the forest that's replicated only between servers hosting global catalogs and all directory objects and properties for their respective domain:

- Schema information for the forest
- Configuration information for all domains in the forest
- A subset of the properties between global catalogs
- All directory objects and properties for their domain

To get a better understanding of replication, consider the following scenario, in which you're installing a new network:

1. Start by installing the first domain controller in domain A. The server is the only domain controller and also hosts the global catalog. No replication occurs because no other domain controllers are on the network.
2. Install a second domain controller in domain A. Because there are now two domain controllers, replication begins. To make sure that data is replicated properly, assign one domain controller as the infrastructure master and the other as the global catalog. The infrastructure master watches for updates to the global catalog and requests updates to changed objects. The two domain controllers also replicate schema and configuration data.
3. Install a third domain controller in domain A. This server isn't a global catalog. The infrastructure master watches for updates to the global catalog, requests updates to changed objects, and then replicates those changes to the third domain controller. The three domain controllers also replicate schema and configuration data.
4. Install a new domain, domain B, and add domain controllers to it. The global catalog hosts in domain A and domain B begin replicating all schema and configuration data in addition to a subset of the domain data in each domain. Replication within domain A continues as previously described. Replication within domain B begins.

Active Directory and LDAP

The *Lightweight Directory Access Protocol (LDAP*) is a standard Internet communications protocol for TCP/IP networks. LDAP is designed specifically for accessing directory services with the least amount of overhead. LDAP also defines operations that can be used to query and modify directory information.

Active Directory clients use LDAP to communicate with computers running Active Directory whenever they log on to the network or search for shared resources. You can also use LDAP to manage Active Directory.

LDAP is an open standard that many other directory services use. This makes interdirectory communications easier and provides a clearer migration path from other directory services to Active Directory. You can also use Active Directory Service Interfaces (ADSI) to enhance interoperability. ADSI supports the standard application programming interfaces (APIs) for LDAP that are specified in Internet standard Request for Comments (RFC) 1823. You can use ADSI with Windows Script Host to create and manage objects in Active Directory.

Understanding operations master roles

Operations master roles accomplish tasks that are impractical to perform in multimaster fashion. Five operations master roles are defined, and you can assign these roles to one or more domain controllers. Although certain roles can be assigned only once in a domain forest, other roles must be defined once in each domain.

Every Active Directory forest must have the following roles:

- **Schema master** Controls updates and modifications to directory schema. To update directory schema, you must have access to the schema master. To determine which server is the current schema master for the domain, start a command prompt and enter **dsquery server -hasfsmo schema**.
- **Domain naming master** Controls the addition or removal of domains in the forest. To add or remove domains, you must have access to the domain naming master. To determine which server is the current domain naming master for the domain, start a command prompt and enter **dsquery server -hasfsmo name**.

These forestwide roles must be unique in the forest. This means that you can assign only one schema master and one domain naming master in a forest.

Every Active Directory domain must have the following roles:

- **Relative ID master** Allocates relative IDs to domain controllers. Whenever you create a user, group, or computer object, domain controllers assign a unique security ID to the related object. The security ID consists of the domain's security ID prefix and a unique relative ID allocated by the relative ID (RID) master. To determine which server is the current relative ID master for the domain, start a command prompt and enter **dsquery server -hasfsmo rid**.
- **PDC emulator** When you use mixed-mode or interim-mode operations, the PDC emulator acts as a Windows NT PDC. Its job is to authenticate Windows NT logons, process password changes, and replicate updates to BDCs. The PDC emulator is the default time server and, as such, also performs time synchronization in a domain. To determine which server is the current PDC emulator for the domain, start a command prompt and enter **dsquery server -hasfsmo pdc**.
- **Infrastructure master** Updates object references by comparing its directory data with that of a global catalog. If the data is outdated, the infrastructure master requests updated data from a global catalog and then replicates the changes to the other domain controllers in the domain. To determine which server is the current infrastructure operations master for the domain, start a command prompt and enter **dsquery server -hasfsmo infr**.

These domainwide roles must be unique in each domain. This means that you can assign only one relative ID master, one PDC emulator, and one infrastructure master in each domain.

Operations master roles are usually assigned automatically, but you can reassign them. When you install a new network, the first domain controller in the first domain is assigned all the operations master roles. If you later create a child domain or a

root domain in a new tree, the first domain controller in the new domain is automatically assigned operations master roles, too. In a new domain forest, the domain controller is assigned all operations master roles. If the new domain is in the same forest, the assigned roles are relative ID master, PDC emulator, and infrastructure master. The schema master and domain naming master roles remain in the first domain in the forest.

When a domain has only one domain controller, that computer handles all the operations master roles. If you're working with a single site, the default operations master locations should be sufficient. As you add domain controllers and domains, however, you'll probably want to move the operations master roles to other domain controllers.

When a domain has two or more domain controllers, you should configure two domain controllers to handle operations master roles. In this case, you make one domain controller the operations master and designate the second as your standby operations master. The standby operations master can then be used if the primary one fails. Be sure that the domain controllers are direct replication partners and are well connected.

As the domain structure grows, you might want to split up the operations master roles and place them on separate domain controllers. This can improve the responsiveness of the operations masters. Pay particular attention to the current responsibilities of the domain controller you plan to use.

BEST PRACTICES **Two roles you should not separate are schema master and domain naming master. Always assign these roles to the same server. For the most efficient operations, you also usually want the RID master and PDC emulator to be on the same server. But you can separate these roles if necessary. For example, on a large network where peak loads are causing performance problems, you probably want to place the RID master and PDC emulator on separate domain controllers. Additionally, you usually shouldn't place the infrastructure master on a domain controller hosting a global catalog. See the "Exploring global catalogs" section earlier in this chapter for details.**

Using the Active Directory Recycle Bin

When your Active Directory forest is operating in the Windows Server 2008 R2 or higher mode, you can use the Active Directory Recycle Bin. The Active Directory Recycle Bin adds an easy-to-use recovery feature for Active Directory objects. When you enable this feature, all link-valued and nonlink-valued attributes of a deleted object are preserved, allowing you to restore the object to the same state it was in before it was deleted. You can also recover objects from the recycle bin without having to initiate an authoritative restore. This differs substantially from the previously available technique, which used an authoritative restore to recover deleted objects from the Deleted Objects container. Previously, when you deleted an object, most of its non-link-valued attributes were cleared and all of its link-valued attributes were removed, which meant that although you could recover a deleted object, the object was not restored to its previous state.

Preparing schema for the Recycle Bin

Before you can make the Recycle Bin available, you must update Active Directory schema with the required Recycle Bin attributes. You do this by preparing the forest and domain for the Windows Server 2008 R2 functional level or higher. When you do this, the schema is updated, and then every object in the forest is also updated with the Recycle Bin attributes. This process is irreversible after it is started.

After you prepare Active Directory, you need to upgrade all domain controllers in your Active Directory forest to Windows Server 2008 R2 or higher and then raise the domain and forest functional levels to the Windows Server 2008 R2 level or higher. Optionally, you can update Active Directory schema in your forests and domains for Windows Server 2012 or later to enable the enhanced Recycle Bin.

After these operations, you can enable and access the Recycle Bin. When Recycle Bin has been enabled, it cannot be disabled. Now when an Active Directory object is deleted, the object is put in a state referred to as *logically deleted* and moved to the Deleted Objects container, shown in Figure 7-7. Also, its distinguished name is altered. A deleted object remains in the Deleted Objects container for the period of time set in the deleted object lifetime value, which is 180 days by default.

REAL WORLD **The msDS-deletedObjectLifetime attribute replaces the tombstone-Lifetime attribute. However, when msDS-deletedObjectLifetime is set to $null, the lifetime value comes from the tombstoneLifetime. If the tombstoneLifetime is also set to $null, the default value is 180 days.**

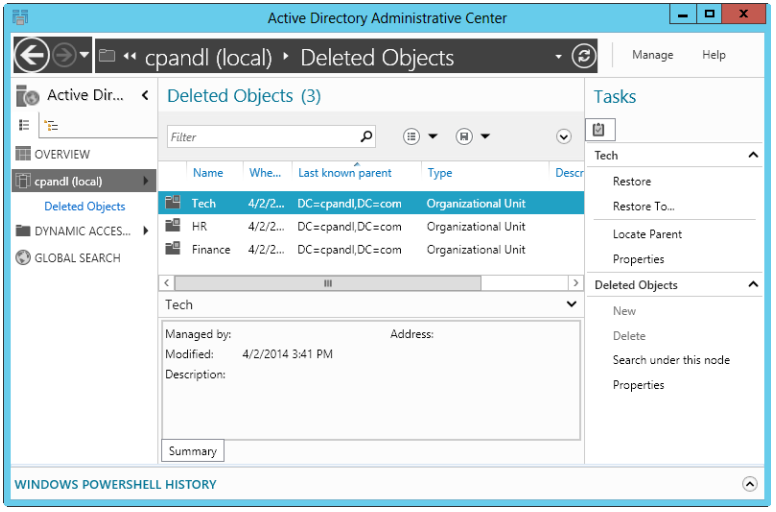


FIGURE 7-7 Deleted objects remain in the Deleted Objects container for the deleted object lifetime value.

Recovering deleted objects

If you elect not to use the Recycle Bin, you can still recover deleted objects from the Deleted Objects container by using an authoritative restore and other techniques I'll discuss in this section. The procedure has not changed from previous releases of Windows Server. What has changed, however, is that the objects are restored to their previous state with all link-valued and non-link-valued attributes preserved. To perform an authoritative restore, the domain controller must be in Directory Services Restore Mode.

Rather than using an authoritative restore and taking a domain controller offline, you can recover deleted objects by using the Ldp.exe administration tool or the Active Directory cmdlets for Windows PowerShell. If you updated the Active Directory schema in your forests and domains for Windows Server 2012 or later, you also can enable the enhanced Recycle Bin, which allows you to recover deleted objects by using Active Directory Administrative Center.

Keep in mind that Active Directory blocks access to an object for a short while after the object is deleted. During this time, Active Directory processes the object's link-value table to maintain referential integrity on the linked attribute's values. Active Directory then permits access to the deleted object.

Using Ldp.exe for basic recovery

You can use Ldp.exe to display the Deleted Objects container and recover a deleted object by following these steps:

1. Type **Ldp.exe** in the Everywhere search box, and then press Enter.
2. On the Options menu, tap or click Controls. In the Controls dialog box, select Return Deleted Objects in the Load Predefined list, and then tap or click OK.
3. Bind to the server that hosts the forest root domain by choosing Bind from the Connection menu. Select the Bind type, and then tap or click OK.
4. On the View menu, tap or click Tree. In the Tree View dialog box, use the BaseDN list to select the appropriate forest root domain name, such as DC=Cpandl,DC=Com, and then tap or click OK.
5. In the console tree, double-tap or double-click the root distinguished name, and then locate the CN=Deleted Objects container.
6. Locate and press and hold or right-click the Active Directory object you want to restore, and then tap or click Modify. This displays the Modify dialog box.
7. In the Edit Entry Attribute box, type **isDeleted**. Do not enter anything in the Values box.
8. Under Operation, tap or click Delete, and then tap or click Enter.
9. In the Edit Entry Attribute box, type **distinguishedName**. In Values, type the original distinguished name of this Active Directory object.
10. Under Operation, tap or click Replace. Select the Extended check box, tap or click Enter, and then tap or click Run.

Using Windows PowerShell for basic and advanced recovery

The Active Directory cmdlets for Windows PowerShell allow you to recover deleted objects by using scripts or by entering commands at a Windows PowerShell prompt. You use Get-ADObject to retrieve the object or objects you want to restore, pass that object or objects to Restore-ADObject, and then Restore-ADObject restores the object or objects to the directory database.

To use the Active Directory cmdlets for recovery, you need to open an elevated, administrator Windows PowerShell prompt by pressing and holding or right-clicking the Windows PowerShell entry on the menu, and then tapping or clicking Run As Administrator. The basic syntax for recovering an object is as follows:

```
Get-ADObject -Filter {ObjectId} -IncludeDeletedObjects | Restore-ADObject
```

ObjectId is a filter value that identifies the object you want to restore. For example, you could restore a deleted user account by display name or SAM account name, as shown in these examples:

```
Get-ADObject -Filter {DisplayName -eq "Rich Tuppy"}
-IncludeDeletedObjects | Restore-ADObject

Get-ADObject -Filter {SamAccountName -eq "richt"} -IncludeDeletedObjects
| Restore-ADObject
```

Note that nested objects must be recovered from the highest level of the deleted hierarchy to a live parent container. For example, if you accidentally deleted an OU and all its related accounts, you need to restore the OU before you can restore the related accounts.

The basic syntax for restoring container objects such as an OU is as follows:

```
Get-ADObject -ldapFilter:"(msDS-LastKnownRDN=ContainerID)"
-IncludeDeletedObjects | Restore-ADObject
```

ContainerID is a filter value that identifies the container object you want to restore. For example, you could restore the Corporate Services OU, as shown in this example:

```
Get-ADObject -ldapFilter:"(msDS-LastKnownRDN=Corporate_Services)"
-IncludeDeletedObjects | Restore-ADObject
```

If the OU contains accounts you also want to restore, you can now restore the accounts by using the technique discussed previously, or you can restore all accounts at the same time. The basic syntax requires that you establish a search base and associate the accounts with their last known parent, as shown here:

```
Get-ADObject -SearchBase "CN=Deleted Objects,ForestRootDN" -Filter
{lastKnownParent -eq "ContainerCN,ForestRootDN"} -IncludeDeletedObjects |
Restore-ADObject
```

ForestRootDN is the distinguished name of the forest root domain, such as DC=Cpandl,DC=Com, and *ContainerCN* is the common name of the container,

such as OU=Corporate_Services or CN=Users. The following example restores all the accounts that were in the Corporate Services OU when it was deleted:

```
Get-ADObject -SearchBase "CN=Deleted Objects,DC=Cpandl,DC=com" –Filter
{lastKnownParent -eq "OU=Corporate_Services,DC=Cpandl,DC=com"}
-IncludeDeletedObjects | Restore-ADObject
```

Using the enhanced Recycle Bin for recovery

The enhanced Recycle Bin makes recovering deleted objects as easy as pointing and clicking or tapping and holding. After you updated the Active Directory schema in your forests and domains for Windows Server 2012 or later, you enable the enhanced Recycle Bin for use by following these steps:

1. In Active Directory Administrative Center, the local domain is opened for management by default. If you want to work with a different domain, tap or click Manage, and then tap or click Add Navigation Nodes. In the Add Navigation Nodes dialog box, select the domain you want to work with, and then tap or click OK.
2. Select the domain you want to work with by tapping or clicking it in the left pane. In the Tasks pane, tap or click Enable Recycle Bin, and then tap or click OK in the confirmation dialog box.
3. Active Directory will begin replicating the change to all domain controllers in the forest. After the change is replicated, the enhanced Recycle Bin will be available for use. If you then tap or click Refresh in Active Directory Administrative Center, you'll notice that a Deleted Object container is now available for domains that use the enhanced Recycle Bin.

Keep in mind that the enhanced Recycle Bin is a forestwide option. When you enable this option in one domain of a forest, Active Directory replicates the change to all domain controllers in all domains of the forest.

With the enhanced Recycle Bin enabled, you can recover deleted objects with ease. In Active Directory Administrative Center, domains using the enhanced Recycle Bin will have a Deleted Object container. In this container, a list of deleted objects is displayed. As discussed previously, deleted objects remain in this container for the deleted object lifetime value, which is 180 days by default.

Each deleted object is listed by name, when it was deleted, the last known parent, and the type. When you select a deleted object by tapping or clicking it, you can use the options in the Tasks pane to work with it. The Restore option restores the object to its original container. For example, if the object was deleted from the Users container, it is restored to this container.

The Restore To option restores the object to an alternate container within its original domain or to a different domain within the current forest. Specify the alternate container in the Restore To dialog box. For example, if the object was deleted from the Users container in the tech.cpandl.com domain, you could restore it to the Devs OU in the eng.cpandl.com domain.

Index

Symbols and Numbers

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

About the author

WILLIAM STANEK (*www.williamstanek.com*) is the award-winning author and series editor of the bestselling Pocket Consultant series. William is one of the world's leading technology experts and has more than 20 years of hands-on experience with advanced programming and development. Over the years, his practical advice has helped millions of programmers, developers, and network engineers all over the world. Dubbed "A Face Behind the Future" in 1998 by *The Olympian*, William has been helping to shape the future of the written word for more than two decades. William's 150th book was published in 2013, and more than 7.5 million people have read his many works. William's current books include *Windows Server 2012 Inside Out* and the *Pocket Consultants* for Exchange Server 2013, Windows Server 2012 R2, and Windows 8.1.

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

William has an MS with distinction in information systems and a BS in computer science, magna cum laude. He is proud to have served in the Persian Gulf War as a combat crew member on an electronic warfare aircraft. He flew on numerous combat missions into Iraq and was awarded nine medals for his wartime service, including one of the United States of America's highest-flying honors, the Air Force Distinguished Flying Cross. Currently, he resides in the Pacific Northwest with his wife and children.

William recently rediscovered his love of the great outdoors. When he's not writing, he can be found hiking, biking, backpacking, traveling, or trekking in search of adventure with his family!

Find William on Twitter at WilliamStanek and on Facebook at *www.facebook.com/William.Stanek.Author*. Please visit *www.Pocket-Consultant.com* to find links to stay in touch with William.