Microsoft® Official Academic Course

Installing and Configuring Windows Server® 2012 R2 Exam 70-410

Craig Zacker

WILEY

Credits

VP & PUBLISHER	Don Fowley
EXECUTIVE EDITOR	John Kane
DIRECTOR OF SALES	Mitchell Beaton
EXECUTIVE MARKETING MANAGER	Chris Ruel
MICROSOFT PRODUCT MANAGER	Keith Loeber of Microsoft Learning
TECHNICAL EDITORS	Jeff T. Parker
	Kenneth Hess
	Brian Svidergol
EDITORIAL PROGRAM ASSISTANT	Allison Winkle
ASSISTANT MARKETING MANAGER	Debbie Martin
SENIOR PRODUCTION MANAGER	Janis Soo
CREATIVE DIRECTOR	Harry Nolan
COVER DESIGNER	Tom Nery
SENIOR PRODUCT DESIGNER	Thomas Kulesa
CONTENT EDITOR	Wendy Ashenberg
PRODUCTION EDITOR	Joyce Poh

This book was set in Garamond by Aptara, Inc. and printed and bound by Bind-Rite Robbinsville. The covers were printed by Bind-Rite Robbinsville.

Copyright © 2014 by John Wiley & Sons, Inc. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc. 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030-5774, (201) 748-6011, fax (201) 748-6008. To order books or for customer service, please call 1-800-CALL WILEY (225-5945).

Microsoft, Active Directory, AppLocker, Bing, BitLocker, DreamSpark, Hyper-V, Internet Explorer, SQL Server, Visual Studio, Win32, Windows Azure, Windows, Windows PowerShell, Windows Server, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other product and company names mentioned herein may be the trademarks of their respective owners.

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

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

ISBN 978-1-118-88231-3

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

www.wiley.com/college/microsoft *or*
call the MOAC Toll-Free Number: 1+(888) 764-7001 (U.S. & Canada only)

Preface

Welcome to the Microsoft Official Academic Course (MOAC) program for becoming a Microsoft Certified Solutions Associate for Windows Server 2012 R2. MOAC represents the collaboration between Microsoft Learning and John Wiley & Sons, Inc. Microsoft and Wiley teamed up to produce a series of textbooks that deliver compelling and innovative teaching solutions to instructors and superior learning experiences for students. Infused and informed by in-depth knowledge from the creators of Windows Server 2012 R2, and crafted by a publisher known worldwide for the pedagogical quality of its products, these textbooks maximize skills transfer in minimum time. Students are challenged to reach their potential by using their new technical skills as highly productive members of the workforce.

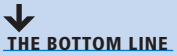
Because this knowledgebase comes directly from Microsoft, the architect of Windows Server 2012 R2 and creator of the Microsoft Certified Solutions Associate exams, you are sure to receive the topical coverage that is most relevant to students' personal and professional success. Microsoft's direct participation not only assures you that MOAC textbook content is accurate and current, it also means that students will receive the best instruction possible to enable their success on certification exams and in the workplace.

The Microsoft Official Academic Course Program

The Microsoft Official Academic Course series is a complete program for instructors and institutions to prepare and deliver great courses on Microsoft software technologies. With MOAC, we recognize that because of the rapid pace of change in the technology and curriculum developed by Microsoft, there is an ongoing set of needs beyond classroom instruction tools for an instructor to be ready to teach the course. The MOAC program endeavors to provide solutions for all these needs in a systematic manner in order to ensure a successful and rewarding course experience for both instructor and student, including technical and curriculum training for instructor readiness with new software releases; the software itself for student use at home for building hands-on skills, assessment, and validation of skill development; and a great set of tools for delivering instruction in the classroom and lab. All are important to the smooth delivery of an interesting course on Microsoft software, and all are provided with the MOAC program.

Conventions and Features Used in This Book

This book uses particular fonts, symbols, and heading conventions to highlight important information or to call your attention to special steps.

Convention	Meaning
	This feature provides a brief summary of the material to be covered in the section that follows.
	*Warning* points out instances when error or misuse could cause damage to the computer or network.
A ***shared printer*** can be used by many individuals on a network.	Key terms appear in bold italic.
`cd\windows\system32`	Commands that are to be typed are shown in a special font.
Click Install Now.	Any button on the screen you are supposed to click on or select will appear in blue.

Instructor Support Program

The Microsoft Official Academic Course programs are accompanied by a rich array of resources that incorporate the extensive textbook visuals to form a pedagogically cohesive package.

- The **Instructor's Guide** contains chapter summaries and lecture notes.
- The **Test Bank** contains hundreds of questions organized by lesson in multiple-choice, best answer, build list, and essay formats. A complete answer key is provided.
- **Lecture Presentation Slides.** A complete set of PowerPoint presentations is available to enhance classroom presentations. Tailored to the text's topical coverage, these presentations are designed to convey key Windows Server 2012 R2 concepts addressed in the text.
- **MOAC Labs Online.** MOAC Labs Online is a cloud-based environment that enables students to conduct exercises using real Microsoft products. These are not simulations but instead are live virtual machines in which faculty and students can perform activities as they would on a local machine. MOAC Labs Online relieves the need for lab setup, configuration, and most troubleshooting tasks. This represents an opportunity to lower costs, eliminate the hassle of lab setup, and support and improve student access and portability. Contact your Wiley rep about including MOAC Labs Online with your course offering.

About the Author

Craig Zacker is an instructor, writer, editor, and networker whose computing experience began in the days of teletypes and paper tape. After making the move from minicomputers to PCs, he worked as a network administrator and PC support technician while operating a freelance desktop publishing business. After earning a Master's Degree in English and American Literature from New York University, Craig worked extensively on the integration of Microsoft Windows operating systems into existing internetworks, supported fleets of Windows workstations, and was employed as a technical writer, content provider, and webmaster for the online services group of a large software company. Since devoting himself to writing and editing full-time, Craig has authored or contributed to dozens of books on operating systems, networking topics, and PC hardware. He has also published articles with top industry publications, developed online training courses for the various firms, and authored the following Microsoft Official Academic Course (MOAC), Academic Learning Series (ALS), and Self-Paced Training Kit titles:

MOAC: Installing and Configuring Windows Server 2012 (Exam 70-410)

MOAC: Windows Server 2008, Enterprise Administrator (Exam 70-647)

MOAC: Windows 7 Configuration (Exam 70-680)

MOAC: Windows Server Administrator (Exam 70-646)

MOAC: Configuring Windows Server 2008 Application Services (Exam 70-643)

MOAC: Configuring Microsoft Windows Vista (Exam 70-620)

MOAC: Implementing & Administering Security in a Windows Server 2003 Network (Exam 70-299)

MOAC: Managing & Maintaining a Microsoft Windows Server 2003 Environment (Exam 70-290)

ALS: Network+ Certification, Second, Third, and Fourth Editions

ALS: Planning & Maintaining a Windows Server 2003 Network Infrastructure (Exam 70-293)

ALS: Microsoft Windows 2000 Network Infrastructure Administration, Second Edition (2002)

MCSE Self-Paced Training Kit (Exam 70-293): Planning & Maintaining a Microsoft Windows Server 2003 Network Infrastructure (2003)

MCSA/MCSE Self-Paced Training Kit: Microsoft Windows 2000 Network Infrastructure Administration, Exam 70-216, Second Edition (2002)

MCSA Training Kit: Managing a Windows 2000 Network Environment (2002)

Network+ Certification Training Kit, First and Second Editions (2001)

Network+ Certification Readiness Review (2001)

Brief Contents

Contents

Installing Servers

LESSON 1

■ Selecting a Windows Server 2012 R2 Edition

THE BOTTOM LINE

Microsoft releases all its operating systems in multiple editions, which provides consumers with various price points and feature sets.

When planning a server deployment, you should choose the operating system edition based on multiple factors, including the following:

- The roles you intend the servers to perform
- The virtualization strategy you intent to implement
- The licensing strategy you plan to use

Compared with Windows Server 2008, Microsoft has simplified the process of selecting a Windows Server 2012 R2 edition by reducing the available products. As with Windows Server 2008 R2, Windows Server 2012 R2 requires a 64-bit processor architecture. All 32-bit versions have been eliminated, and there is no build supporting Itanium processors. This leaves Windows Server 2012 R2 with the following core editions:

- **Windows Server 2012 R2 Datacenter:** This edition is designed for large and powerful servers with up to 64 processors and fault-tolerance features such as hot add processor support. As a result, this edition is available only through the Microsoft volume-licensing program and from original equipment manufacturers (OEMs), bundled with a server.
- **Windows Server 2012 R2 Standard:** This edition includes the full set of Windows Server 2012 features, varying from the Datacenter edition only by the number of virtual machine instances permitted by the license.
- **Windows Server 2012 R2 Essentials:** This edition includes nearly all the features in the Standard and Datacenter editions, except for Server Core, Hyper-V, and Active Directory Federation Services. This edition is limited to one physical or virtual server instance and a maximum of 25 users.
- **Windows Server 2012 R2 Foundation:** This reduced version of the operating system is designed for small businesses that require only basic server features such as file and print services and application support. This edition includes no virtualization rights and is limited to 15 users.

These various editions are priced commensurate with their capabilities. Obviously, your goal is to purchase the most inexpensive edition that provides all your needs. The following sections examine the primary differences between the Windows Server 2012 R2 editions.

Supporting Server Roles

Windows Server 2012 R2 includes predefined combinations of services called ***roles*** that implement common server functions.

Computers running the Windows Server 2012 R2 operating system can perform a wide variety of tasks, using both the software included with the product and third-party applications. The activities Windows Server 2012 R2 performs for network clients are known as roles. After you install the Windows Server 2012 R2 operating system, you can use Server Manager or ***Windows PowerShell*** to assign one or more roles to that computer.

The roles included with Windows Server 2012 R2 fall into three basic categories:

- **Directory services** store, organize, and supply information about a network and its resources.
- **Infrastructure services** provide support services for network clients.
- **Application services** provide communications services, operating environments, or programming interfaces for specific applications.

Table 1-1 lists the roles that Microsoft supplies with Windows Server 2012 R2.

Table 1-1

Windows Server 2012 R2 Server Roles

Directory Services	Infrastructure Services	Application Services
Active Directory Certificate Services implements certification authorities (CAs) and other services that facilitate the creation and management of the public key certificates used by the identity and access control elements of the Windows Server 2012 R2 security infrastructure.	**DHCP (Dynamic Host Configuration Protocol) Server** provides network clients with dynamically assigned IP addresses and other TCP/IP configuration settings, such as subnet masks, default gateway addresses, and Domain Name System (DNS) server addresses.	**Application Server** provides an integrated environment for deploying and running server-based business applications designed within (or expressly for) the organization, such as those requiring the services provided by Internet Information Services (IIS), Microsoft .NET Framework 2.0 and 3.0, COM+, ASP .NET, Message Queuing, or Windows Communication Foundation (WCF).
Active Directory Domain Services (AD DS) configure the server to function as an Active Directory domain controller, which stores and manages a distributed database of network resources and application-specific information.	**DNS Server** provides name-to-address and address-to-name resolution services for AD DS and Internet clients. The Windows Server 2012 R2 DNS server implementation also supports dynamic DNS and DHCP integration.	**Fax Server** enables you to manage fax devices and clients to send and receive faxes over the network.
Active Directory Federation Services create a single sign-on environment by implementing trust relationships that enable users on one network to access applications on other networks without providing a secondary set of logon credentials.	**Hyper-V** provides a hypervisor-based environment in which administrators can create virtual machines, each of which provides an isolated instance of the operating system environment.	**File and Storage Services** install tools and services that enhance Windows Server 2012 R2's basic ability to provide network clients with access to files stored on server drives, including Distributed File System (DFS), DFS Replication, Storage Manager for Storage Area Networks (SANs), fast file searching, and file services for UNIX clients.

(continued)

Table 1-1

(continued)

Directory Services	Infrastructure Services	Application Services
Active Directory Lightweight Directory Services (AD LDS) implement a Lightweight Directory Access Protocol (LDAP) directory service that provides support for directory-enabled applications without incurring the extensive overhead of AD DS.	**Network Policy and Access Services (NPAS)** implement services such as Network Policy Server (NPS), Health Registration Authority (HRA), and Host Credential Authorization Protocol (HCAP), which enforce security policies for network users.	**Print and Document Services** provides clients with access to printers attached to the server or to the network, as well as centralized network printer and print server management, and printer deployment using Group Policy. Document services enable you to route images from network-attached scanners to users.
Active Directory Rights Management Services (AD RMS) make up a client/server system that uses certificates and licensing to implement persistent usage policies, which can control access to information, no matter where a user moves it.	**Remote Access** provides remote users with access to network resources by using DirectAccess and VPNs, as well as LAN and NAT routing services.	**Remote Desktop Services** enable clients on the network or on the Internet to access server-based applications remotely or the entire Windows desktop by using server resources.
	Volume Activation Services automate the management of Microsoft host keys and Key Management System (KMS) hosts.	**Web Server (IIS)** installs Internet Information Services (IIS) 7.5, which enables the organization to publish websites and web-based applications for use by intranet, extranet, and/or Internet clients.
	Windows Deployment Services (WDS) enable you to install Windows operating systems remotely on computers throughout the enterprise.	
	Windows Server Essentials Experience enables Standard and Datacenter servers to run Essentials services, such as Remote Web Access.	
	Windows Server Update Services (WSUS) automate the process of disseminating operating-system updates to Windows computers throughout the enterprise.	

Some Windows Server 2012 R2 editions include all these roles, whereas others include only some of them. Selecting the appropriate edition of Windows Server has always been a matter of anticipating the roles that the computer must perform. At one time, this was a relatively simple process. You planned your server deployments by deciding which ones would be domain controllers, which ones would be web servers, and so forth. After you made these decisions, you were done, because server roles were largely static.

With the increased focus on virtualization in Windows Server 2012 R2, however, more administrators must consider not only what roles servers must perform at the time of the deployment, but also what roles they will perform in the future.

By using virtualized servers, you can modify your network's server strategy at will to accommodate changing workloads and business requirements, or to adapt to unforeseen circumstances. Therefore, the process of anticipating the roles servers will perform must account for the potential expansion of your business, as well as possible emergency needs.

■ Installing Windows Server 2012 R2

THE BOTTOM LINE

A clean installation is the simplest way to deploy Windows Server 2012 R2 on a bare metal computer—that is, a computer with no operating system installed—or a computer with a partition that you are willing to reformat (losing all the data on the partition in the process).

If a computer is brand new and has no operating system installed on it, it cannot start until you supply a boot disk, such as a Windows Server 2012 R2 installation disk. During installation, you select the disk partition on which you want to install the operating system, and the Setup program copies the operating system files there.

System Requirements

Choosing the correct hardware for a server requires an understanding of the tasks it will perform.

As of this writing, the minimum system requirements for all editions of Windows Server 2012 R2 are as follows:

- 1.4 GHz 64-bit processor
- 512 MB RAM
- 32 GB disk space
- DVD or USB flash drive
- Super VGA (1024x768) or higher resolution monitor

Having 32 GB of available disk space should be considered an absolute minimum. The system partition needs extra space if you install the system over a network or your computer has more than 16 GB of RAM installed. The additional disk space is required for paging, hibernation, and dump files. In practice, you are unlikely to come across a computer with 32 GB RAM and only 32 GB disk space. If you do, free more disk space or invest in additional storage hardware.

Not until you have decided how you will deploy your applications and what roles an application server will perform should you begin selecting the hardware that goes into the computer. Suppose that your organization decides to deploy an application suite such as Microsoft Office on all company workstations. If you decide to install the applications on each individual workstation, each computer must have sufficient memory and processor speed to run them efficiently. The application servers on the network then has to perform only relatively simple roles, such as file and print services, which do not require enormous amounts of server resources.

By contrast, if you decide to deploy the applications using Remote Desktop Services, you can use workstations with a minimal hardware configuration, because the servers take most of the burden. In this case, you need a more powerful application server in terms of processor and memory, or perhaps even several servers sharing the client load.

Server roles can also dictate requirements for specific subsystems within the server computers, as in the following examples:

- Servers hosting complex applications might require more memory and faster processors.
- File servers can benefit from disk arrays and hard drives with higher speeds and larger caches, or even a high performance drive interface, such as SCSI (Small Computer System Interface, pronounced "scuzzy").
- Web servers receiving large amounts of traffic might need higher-end network adapters or multiple adapters to connect to different subnets.
- Streaming media servers require sufficient hardware in all subsystems, because any performance bottleneck in the server can interrupt the client's media experience.

Enterprises with extensive server requirements might want to consider specialized server hardware, such as a storage area network, network attached storage, or a server cluster.

As part of Microsoft's increased emphasis on virtualization and cloud computing in its server products, the company has increased the maximum hardware configurations significantly for Windows Server 2012 R2. Table 1-2 lists these maximums.

Table 1-2

Maximum Hardware Configurations in Windows Server Versions

	Windows Server 2012 R2	Windows Server 2008 R2
Logical Processors	640	256
RAM	4 terabytes	2 terabytes
Failover cluster nodes	64	16

Performing a Clean Installation

A clean installation can be the basis for a new server, or the initial phase of a server migration.

To perform a clean installation of Windows Server 2012 R2, use the following procedure.

PERFORM A CLEAN INSTALLATION

GET READY. Prepare the computer for the Windows Server 2012 R2 installation by making sure that all its external peripheral devices are connected and powered on.

1. Turn on the computer and insert the Windows Server 2012 R2 installation disk into the DVD drive.
2. Press any key to boot from the DVD (if necessary). A progress indicator screen appears as Windows is loading files.

MORE INFORMATION

The device that a PC uses to boot is specified in its system (or BIOS) settings. In some cases, you might have to modify these settings to enable the computer to boot from the Windows Server 2012 R2 DVD. If you are not familiar with the operation of a particular computer, watch the screen carefully as the system starts and look for an instruction specifying what key to press to access the system settings.

The computer loads the Windows graphical interface and the Windows Setup page appears, as shown in Figure 1-1.

Figure 1-1

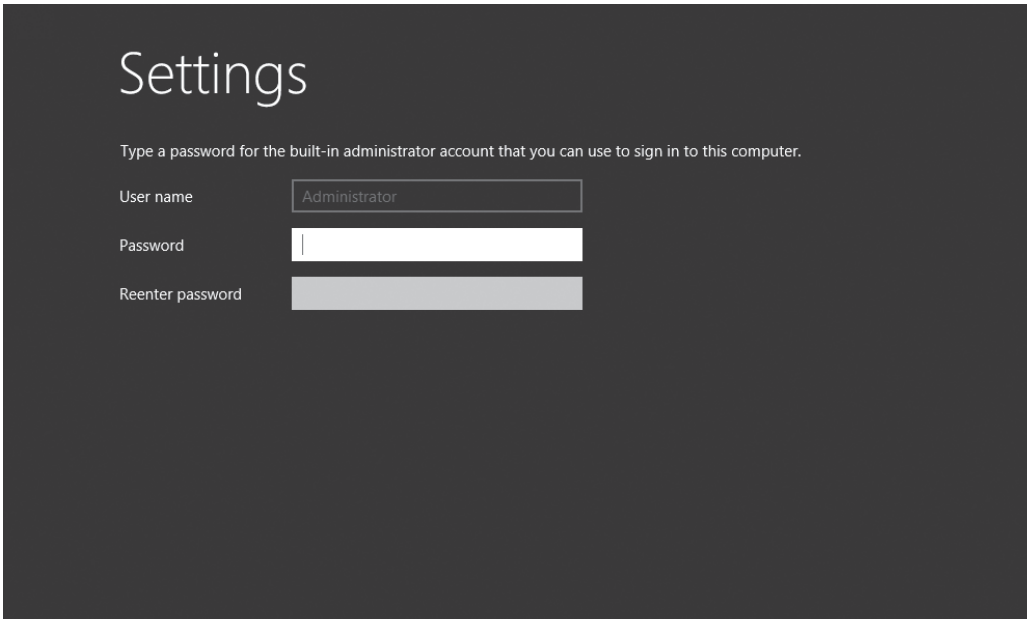
The Windows Setup page

3. By using the drop-down lists provided, select the appropriate language to install, time and currency format, and keyboard or input method, and then click Next. Another Windows Setup page appears.
4. Click Install Now. The Windows Setup Wizard appears, displaying the *Select the operating system you want to install* page.
5. Select the operating system edition and installation option you want to install and click Next. The *License Terms* page appears.
6. Select the I accept the license terms check box and click Next. The *Which type of installation do you want?* page, appears.
7. Because you are performing a clean installation and not an upgrade, click the Custom: Install Windows Only (advanced) option. The *Where do you want to install Windows?* page appears.
8. From the list provided, select the partition on which you want to install Windows Server 2012 R2, or select an area of unallocated disk space where the Setup program can create a new partition. Then click Next. The *Installing Windows* page appears.

9. After several minutes, during which the Setup program installs Windows Server 2012 R2, the computer restarts and the *Settings* page appears, as shown in Figure 1-2.

Figure 1-2

The Settings page



10. In the Password and Reenter Password text boxes, type the password to be associated with the Administrator account and press *Enter*. The system finalizes the installation and the Windows sign-on screen appears.

■ Choosing Installation Options

THE BOTTOM LINE

Many enterprise networks today use servers dedicated to a particular role. When a server is performing a single role, does it really make sense to have so many other processes running on the server that contribute little to that role?

Many IT administrators today are so accustomed to graphical user interfaces (GUIs) that they are unaware that there was ever any other way to operate a computer. When the first version of Windows NT Server appeared in 1993, many complained about wasting server resources on graphical displays and other elements that they deemed unnecessary. Up until that time, server displays were usually minimal, character-based, monochrome affairs. In fact, many servers had no display hardware at all, relying instead on text-based remote administration tools, such as Telnet.

Using Server Core

Windows Server 2012 R2 includes an installation option that addresses those old complaints about wasting server resources on graphical displays.

When you select the Windows ***Server Core*** installation option, you get a stripped-down version of the operating system. There is no Taskbar, no desktop Explorer shell, no Microsoft Management Console, and virtually no graphical applications. All you see when you start the computer is a single window with a command prompt.

The advantages of running servers using Server Core are several:

- **Hardware resource conservation:** Server Core eliminates some of the most memory- and processor-intensive elements of the Windows Server 2012 R2 operating system, thus devoting more of the system hardware to running essential services.
- **Reduced disk space:** Server Core requires less disk space for the installed operating system elements, as well as less swap space, which maximizes the utilization of the server's storage resources.
- **Reduced patch frequency:** Windows Server 2012 R2's graphical elements are among the most frequently patched features, so running Server Core reduces the number of patches that you must apply. Fewer patches also mean fewer server restarts and less downtime.
- **Reduced attack surface:** The less software there is running on the computer, the fewer entrances are available for attackers to exploit. Server Core reduces the potential openings presented by the operating system, increasing its overall security.

When Microsoft first introduced the Server Core installation option in Windows Server 2008, the idea was intriguing, but few administrators took advantage of it. The main reason for this was that most server administrators were not sufficiently conversant with the command-line interface to manage a Windows server without a GUI.

In Windows Server 2008 and Windows Server 2008 R2, the decision to install the operating system via the Server Core option was irrevocable. After you installed the operating system using Server Core, in no way could you get the GUI back except to perform a complete reinstallation. That has all changed in Windows Server 2012 and Windows Server 2012 R2. You can now switch a server from the Server Core option to the Server with a GUI option, and back again, at will, using Windows PowerShell commands.

This ability means that you can install Windows Server 2012 R2 using the Server with a GUI option, if you want to, configure the server using the familiar graphical tools, and then switch the server to Server Core, to take advantage of the benefits listed earlier.

SERVER CORE DEFAULTS

In Windows Server 2012 R2, Microsoft is attempting to fundamentally modify the way administrators work with their servers. Server Core is now the default installation option because in the new way of managing servers, you should rarely, if ever, have to work at the server console, either physically or remotely.

Windows Server has long been capable of remote administration, but this capability has been a piecemeal affair. Some Microsoft Management Console (MMC) snap-ins enabled administrators to connect to remote servers, and Windows PowerShell 2.0 provided some remote capabilities from the command line, but Windows Server 2012 R2, includes comprehensive remote administration tools that virtually eliminate the need to work at the server console.

The Server Manager application in Windows Server 2012 R2 enables you to add servers from all over the enterprise and create server groups to facilitate the configuration of multiple systems simultaneously. The new Windows PowerShell 4.0 environment increases the number of available commands—known as ***cmdlets***—from 230 to well over 2,000.

With tools like these, it is possible for you to install your servers using the Server Core option, execute a few commands to join each server to an AD DS domain, and then never touch the server console again. You can perform all subsequent administration tasks, including deployment of roles and features, by using Server Manager and Windows PowerShell from a remote workstation.

SERVER CORE CAPABILITIES

In addition to omitting most of the graphical interface, a Server Core installation omits some of the server roles found in a Server with a GUI installation. However, the Server Core option in Windows Server 2012 R2 includes 12 of the 19 roles, plus support for SQL Server 2012, as opposed to only 10 roles in Windows Server 2008 R2 and 9 in Windows Server 2008.

Table 1-3 lists the roles and features that are available and not available in a Windows Server 2012 R2 Server Core installation.

Table 1-3

Windows Server 2012 R2 Server Core Roles

Roles Available in Server Core Installation	Roles Not Available in Server Core Installation
Active Directory Certificate Services	Active Directory Federation Services
Active Directory Domain Services	Application Server
Active Directory Lightweight Directory Services	Fax Server
Active Directory Rights Management Services	Network Policy and Access Services
DHCP Server	Remote Desktop Services: • Remote Desktop Gateway • Remote Desktop Session Host • Remote Desktop Web Access
DNS Server	Volume Activation Services
File and Storage Services	Windows Deployment Services
Hyper-V	
Print and Document Services	
Remote Desktop Services: • Remote Desktop Connection Broker • Remote Desktop Licensing • Remote Desktop Virtualization Host	
Remote Access	
Web Server (IIS)	
Windows Server Update Services	

Using Features on Demand

During a Windows Server 2012 R2 installation, the Setup program copies the files for all operating system components from the installation medium to a directory called ***WinSxS,*** the side-by-side component store. This enables you to activate any features included with Windows Server 2012 R2 without having to supply an installation medium.

The drawback of this arrangement is that the WinSxS directory occupies a significant amount of disk space, much of which is, in many cases, devoted to data that will never be used.

With the increasing use of virtual machines to distribute server roles, enterprise networks often have more copies of the server operating system than ever before, and therefore more wasted disk space. Also, the advanced storage technologies often used by today's server infrastructures, such as storage area networks (SANs) and solid state drives (SSDs), are making that disk space more expensive.

Features on Demand, introduced in to Windows Server 2012, is a third state for operating system features that enables administrators to conserve disk space by removing specific features not only from operation, but also from the WinSxS directory.

This state is intended for features that you do not intend to install on a particular server. If, for example, you want to disable the Server Graphical Shell feature in Windows Server 2012 R2 to prevent Internet Explorer, Windows Explorer, and the desktop shell from running, and you want to remove the files that provide those features from the disk completely, you can do so with Features on Demand. By removing all the disk files for all your unused features on all your virtual machines, the accumulated savings in disk space can be substantial.

Features on Demand provide a third installation state for each feature in Windows Server 2012 R2. In versions of the operating system prior to Windows Server 2012, you could only enable or disable features. Windows Server 2012 R2 provides the following three states:

- Enabled
- Disabled
- Disabled with payload removed

To implement this third state, you must use the Windows PowerShell `Uninstall-WindowsFeature` cmdlet, which now supports a new `–Remove` flag. Thus, the Windows PowerShell command to disable the Server Graphical Shell and remove its source files from the WinSxS directory would be as follows:

```
Uninstall-WindowsFeature Server-Gui-Shell -Remove
```

Deleting the source files for a feature from the WinSxS folder does not make them irretrievably gone. If you try to enable that feature again, the system downloads it from Windows Update or, alternatively, retrieves it from an image file you specify using the `–Source` flag with the `Install-WindowsFeature` cmdlet. This enables you to retrieve the required files from a removable disk or from an image file on the local network. You can also use Group Policy to specify a list of installation sources.

TAKE NOTE* This ability to retrieve source files for a feature from another location is the actual functionality to which the name Features on Demand is referring. Microsoft often uses this capability to reduce the size of updates downloaded from the Internet. After the user installs the update, the program downloads the additional files required and completes the installation.

■ Upgrading Servers

THE BOTTOM LINE

An in-place upgrade is the most complicated form of Windows Server 2012 R2 installation. It is also the lengthiest and the most likely to cause problems during its execution. Whenever possible, Microsoft recommends that administrators perform a clean installation, or migrate required applications and settings instead.

During an in-place upgrade, the Setup program creates a new Windows folder and installs the Windows Server 2012 R2 operating system files into it. This is only half of the process, however. The program must then migrate the applications, files, and settings from the old OS. This calls for a variety of procedures, such as importing the user profiles, copying all pertinent settings from the old registry to the new one, locating applications and data files, and updating device drivers with new versions.

While in-place upgrades often proceed smoothly, the complexity of the upgrade process and the large number of variables involved means that many things can potentially go wrong. To minimize the risks involved, you must to take the upgrade process seriously, prepare the system beforehand, and have the ability to troubleshoot any problems that might arise. The following sections discuss these subjects in detail.

Upgrade Paths

Upgrade paths for Windows Server 2012 R2 are quite limited. In fact, they are easier to specify when you can perform an upgrade than when you cannot.

If you have a 64-bit computer running Windows Server 2008 or Windows Server 2008 R2, you can upgrade it to Windows Server 2012 R2 as long as you use the same (or a lower) operating system edition.

Windows Server 2012 R2 does not support the following:

- Upgrades from Windows Server versions prior to Windows Server 2008
- Upgrades from Windows workstation operating systems
- Cross-edition upgrades, such as Windows Server 2008 Standard Edition to Windows Server 2012 R2 Datacenter Edition
- Cross-platform upgrades, such as 32-bit Windows Server 2008 to 64-bit Windows Server 2012 R2
- Upgrades from any Itanium edition
- Cross-language upgrades, such as from Windows Server 2008, U.S. English, to Windows Server 2012 R2, French

In any of these cases, the Windows Setup program does not permit the upgrade to proceed.

Preparing to Upgrade

Before you begin an in-place upgrade to Windows Server 2012 R2, you should perform a number of preliminary procedures to ensure that the process goes smoothly and that server data is protected.

Consider the following before you perform any upgrade to Windows Server 2012 R2:

- **Check hardware compatibility.** Make sure that the server meets the minimum hardware requirements for Windows Server 2012 R2.
- **Check disk space.** Make sure that sufficient free disk space is on the partition where the old operating system is installed. During the upgrade procedure, sufficient disk space is needed to hold both operating systems simultaneously. After the upgrade is complete, you can remove the old files, freeing up some additional space.
- **Confirm that software is signed.** All kernel-mode software on the server, including device drivers, must be digitally signed, or the upgrade will not proceed. If you cannot locate a software update for any signed application or driver, you must uninstall the application or driver before you proceed with the installation.
- **Check application compatibility.** The Setup program displays a Compatibility Report page that can point out possible application compatibility problems. You can sometimes solve these problems by updating or upgrading the applications. Create an inventory of the software products installed on the server and check the manufacturers' websites for updates, availability of upgrades, and announcements regarding support for Windows Server 2012 R2. In an enterprise environment, you should test all applications for Windows Server 2012 R2 compatibility, no matter what the manufacturer says, before you perform any operating system upgrades.
- **Ensure computer functionality.** Make sure that Windows Server 2008 or Windows Server 2008 R2 is running properly on the computer before you begin the upgrade process. Check the Event Viewer console for warnings and errors. You must start an in-place upgrade from within the existing operating system, so you cannot count on Windows Server 2012 R2 to correct any problems that prevent the computer from starting or running the Setup program.
- **Perform a full backup.** Before you perform any upgrade procedure, you should back up the entire system, or at the very least the essential data files. Removable hard drives make this a simple process, even if the computer does not have a suitable backup device.
- **Purchase Windows Server 2012 R2.** Be sure to purchase the appropriate Windows Server 2012 R2 edition for the upgrade, and have the installation disk and product key handy.

Performing an Upgrade Installation

Windows Server 2012 R2 permits you to perform an upgrade installation only after you have met the prerequisites described in the previous section.

To perform a Windows Server 2012 R2 upgrade installation from Windows Server 2008 or Windows Server 2008 R2, use the following procedure.

PERFORM AN UPGRADE INSTALLATION

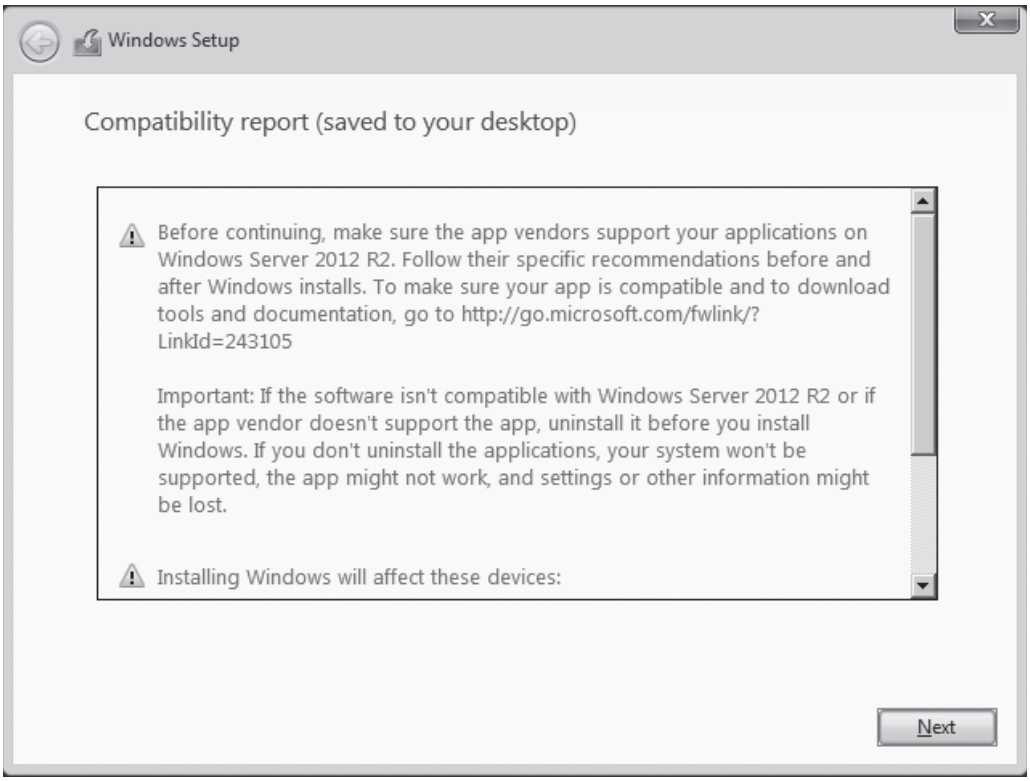
GET READY. Start the server and log on using an account with administrative privileges.

1. Insert the Windows Server 2012 R2 installation disk into the DVD drive and start the Setup program. The Windows Setup window appears.
2. Click Install Now. The *Get important updates for Windows Server* page appears.
3. Click No thanks. The Windows Setup Wizard appears, displaying the *Select the operating system you want to install* page.

4. Select the operating system edition and installation option you want to install and click Next. The *License Terms* page appears.
5. Select the I accept the license terms check box and click Next. *The Which type of installation do you want?* page appears.
6. Click the Upgrade: Install Windows and keep files, settings, and applications option. The *Compatibility report (saved to your desktop)* page appears, as shown in Figure 1-3.

Figure 1-3

The Compatibility Report page



7. Note the compatibility information provided by the Setup program and click Next. The *Upgrading Windows* page appears.

   After several minutes, during which the Setup program upgrades Windows Server 2008 or Windows Server 2008 R2 to Windows Server 2012 R2 and restarts the computer several times, the system finalizes the installation and the Windows sign-on screen appears.

During the upgrade process, when the system restarts, the boot menu provides an option to roll back to the previous operating system version. However, after the upgrade is complete, this option is no longer available; uninstalling Windows Server 2012 R2 and reverting to the old operating system version is not possible.

■ Migrating Roles

THE BOTTOM LINE

Migration is the preferred method of replacing an existing server with one running Windows Server 2012 R2. Unlike an in-place upgrade, a migration copies vital information from an existing server to a clean Windows Server 2012 R2 installation.

During a migration, virtually all the restrictions listed earlier concerning upgrades do not apply. By using the Windows Server Migration Tools and migration guides supplied with Windows Server 2012 R2, you can migrate data between servers under any of the following conditions:

- **Between versions:** You can migrate data from any Windows Server version since Windows Server 2003 SP2 to Windows Server 2012 R2. This includes migrations from one server running Windows Server 2012 R2 to another.
- **Between platforms:** You can migrate data from an x86- or x64-based server to an x64-based server running Windows Server 2012 R2.
- **Between editions:** You can migrate data between servers running different Windows Server editions.
- **Between physical and virtual instances:** You can migrate data from a physical server to a virtual one, or the reverse.
- **Between installation options:** You can migrate data from a server running Windows Server 2008 R2 to one running Windows Server 2012 R2, even when one server is using the Server Core installation option and the other uses the Server Core with a GUI option.

TAKE NOTE*

Windows Server 2012 R2 does not support migrations between different language versions of the operating system. You also cannot migrate data from Server Core installations of Windows Server 2008, because Server Code in that version does not include support for Microsoft .NET Framework.

Migration at the server level is different from any migrations you might have performed on workstation operating systems. Rather than perform a single migration procedure that copies all user data from the source to the destination computer at once, in a server migration you migrate roles or role services individually.

Windows Server 2012 R2 includes a collection of migration guides that provide individualized instructions for each role supported by Windows Server 2012 R2. Some roles require the use of the Windows Server Migration Tools; others do not.

Installing Windows Server Migration Tools

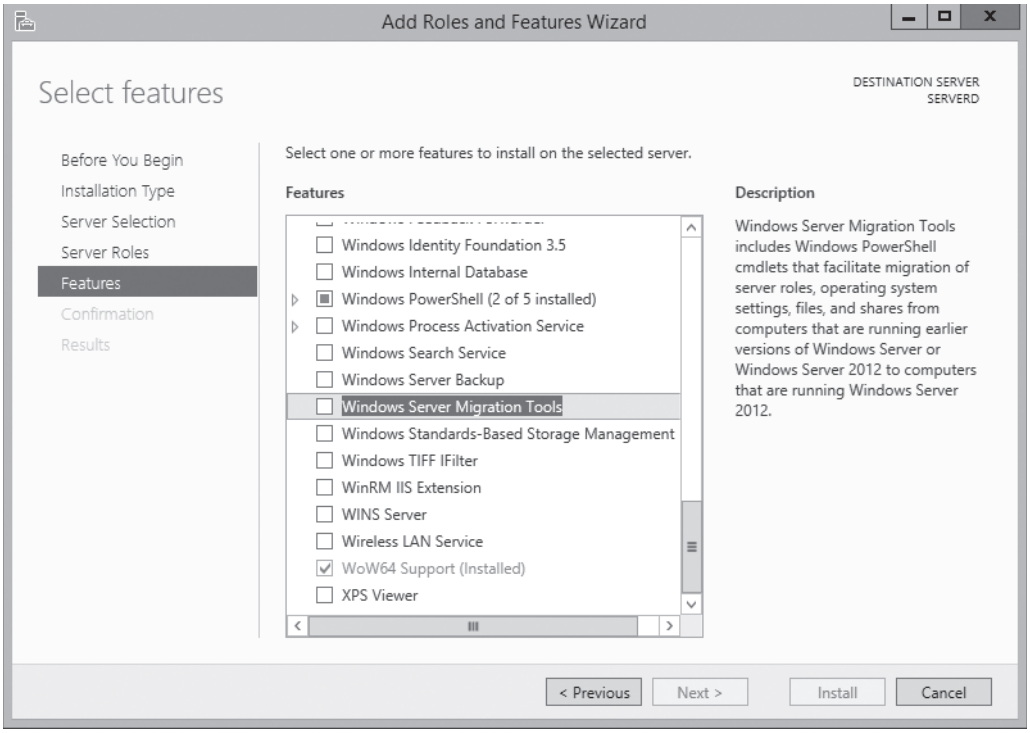
Windows Server Migration Tools is a Windows Server 2012 R2 feature that consists of Windows PowerShell cmdlets and help files that enable administrators to migrate certain roles between servers.

Before you can use the migration tools, however, you must install the Windows Server Migration Tools feature on the destination server running Windows Server 2012 R2, and then copy the appropriate version of the tools to the source server.

Windows Server Migration Tools is a standard feature that you install on Windows Server 2012 R2 using the Add Roles and Features Wizard in Server Manager, as shown in Figure 1-4, or the `Install-WindowsFeature` Windows PowerShell cmdlet.

Figure 1-4

The Select Features page of the Add Roles and Features Wizard



After you install the Windows Server Migration Tools feature on the destination server, you must create a distribution folder containing the tools for the source server. This distribution folder must contain the appropriate files for the platform and the operating system version of the source server.

To create the distribution folder on a server running Windows Server 2012 R2 with the Windows Server Migration Tools feature already installed, use the following procedure.

CREATE A WINDOWS SERVER MIGRATION TOOLS DISTRIBUTION FOLDER

GET READY. Start the destination server running Windows Server 2012 R2 and log on using an account with administrative privileges.

1. Open a Command Prompt window.
2. Switch to the directory containing the Windows Server Migration Tools files by typing the following command and pressing Enter:

   ```
   cd\windows\system32\ServerMigrationTools
   ```

3. Run the SmigDeploy.exe program with the appropriate command line switches for the platform and operating system version of the source server, using the following syntax:

   ```
   SmigDeploy.exe /package /architecture [x86|amd64] /os
   [WS08|WS08R2|WS03] /path <deployment_folder_path>
   ```

The SmigDeploy.exe program creates a new folder in the directory you specify for the *<deployment_folder_path>* variable, assigning it a name and location based on the command-line switches you specify. For example, if you enter the following command and press Enter, the program creates a folder called C:\SMT_ws08R2_amd64 containing the Server Migration Tools.

```
SmigDeploy.exe /package /architecture amd64 /os WS08R2 /path C:\
```

After you create the distribution folder, you must copy it to the source server by any standard means, and then register the Windows Server Migration Tools on the source server using the following procedure.

CREATE A WINDOWS SERVER MIGRATION TOOLS DISTRIBUTION FOLDER

GET READY. Start the source server and log on using an account with administrative privileges.

1. Open a Command Prompt window.
2. Switch to the folder containing the Windows Server Migration Tools that you previously copied to the server.
3. Run the SmigDeploy.exe program with no parameters on the command line, as follows:

   ```
   SmigDeploy.exe
   ```

When you execute SmigDeploy.exe, the program registers the Windows Server Migration Tools on the source server and opens a Windows PowerShell window in which you can use those tools.

To use the migration tools in a new Windows PowerShell session, you must open a Windows PowerShell window with elevated user rights and then add the appropriate snap-in, using the following syntax:

```
Add-PSSnapin Microsoft.Windows.ServerManager.Migration
```

Using Migration Guides

After you install the Windows Server Migration Tools on both the source and the destination servers, you can proceed to migrate data between the two.

By using the migration tools, you can migrate certain roles, features, shares, operating system settings, and other data from the source server to the destination server running Windows Server 2012 R2. Some roles require the use of the migration tools while others do not, having their own internal communication capabilities.

For example, the Print and Document Services role includes a Printer Migration Wizard (and a command-line tool called Printbrm.exe) that enables you to export printers on a source server to a file and import the file on the destination server. Roles that do not have capabilities like this rely on the Windows Server Migration Tools.

Migrating all the Windows Server roles does not involve any one procedure, whether the roles have their own migration tools or not. Instead, Microsoft provides detailed migration guides for individual roles, and sometimes for individual role services within a role.

■ Business Case Scenarios

Scenario 1-1: Preparing for an Upgrade to Windows Server 2012 R2

Walk through the steps an administrator needs to do to prepare for an upgrade to Windows Server 2012 R2.

Scenario 1-2: Switching to GUI Installation

A server is running the Server Core installation of Windows Server 2012 R2. What would you do if you desired the GUI installation?

2 LESSON

Configuring Servers

■ Completing Post-Installation Tasks

THE BOTTOM LINE

As part of the new emphasis on cloud-based services in Windows networking, Windows Server 2012 R2 contains various tools that have been overhauled to facilitate remote server management capabilities.

With the new Server Manager, for example, you can fully manage Windows servers without ever having to interact directly with the server console, either physically or remotely. However, immediately after the operating system installation, you might have to perform some tasks that require direct access to the server console. These tasks might include the following:

- Configuring the network connection
- Setting the time zone
- Renaming the computer
- Joining a domain
- Enabling Remote Desktop
- Configuring Windows Update settings
- Installing and configuring Windows PowerShell Desired State Configuration (DSC)

Using Command-Line Tools

If you selected the Server Core option when installing Windows Server 2012 R2, you can perform the same post-installation tasks from the command line.

At the very minimum, you need to rename the computer and join it to a domain. To perform these tasks, use the Netdom.exe command.

To rename a computer, run Netdom.exe with the following syntax:

```
netdom renamecomputer %ComputerName% /NewName: <NewComputerName>
```

To restart the computer as directed, use the following command:

```
shutdown /r
```

Then, to join the computer to a domain, use the following syntax:

```
netdom join %ComputerName% /domain:<DomainName>
/userd:<UserName> /passwordd:*
```

In this command, the asterisk (*) in the `/passwordd` parameter causes the program to prompt you for the password to the user account you specified.

These commands assume that a DHCP server has already configured the computer's TCP/IP client. If this is not the case, you must configure it manually before you can join a domain. To assign a static IP address to a computer using Server Core, you can use the Netsh.exe program or the New-NetIPAddress cmdlet provided by Windows PowerShell.

Converting Between GUI and Server Core

In Windows Server 2012 R2, you can convert a computer installed with the full GUI option to Server Core and add the full GUI to a Server Core computer.

This is a major improvement in the usefulness of Server Core over the version in Windows Server 2008 R2, in which you can change the interface only by reinstalling the entire operating system. With this capability, you can install servers with the full GUI, use the graphical tools to perform the initial setup, and then convert them to Server Core to conserve system resources. If later it becomes necessary, it is possible to reinstall the GUI components.

To convert a full GUI installation of Windows Server 2012 R2 to Server Core using Server Manager, use the following procedure.

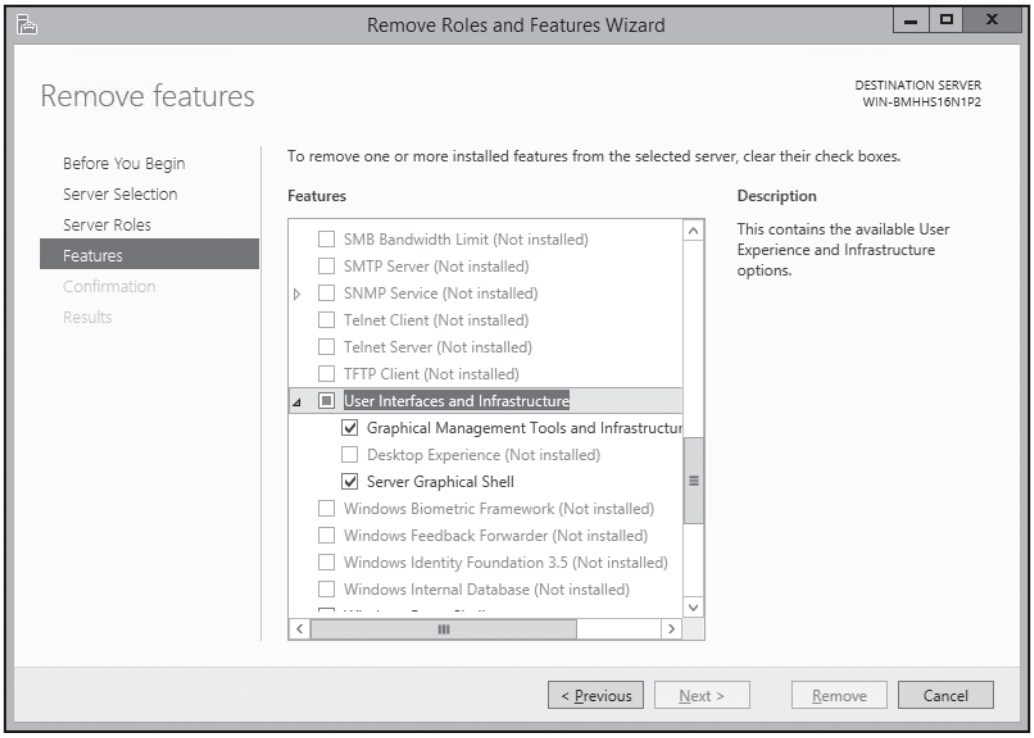
CONVERT A GUI SERVER TO SERVER CORE

GET READY. Log on to the server running Windows Server 2012 R2 by using an account with administrative privileges. The Server Manager window appears.

1. From the Manage menu, select Remove Roles and Features. The Remove Roles and Features Wizard appears, displaying the *Before you begin* page.
2. Click Next. The *Select destination server* page appears.
3. Select the server you want to convert to Server Core and click Next. The *Remove Server Roles* page appears.
4. Click Next. The *Remove features* page appears.
5. Scroll down in the list and expand the User Interfaces and Infrastructure feature, as shown in Figure 2-1.

Figure 2-1

The *Remove features* page in Server Manager



6. Clear the check boxes for the following components:
   - Graphical Management Tools and Infrastructure
   - Server Graphical Shell
7. The *Remove features that require Graphical Management Tools and Infrastructure* dialog box appears, with a list of dependent features that must be uninstalled. Click Remove Features.
8. Click Next. The *Confirm removal selections* page appears.
9. Select the Restart the destination server automatically if required check box and click Remove. The *Removal progress* page appears as the wizard uninstalls the feature.
10. Click Close. When the removal is completed, the computer restarts.

To add the full GUI to a Server Core computer, you must use Windows PowerShell to install the same features you removed in the previous procedure.

Configuring NIC Teaming

A new feature in Windows Server 2012 and Windows Server 2012 R2, NIC teaming enables administrators to combine the bandwidth of multiple network interface adapters, providing increased performance and fault tolerance.

Virtualization enables you to separate vital network functions on different systems without having to purchase a separate physical computer for each one. However, one drawback of this practice is that a single server hosting multiple virtual machines is still a single point of failure for all of them. A single malfunctioning network adapter, a faulty switch, or even an unplugged cable can bring down a host server and all its VMs with it.

NIC teaming—also called bonding, balancing, and aggregation—is a technology that has been available for some time, but was always tied to specific hardware implementations. The NIC teaming capability in Windows Server 2012 R2 is hardware independent and enables you to combine multiple physical network adapters into a single interface. The results can include increased performance by combining the throughput of the adapters and protection from adapter failures by dynamically moving all traffic to the functioning NICs.

NIC teaming in Windows Server 2012 R2 supports two modes:

- **Switch Independent Mode:** All network adapters are connected to different switches, providing alternative routes through the network.
- **Switch Dependent Mode:** All network adapters are connected to the same switch, providing a single interface with the adapters' combined bandwidth.

In Switch Independent Mode, you can choose between two configurations. The active/active configuration leaves all network adapters functional, providing increased throughout. If one adapter fails, all traffic shunts to the remaining adapters. In the active/standby configuration, one adapter is left offline, to function as a failover in the event the active adapter fails. In active/active mode, an adapter failure causes a performance reduction; in active/standby mode, the performance remains the same before and after an adapter failure.

In Switch Dependent Mode, you can choose static teaming, a generic mode that balances traffic between the adapters in the team, or you can opt to use the Link Aggregation Control Protocol defined in IEEE 802.3ax, assuming that your equipment supports it.

In Windows Server 2012, there was one significant limitation in NIC teaming. If your traffic consisted of large TCP sequences, such as a Hyper-V live migration, the system avoided using multiple adapters for those sequences to minimize the number of lost and out-of-order TCP segments. You therefore did not realize any performance increase for large file transfers using TCP. In Windows Server 2012 R2, a new Dynamic Mode splits these large TCP sequences into smaller units and sitributes them among the NICs in a team. This is now the default load balancing mode in Windows Server 2012 R2.

You can create and manage NIC teams using Server Manager or Windows PowerShell. To create a NIC team using Server Manager, use the following procedure.

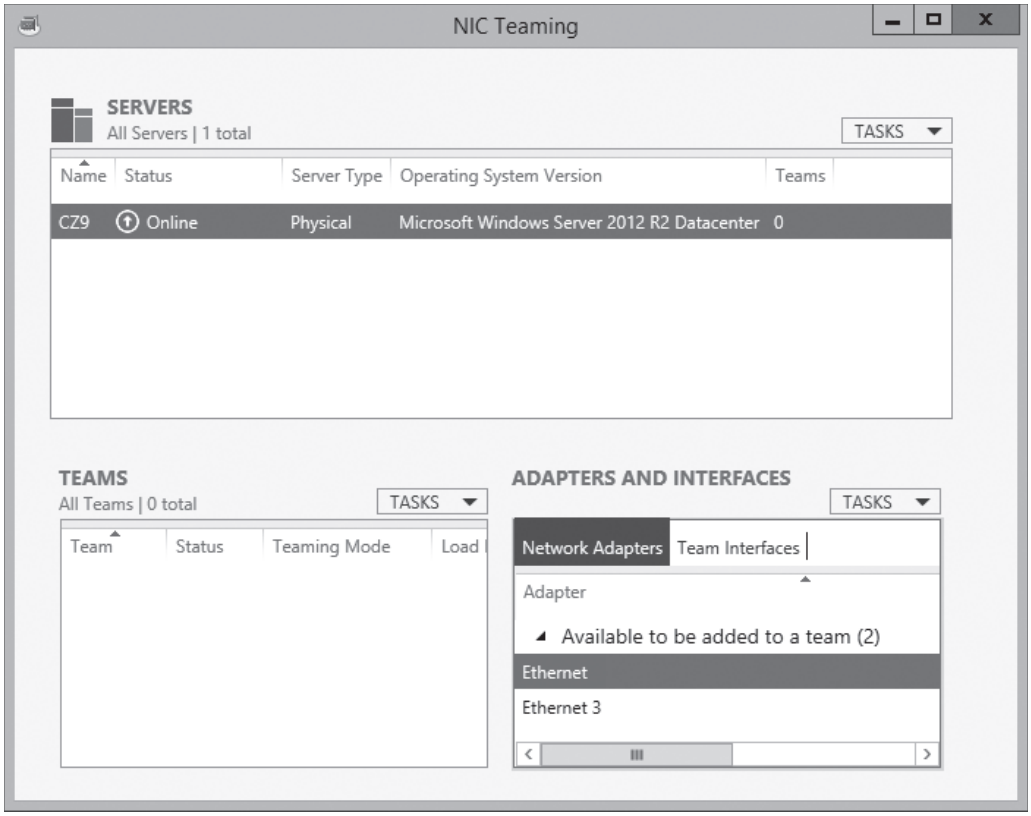
CREATE A NIC TEAM

GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges. The Server Manager window appears.

1. In the navigation pane, click the Local Server icon. The Local Server homepage appears.
2. In the Properties tile, click the *NIC Teaming* hyperlink. The *NIC Teaming* window appears, as shown in Figure 2-2.

Figure 2-2

The *NIC Teaming* window in Server Manager

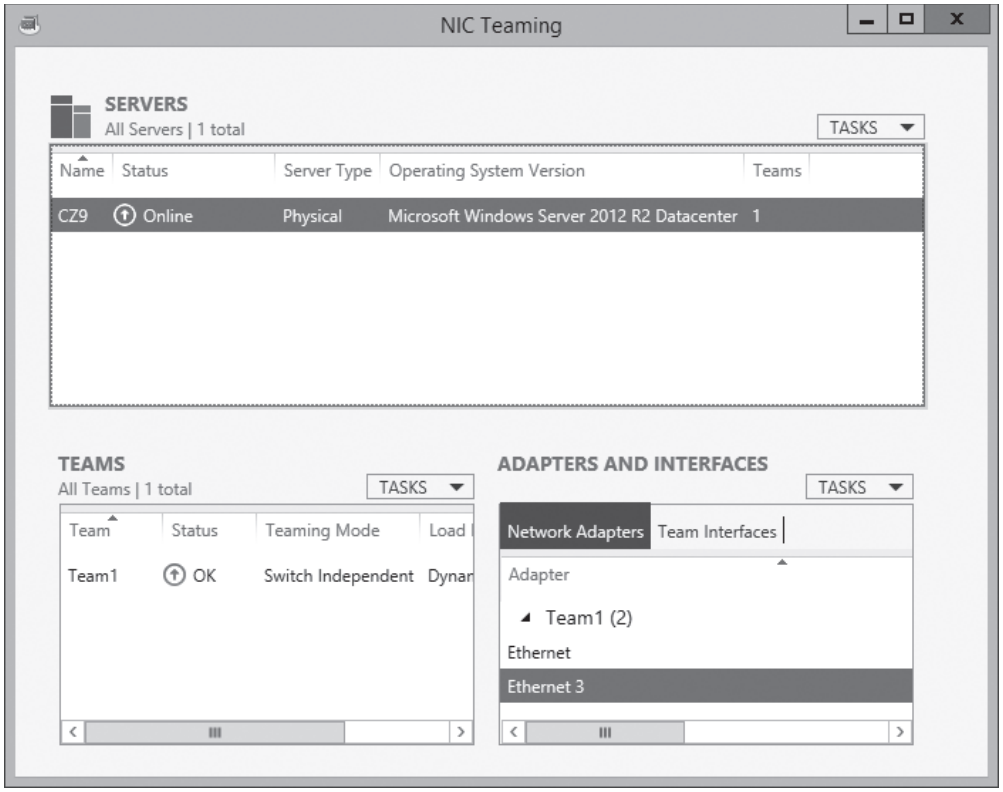


3. In the Teams tile, click the Tasks menu and select New Team. The *New team* page appears.
4. Click the Additional properties down arrow to expand the window.
5. In the Team Name text box, type the name you want to assign to the team.
6. In the Member adapters box, select the network adapters you want to add to the team.
7. In the Teaming Mode drop-down list, select one of the following options:
   - Static Teaming
   - Switch Independent
   - LACP
8. In the Load balancing mode drop-down list, select one of the following options:
   - Address Hash
   - Hyper-V Port
   - Dynamic
9. If you selected Switch Independent for the Teaming mode value, use the Standby adapter drop-down list to select one of adapters you added to the team to function as the offline standby.
10. Click OK. The new team appears in the Teams tile, as shown in Figure 2-3.

After you create a NIC team, you can use the NIC Teaming window to monitor the status of the team and the team interface you created. The team itself and the individual adapters all have status indicators that inform you if an adapter goes offline.

Figure 2-3

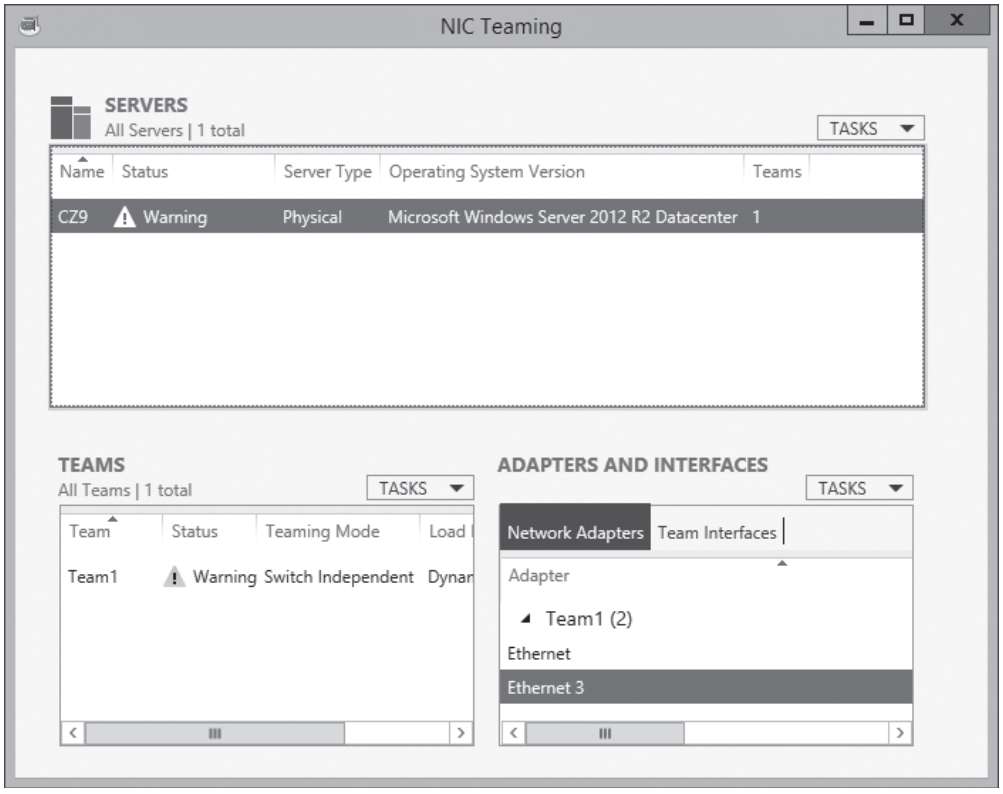
A new NIC team in the *NIC Teaming* window in Server Manager



If this does occur, the indicator for the faulty adapter immediately switches to disconnected, as shown in Figure 2-4, and depending on which teaming mode you chose, the status of the other adapter might change as well.

Figure 2-4

A NIC team with a failed adapter



Using Server Manager

THE BOTTOM LINE

The Server Manager tool in Windows Server 2012 R2 is a completely new application that is the first and most obvious evidence of a major paradigm shift in Windows Server administration.

In previous version of Windows Server, an administrator wanting to install a role using graphical controls had to work at the server console by either physically sitting at the keyboard or connecting to it using Remote Desktop Services (formerly Terminal Services). By contrast, the Windows Server 2012 R2 Server Manager can install roles and features to any server on the network, and even to multiple servers or groups of servers at once.

Adding Roles and Features

The Server Manager program in Windows Server 2012 R2 combines what used to be separate wizards for adding roles and features into one, the Add Roles and Features Wizard.

If you add multiple servers to the Server Manager interface, they are integrated into the Add Roles and Features Wizard, so you can deploy roles and features to any of your servers.

To install roles and features using Server Manager, use the following procedure.

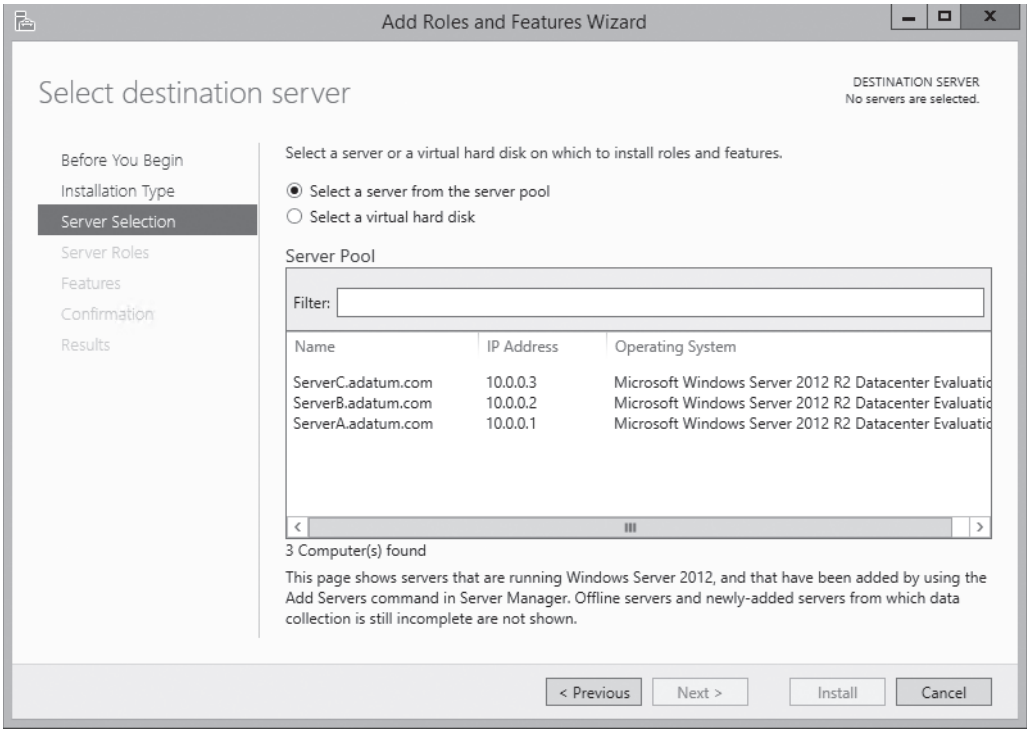
INSTALL ROLES AND FEATURES USING SERVER MANAGER

GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges. The Server Manager window appears.

1. From the Manage menu, select Add Roles and Features. The Add Roles and Features Wizard appears, displaying the *Before you begin* page.
2. Click Next. The *Select Installation Type* page appears.
3. Leave the *Role-based or feature-based installation* radio button selected and click Next. The *Select destination server* page appears, as shown in Figure 2-5.

Figure 2-5

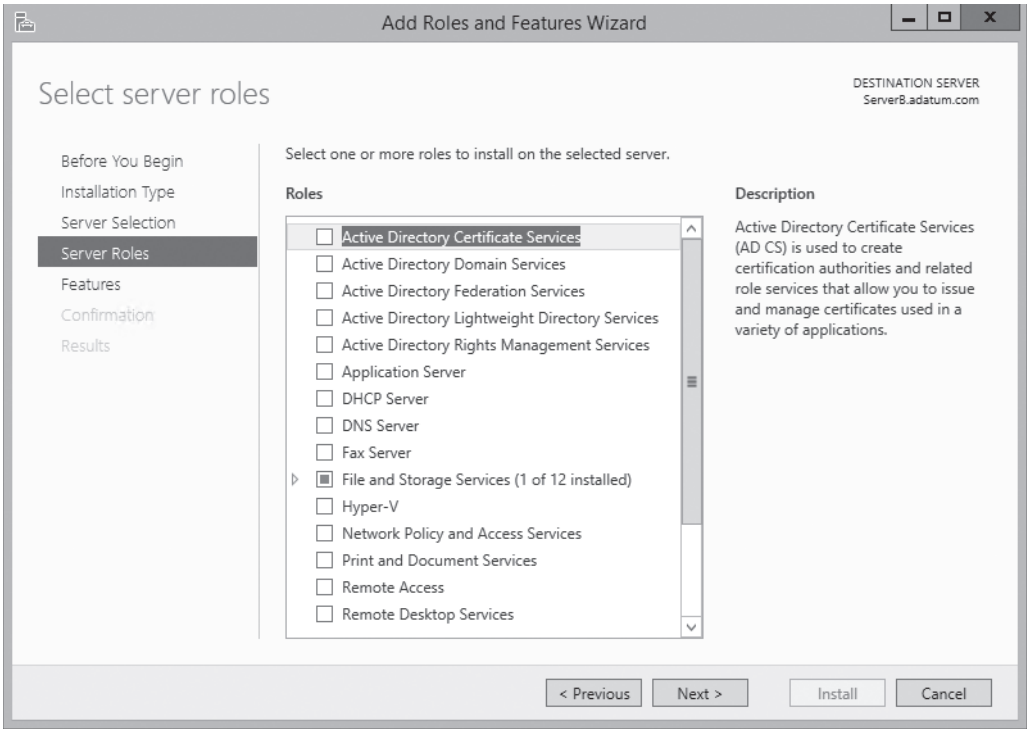
The *Select destination server* page in the Add Roles and Features Wizard



4. Select the server on which you want to install the roles and/or features. If the server pool contains a large number of servers, you can use the filter text box to display a subset of the pool based on a text string. After you select the server, click Next. The *Select Server Roles* page appears, as shown in Figure 2-6.

Figure 2-6

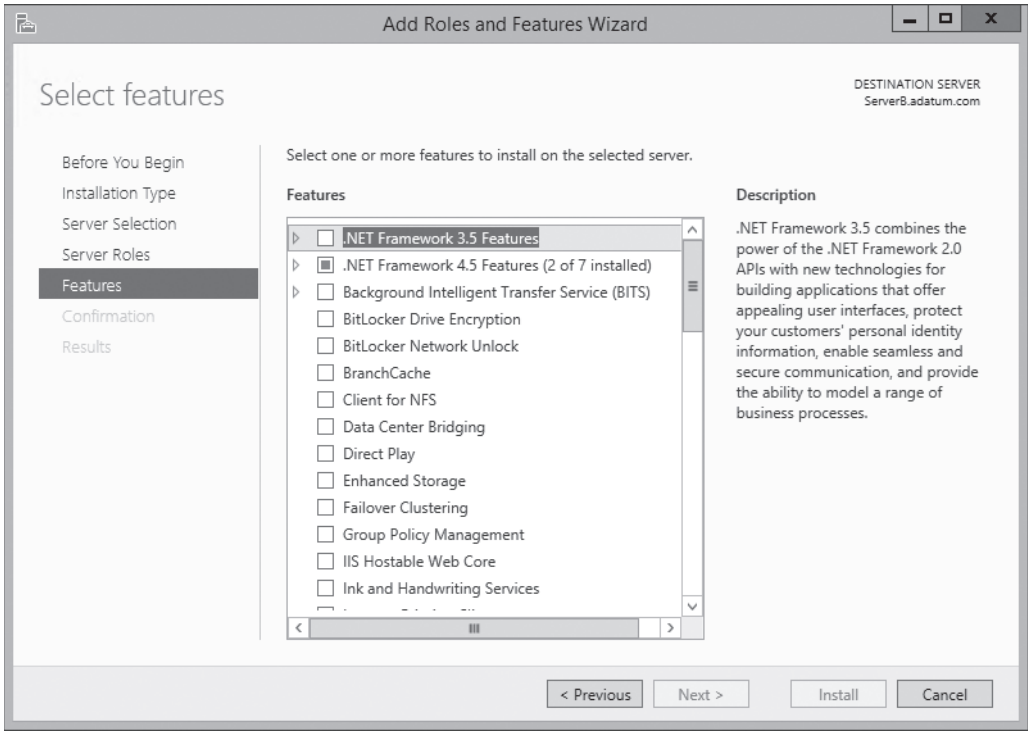
The *Select Server Roles* page in the Add Roles and Features Wizard



5. Select the role or roles you want to install on the selected server. If the roles you select have other roles or features as dependencies, an *Add features that are required* dialog box appears.
6. Click Add Features to accept the dependencies, and then click Next. The *Select features* page appears, as shown in Figure 2-7.

Figure 2-7

The *Select features* page in the Add Roles and Features Wizard



7. Select any features you want to install in the selected server and click Next. Dependencies also might appear for your feature selections.
8. The wizard displays pages specific to the roles and/or features you have chosen. Most roles have a *Select role services* page, on which you can select which elements of the role you want to install. Complete each of the role- or feature-specific pages and click Next. A *Confirm installation selections* page appears.
9. Select from the following optional functions, if desired:
   - **Restart the destination server automatically if desired** causes the server to restart automatically when the installation completes, if the selected roles and features require it.
   - **Export configuration settings** create an XML script documenting the procedures performed by the wizard, which you can use to install the same configuration on another server using Windows PowerShell.
   - **Specify an alternate source path** specifies the location of an image file containing the software needed to install the selected roles and features. Use this option when you have previously deleted teh source fiels from the system using Features on Demand.
10. Click Install. The *Installation progress* page appears. Depending on the roles and features installed, the wizard might display hyperlinks to the tools needed to perform required post-installation tasks. When the installation is completed, click Close to terminate the wizard.

After you install roles on your servers, the roles appear as icons Server Manager's the navigation pane. These icons actually represent ***role groups***. Each role group contains all instances of that role found on any of your added servers. You can therefore administer the role across all servers on which you have installed it.

Deploying Roles to VHDs

In addition to installing roles and features to servers on the network, Server Manager also enables administrators to install them to virtual machines currently in an offline state.

In an enterprise virtualization strategy, administrators frequently maintain virtual machines (VMs) in an offline state. For example, you might have an offline web server VM stored on a backup host server, in case the computer hosting your main web server VMs should fail. Server Manager enables you to select a virtual hard disk (VHD) file and install or remove roles and features without having to deploy the VM.

To install roles and/or features to an offline VHD file, use the following procedure.

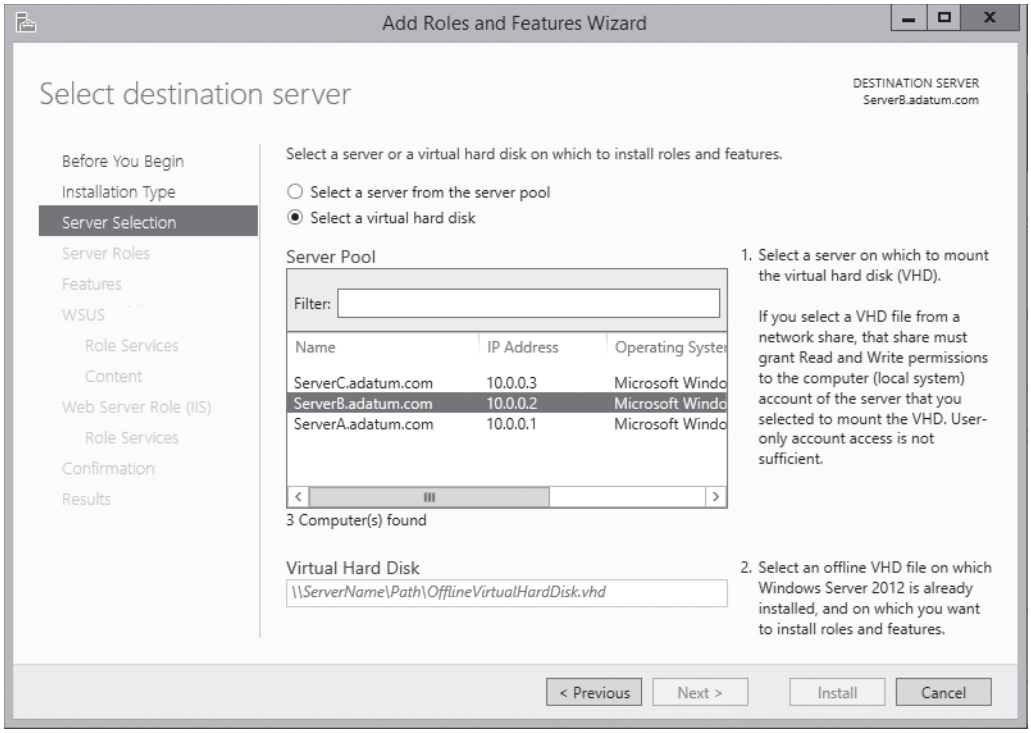
INSTALL ROLES AND FEATURES TO AN OFFLINE VHD FILE

GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges. The Server Manager window appears.

1. From the Manage menu, select Add Roles and Features. The Add Roles and Features Wizard appears, displaying the *Before you begin* page.
2. Click Next. The *Select Installation Type* page appears.
3. Leave the *Role-based or feature-based installation* radio button selected and click Next. The *Select Destination Server* page appears.
4. Select the Select a virtual hard disk radio button.
5. A Virtual Hard Disk text box appears at the bottom of the page. In this text box, type in or browse to the location of the VHD file you want to modify.
6. In the Server Pool box, select the server that the wizard should use to mount the VHD file, as shown in Figure 2-8, and click Next. The *Select Server Roles* page appears.

Figure 2-8

The *Select Destination Server* page in the Add Roles and Features Wizard



TAKE NOTE*

The wizard must mount the VHD file on the server you select, and look inside and determine which roles and features are already installed and which are available for installation. Mounting a VHD file makes it available only through the computer's file system; it is not the same as starting the virtual machine using the VHD.

7. Select the role or roles you want to install on the selected server, adding the required dependencies, if necessary, and click Next. The *Select features* page appears.
8. Select any features you want to install in the selected server and click Next. Dependencies also might appear for your feature selections.
9. The wizard then displays pages specific to the roles and/or features you have chosen, enabling you to select role services and configure other settings. Complete each of the role- or feature-specific pages and click Next. A *Confirmation* page appears.
10. Click Install. The *Installation progress* page appears.
11. When the installation is completed, click Close to dismount the VHD and terminate the wizard.

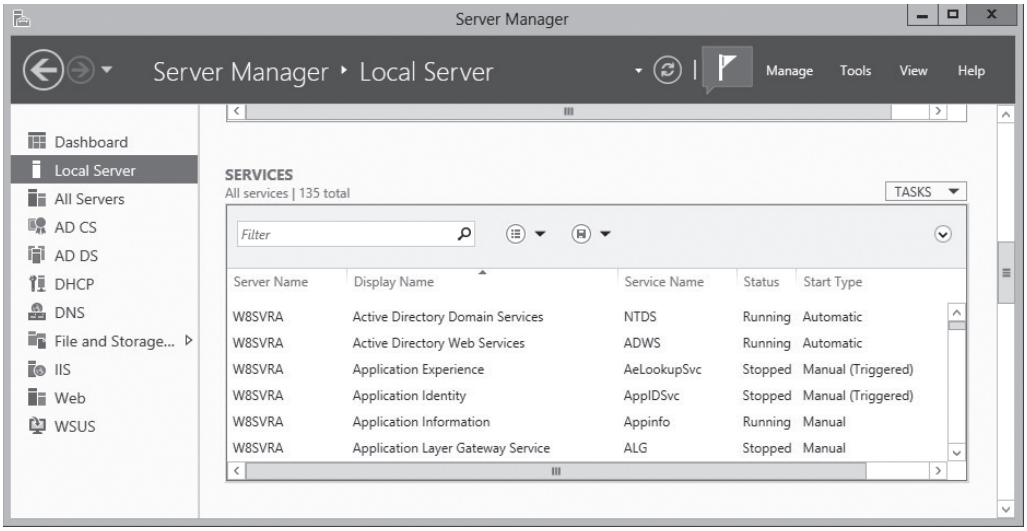
Configuring Services

Most Windows Server roles and many features include *services*, programs that run continuously in the background, typically waiting for a client process to send a request to them. Server Manager provides access to services running on servers all over the network.

When you first look at the Local Server homepage in Server Manager, one tile that you find there is the Services tile, as shown in Figure 2-9. This tile lists all the services installed on the server and specifies the operational status and their Start Types. When you right-click a service, the context menu provides controls that enable you to start, stop, restart, pause, and resume the service.

Figure 2-9

The Services tile in Server Manager



The Services tile in Server Manager is not unlike the traditional Services MMC snap-in found in previous versions of Windows Server. However, although you can start and stop a service in Server Manager, you cannot modify its Start Type, which specifies whether the service should start automatically with the operating system. For that, you must use the Services MMC snap-in.

Another difference of the Services tile in Windows Server 2012 R2 Server Manager is that it appears in many locations throughout Server Manager, displaying a list of services for a different context in each location. This is a good example of the organizational principle of the new Server Manager. The same tools, repeated in many places, provide a consistent management interface to different sets of components.

For example, when you select the All Servers icon in the navigational pane, you see first the Servers tile, as usual, containing all the servers you have added to the Server Manager console. When you select some or all servers and scroll down to the Services tile, you see the same display as before, except that it now contains all services for all the computers you selected. This enables you to monitor the services on all servers at once.

In the same way, when you select one of the role group icons, you can select from the servers running that role and the Services tile will contain only the services associated with that role for the servers you selected.

To manipulate other server configuration settings, you must use the Services MMC snap-in as mentioned earlier. However, you can launch that, and many other snap-ins, by using Server Manager.

After selecting a server from the Servers pane in any group homepage, click the Tools menu to display a list of the server-specific utilities and MMC snap-ins, including the Services snap-in, directed at the selected server.

■ Delegating Server Administration

THE BOTTOM LINE

As networks grow in size, so does the number of administrative tasks to perform regularly and the size of the IT staffs needed to perform them. Delegating administrative tasks to specific individuals is a natural part of enterprise server management, as is assigning those individuals the permissions they need—and only the permissions they need—to perform those tasks.

On smaller networks, with small IT staffs, it is common for task delegation to be informal, and for everyone in the IT department to have full access to the entire network. However, on larger networks, with larger IT staffs, this becomes increasingly impractical. For example, you might want the newly hired junior IT staffers to be able to create new user accounts, but you do not want them to be able to redesign your Active Directory tree or change the CEO's password.

Delegation, therefore, is the practice by which administrators grant other users a subset of the privileges that they themselves possess. As such, delegation is as much a matter of restricting permissions as it is of granting them. You want to provide individuals with the privileges they need, while protecting sensitive information and delicate infrastructure.

■ Using Windows PowerShell Desired State Configuration

THE BOTTOM LINE

Desired State Configuration (DSC) is the next phase in the development of Windows PowerShell, a process that began over a decade ago and first appeared as a Windows component in Windows PowerShell 1.0, released in 2006.

Windows Server 2012 R2 expanded the functionality of Windows PowerShell by using the command line infrastructure as an underlayment for all of the new graphical capabilities in the operating system. Windows PowerShell 3.0 added thousands of new cmdlets, making it possible to accomplish any task you might perform in Server Manager from the command line.

In Windows PowerShell 4.0, DSC provides a new scripting model that enables administrators to create modules called *configurations*, which consist of *nodes* representing computers and *resources* that define elements that administrators want to define as part of the configuration for a particular node. The scripts use the standard Microsoft Operations Framework (MOF) format.

For example, a relatively simple script to deploy a Web server might appear as follows:

```
Configuration CompanyWeb
{
   Node “ServerB”
   {
      WindowsFeature IIS
      {
          Ensure = “Present”
          Name = “Web-Server”
      }
      File CopyWebSite
      {
         Ensure = “Present”
         Type = “Directory”
         Recurse = $true
```

```
            SourcePath = $WebsitePath
            DestinationPath = "C:\inetpub\wwwroot"
            Requires = "[WindowsFeature]IIS"
        }
    }
}
```

In this script, the node block identifies the computer to be configured, and the WindowsFeature and File blocks are both built-in resources that you can use to define the configuration you want to deploy. The WindowsFeature block specifies that the configuration must install the Web-Server role, and the File block copies the content files for a Web site to the node from a location defined by the $WebsitePath variable. DSC includes many other built-in resources that you can use to define more complex configuration elements, such as system services, registry settings, environment variables, and user and group accounts. It is also possible for administrators to create their own custom resources.

Once you have created a configuration script, you can deploy it by executing the defined configuration name–in this case CompanyWeb–from a Windows PowerShell prompt, using the Start-DscConfiguration cmdlet.

In large enterprise deployments, administrators can create a centralized DSC server by installing the Windows PowerShell Desired State Configuration Service, a Windows PowerShell feature that uses the Internet Information Services Web server to deploy configuration logic and data to nodes all over the network. After storing DSC configuration scripts on the server, administrators can configure nodes to check periodically for changes in their configurations, using the Test-DscConfiguration cmdlet, or configure the server to push new configurations to nodes as needed.

■ Business Case Scenarios

Scenario 2-1: Installing Roles with a Batch File

Mark Lee is an IT technician whose supervisor has assigned the task of configuring 20 new servers, which Mark is to ship to the company's branch offices around the country. He must configure each server to function as a file server with support for DFS and UNIX clients, a print server with support for Internet and UNIX printing, a fax server, and a secured, intranet Web/FTP server for domain users. Write a Windows PowerShell script that Mark can use to install all of the required software elements on a server.

Scenario 2-2: Deploying Roles to VHDs

You maintain several virtual machines (VMs) in an offline state. How do you proceed to add a particular role to one of those VMs?

3 LESSON

Configuring Local Storage

■ Planning Server Storage

THE BOTTOM LINE

A Windows server can conceivably perform its tasks using the same type of storage as a workstation—that is, one or more standard hard disks connected to a standard drive interface such as Serial ATA (SATA). However, a server's I/O burdens vary quite differently from those of a workstation, and file requests from dozens or hundreds of users can easily overwhelm a standard storage subsystem. Also, standard hard disks offer no fault tolerance and their scalability is limited.

A variety of storage technologies are better suited for server use. The process of designing a storage solution for a server depends on several factors, including the following:

- The amount of storage the server needs
- The number of users that will be accessing the server at the same time
- The sensitivity of the data to be stored on the server
- The importance of the data to the organization

Determining the Number of Servers Needed

When is one big file server preferable to several smaller ones?

One of the most frequently asked questions when planning a server deployment is whether using one big server or several smaller ones is better. In the past, you might have considered the advantages and disadvantages of using one server to perform several roles versus distributing the roles among several smaller servers. Today, however, the emphasis is on virtualization, which means that although you might have many virtual machines running different roles, they could all be running on a single large physical server.

If you are considering large physical servers or your organization's storage requirements are extremely large, you must also consider the inherent storage limitations of Windows Server 2012 R2, as listed in Table 3-1.

Table 3-1
Windows Server 2012 R2 Storage Limitations

Attribute	Limit Based on the On-Disk Format
Maximum size of a single file	2^{64}-1 bytes
Maximum size of a single volume	Format supports 2^{78} bytes with 16KB cluster size. Windows stack addressing allows 2^{64} bytes
Maximum number of files in a directory	2^{64}
Maximum number of directories in a volume	2^{64}
Maximum filename length	32K Unicode characters
Maximum path length	32K
Maximum size of any storage pool	4 petabytes
Maximum number of storage pools in a system	No limit
Maximum number of spaces in a storage pool	No limit

The number of sites your enterprise network encompasses and the technologies you use to provide network communication between those sites can also affect your plans. If, for example, your organization has branch offices scattered around the world and uses relatively expensive wide area networking (WAN) links to connect them, installing a server at each location would probably be more economical than to have all your users access a single server via WAN links.

Within each site, the number of servers you need can depend on how often your users work with the same resources and how much fault tolerance and high availability you want to build into the system. For example, if each department in your organization typically works with its own applications and documents and rarely needs access to those of other departments, deploying individual servers to each department might be preferable. If everyone in your organization works with the same set of resources, centralized servers might be a better choice.

Estimating Storage Requirements

The amount of storage space you need in a server depends on various factors, not just the initial requirements of your applications and users.

For an application server, start by allocating the amount of space needed for the application files themselves, plus any other space the application needs, as recommended by the developer. If users will store documents on the server, allocate a specific amount of space for each user the server will support. Then, factor in the potential growth of your organization and your network, both in terms of additional users and additional space required by each user, and of the application itself, in terms of data files and updates.

In addition to the space allocated to applications and individual users, you must also consider the storage requirements for the following server elements:

- **Operating system:** The size of the operating system installation depends on the roles and features you choose to install. A typical Windows Server 2012 R2 installation with the File Services role needs just over 10 GB, but the system requirements recommend 40 GB.
- **Paging file:** The traditional formula for the size of the paging file—pagefile.sys—on a computer running Windows is 1½ times the amount of memory installed on the computer. However, this formula has now come into question, due to the large amounts

of memory in some servers and the increasing use of Hyper-V. Virtual machines require physical, not virtual, memory, so you do not need to count the memory allotted to your VMs when calculating your paging file size.

- **Memory dump:** When Windows Server 2012 R2 experiences a serious malfunction, it offers to dump the contents of the system memory to a file, which technicians can use for diagnostic purposes. The maximum size for a memory dump file is the amount of memory installed in the computer plus 1 MB. However, blue screens are relatively rare on Windows servers these days, and unless you are troubleshooting a chronic problem with the aid of a technician who can make use of a memory dump, you probably do not need to reserve space for this purpose.
- **Log files:** Be sure to consider any applications that maintain their own logs, in addition to the operating system logs. You can configure the maximum log size for Windows event logs and for most application logs, and add those values to calculate the total log space required.
- **Shadow copies:** The Windows Server 2012 R2 shadow copies feature automatically retains copies of files on a server volume in multiple versions from specific points in time. Shadow copies can use up to 10% of a volume, by default. However, Microsoft recommends enlarging this value for volumes containing frequently modified files.
- **Fault tolerance:** Fault-tolerance technologies, such as disk mirroring and disk parity, can profoundly affect disk consumption. Mirroring disks cuts the effective storage size in half, and parity can reduce it by as much as one third.

Using Storage Spaces

Windows Server 2012 R2 includes a new disk virtualization technology called ***Storage Spaces***, which enables a server to concatenate storage space from individual physical disks and allocate that space to create virtual disks of any size supported by the hardware.

This type of virtualization is a feature often found in SAN and NAS technologies, which require a substantial investment in specialized hardware and administrative skill. Storage Space provides similar capabilities, using standard direct-attached disk drives or simple external JBOD arrays.

Storage Spaces uses unallocated disk space on server drives to create storage pools. A ***storage pool*** can span multiple drives invisibly, providing an accumulated storage resource that you can expand or reduce as needed by adding disks to or removing them from the pool. By using the space in the pool, you can create ***virtual disks*** of any size.

Once created, a virtual disk behaves much like a physical disk, except that the actual bits might be stored on any number of physical drives in the system. Virtual disks can also provide fault tolerance by using the physical disks in the storage pool to hold mirrored or parity data.

Virtual disks can also be thinly provisioned, meaning that while you specify a maximum size for the disk, it starts out small and grows as you add data to it. You can therefore create a virtual disk with a maximum size that is larger than that of your storage space.

For example, if you plan to allocate a maximum of 10 TB for your database files, you can create a thin 10 TB virtual disk, even if you only have a 2 TB storage pool. The application using the disk will function normally, gradually adding data until the storage pool is nearly consumed, at which point the system notifies you to add more space to the pool. You can then install more physical storage and add it to the pool, gradually expanding it until it can support the entire 10 TB required by the disk.

After creating a virtual disk, you can create volumes on it, just as you would on a physical disk. Server Manager provides the tools needed to create and manage storage pools and virtual disks, as well as the capability to create volumes and file system shares, with some limitations.

■ Understanding Windows Disk Settings

THE BOTTOM LINE When preparing a disk for use, Windows Server 2012 R2 servers often require different settings than workstations.

When you install Windows Server 2012 R2 on a computer, the setup program automatically performs all preparation tasks for the primary hard disk in the system. However, when you install additional hard disk drives on a server, or when you want to use different settings from the system defaults, you must perform the following tasks manually:

- **Select a partitioning style:** Windows Server 2012 R2 supports two hard disk partition styles: the master boot record (MBR) partition style and the GUID (globally unique identifier) partition table (GPT) partition style. You must choose one of these partition styles for a drive; you cannot use both.
- **Select a disk type:** Windows Server 2012 R2 supports two disk types: basic and dynamic. You cannot use both types on the same disk drive, but you can mix disk types in the same computer.

Selecting a Partition Style

The term ***partition style*** refers to the method Windows operating systems use to organize partitions on the disk.

Servers running Windows Server 2012 R2 computers can use either of the following hard disk partition styles:

- ***Master Boot Record (MBR):*** The MBR partition style has been around since before Windows and is still a common partition style for x86-based and x64-based computers.
- ***GUID Partition Table (GPT):*** GPT has existed since the late 1990s, but no x86 versions of Windows prior to Windows Server 2008 and Windows Vista supports it. Today, most operating systems support GPT, including Windows Server 2012 R2.

MBR uses a partition table to point to the locations of the partitions on the disk. The MBR disk partitioning style supports volumes up to 2 TB in size, and up to either four primary partitions or three primary partitions and one extended partition on a single drive.

GPT varies from MBR in that partitions, rather than hidden sectors, store data critical to platform operation. GPT-partitioned disks also use redundant primary and backup partition tables for improved integrity. Although GPT specifications permit an unlimited number of partitions, the Windows implementation restricts partitions to 128 per disk. The GPT disk partitioning style supports volumes up to 18 exabytes (1 exabyte = 1 billion gigabytes, or 2^{60} bytes).

Unless the computer's architecture provides support for an Extensible Firmware Interface (EFI)–based boot partition, it is not possible to boot from a GPT disk. If this is the case, the system drive must be an MBR disk, and you can use GPT only on separate non-bootable disks used for data storage.

Before Windows Server 2008 and Windows Vista, all x86-based Windows computers used only the MBR partition style. Computers based on the x64 platform could use either the MBR or GPT partition style, as long as the GPT disk was not the boot disk.

Now that hard drives larger than 2 TB are readily available, the selection of a partition style is more critical than ever. When you initialize a physical disk using the traditional Disk Management snap-in, MBR is the default partition style, as it always has been. You can also use the snap-in to convert a disk between MBR and GPT partition styles, although you can do so only on disks that do not have partitions or volumes created on them.

When you use Server Manager to initialize a disk in Windows Server 2012 R2, it uses the GPT partition style, whether the disk is physical or virtual. Server Manager has no controls supporting MBR, although it does display the partition style in the Disks tile.

Table 3-2 compares some of the characteristics of the MBR and GPT partition styles.

Table 3-2

MBR and GPT Partition Style Comparison

Master Boot Record (MBR)	GUID Partition Table (GPT)
Supports up to four primary partitions or three primary partitions and one extended partition, with unlimited logical drives on the extended partition	Supports up to 128 primary partitions
Supports volumes up to 2 terabytes	Supports volumes up to 18 exabytes
Hidden (unpartitioned) sectors store data critical to platform operation	Partitions store data critical to platform operation
Replication and cyclical redundancy checks (CRCs) are not features of MBR's partition table	Replication and CRC protection of the partition table provide increased reliability

Understanding Disk Types

Most personal computers use basic disks because they are easiest to manage. Advanced volume types require the use of dynamic disks.

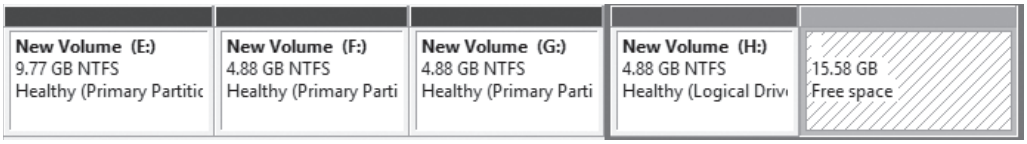
A ***basic disk*** using the MBR partition style uses primary partitions, extended partitions, and logical drives to organize data. A primary partition appears to the operating system as though it is a physically separate disk and can host an operating system, in which case it is known as the active partition.

During the operating system installation, the setup program creates a system partition and a boot partition. The system partition contains hardware-related files that the computer uses to start. The boot partition contains the operating system files, which are stored in the Windows file folder. In most cases, these two partitions are one and the same, the active primary partition that Windows uses when starting. The active partition tells the computer which system partition and operating system to use to start Windows.

When you work with basic MBR disks in Windows Server 2012 R2, you can create three volumes that take the form of primary partitions. When you create the fourth volume, the system creates an extended partition, with a logical drive on it, of the size you specified. If the disk still has free space left, the system allocates it to the extended partition (see Figure 3-1), which you can use to create additional logical drives.

Figure 3-1

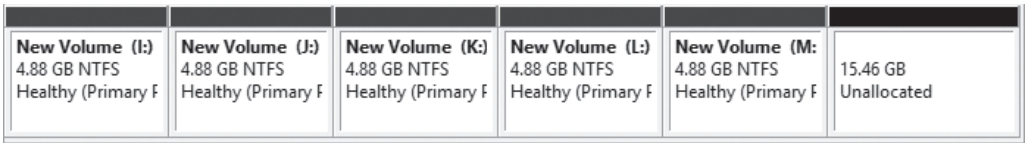
Primary and extended partitions on a basic disk using MBR



When you select the GPT partition style, the disk still appears as a basic disk, but you can create up to 128 volumes, each of which appears as a primary partition, as shown in Figure 3-2. GPT disks have no extended partitions or logical drives.

Figure 3-2

Primary partitions on a basic disk using GPT



The alternative to using a basic disk is to convert it to a ***dynamic disk***. Converting a basic disk to a dynamic disk creates a single partition that occupies the entire disk. You can then create an unlimited number of volumes out of the space in that partition. Dynamic disks support several different types of volumes, as described in the next section.

■ Working with Disks

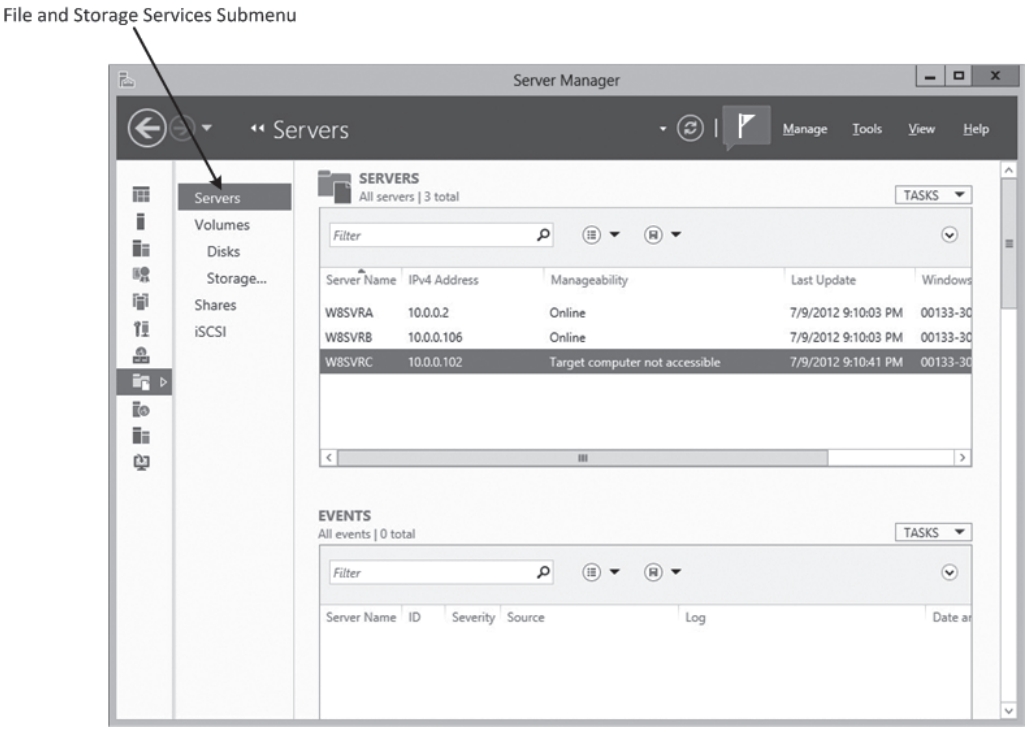
THE BOTTOM LINE

Windows Server 2012 R2 includes tools that enable you to manage disks graphically or from the command prompt.

All Windows Server 2012 R2 installations include the File and Storage Services role, which causes Server Manager to display a submenu when you click the icon in the navigational pane (see Figure 3-3). This submenu provides access to homepages that enable you to manage volumes, disks, storage pools, shares, and iSCSI devices.

Figure 3-3

The File and Storage Services submenu in Server Manager



Server Manager is the only graphical tool that can manage storage pools and create virtual disks. It can also perform some—but not all—of the standard disk and volume management operations on physical disks. As with the other Server Manager homepages, the File and Storage Services pages also enable you to perform tasks on any servers you have added to the interface.

Disk Management is a Microsoft Management Console (MMC) snap-in that is the traditional tool for performing disk-related tasks, such as the following:

- Initializing disks
- Selecting a partition style
- Converting basic disks to dynamic disks
- Creating partitions and volumes
- Extending, shrinking, and deleting volumes
- Formatting partitions and volumes
- Assigning and changing drive letters and paths

To access the Disk Management snap-in, you can open the Computer Management console and select Disk Management.

You can also manage disks and volumes from the command line by using the DiskPart.exe utility.

Creating and Mounting VHDs

Hyper-V relies on the ***Virtual Hard Disk (VHD)*** format to store virtual disk data in files that can easily be transferred from one computer to another.

The Disk Management snap-in in Windows Server 2012 R2 enables you to create VHD and VHDX files and mount them on the computer. As soon as the VHDs or VHDXes are mounted, you can treat them just like physical disks and use them to store data. Dismounting a VHD or VHDX packages the stored data in the file, so you can copy or move it as needed.

To create a VHD in Disk Management, use the following procedure.

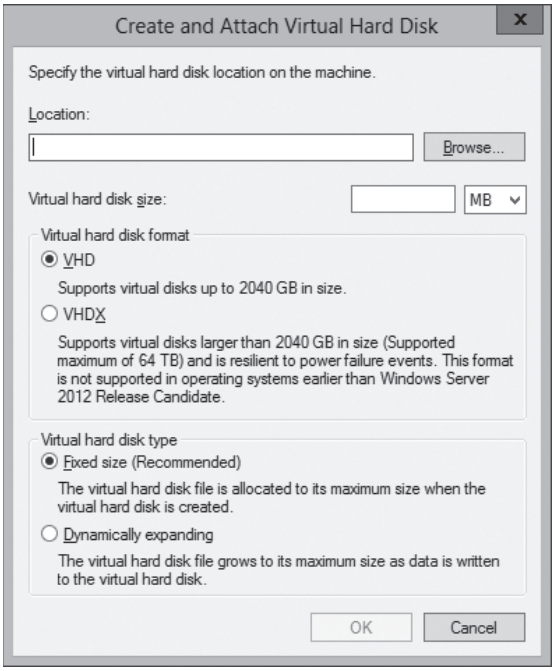
CREATE A VHD

GET READY. Log on to Windows Server 2012 R2, using an account with Administrator privileges. The *Server Manager* window appears.

1. Click Tools > Computer Management.
2. In the Computer Management console, click Disk Management. The Disk Management snap-in appears.
3. From the Action menu, select Create VHD. The *Create and Attach Virtual Hard Disk* dialog box appears, as shown in Figure 3-4.

Figure 3-4

The *Create and Attach Virtual Hard Disk* dialog box

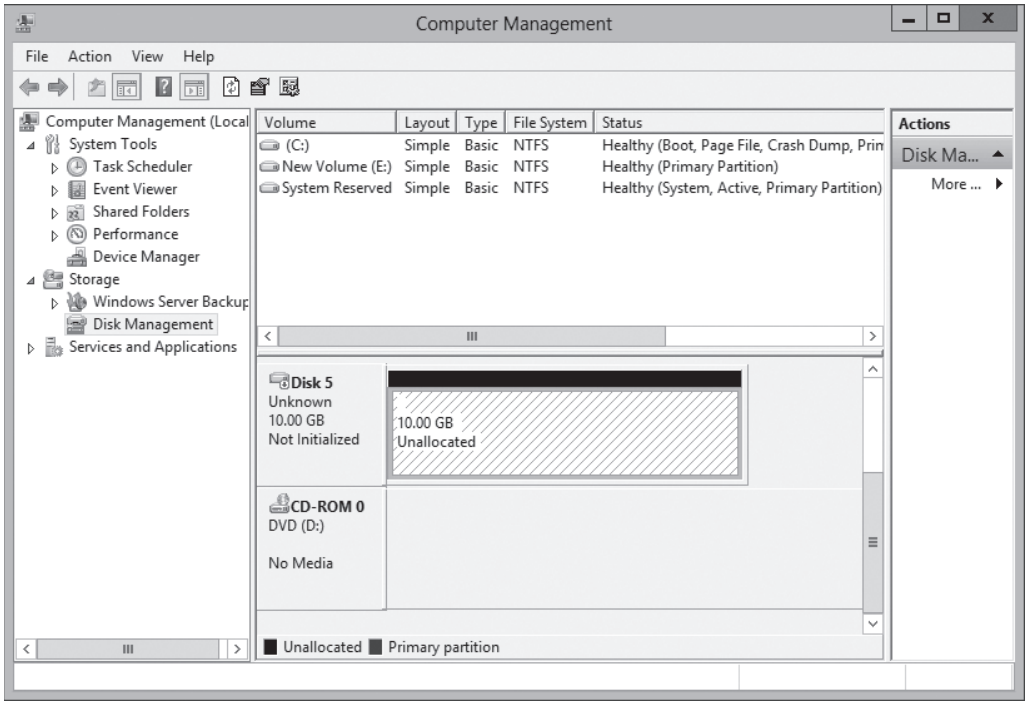


4. In the Location text box, specify the path and name for the file you want to create.
5. In the Virtual hard disk size text box, specify the maximum size of the disk you want to create.
6. Select one of the following virtual hard disk format options:
   - VHD: The original and more compatible format, which supports files up to 2,040 GB.
   - VHDX: A new version of the format that supports files up to 64 TB, but can be read only by computers running Windows Server 2012 R2, Windows Server 2012 and Windows 8.

7. Select one of the following virtual hard disk type options:
   - Fixed size allocates all disk space for the VHD file at once.
   - Dynamically expanding allocates disk space to the VHD or VHDX file as you add data to the virtual hard disk.
8. Click OK. The system creates the VHD or VHDX file and attaches it, so that it appears as a disk in the snap-in, as shown in Figure 3-5.

Figure 3-5

A newly created and attached VHD



After you create and attach the VHD or VHDX, it appears as an uninitialized disk in the Disk Management snap-in and in Server Manager. By using either tool, you can initialize the disk and create volumes on it, just as you would a physical disk. After storing data on the volumes, you can detach the VHD or VHDX and move it to another location or mount it on a Hyper-V virtual machine.

Creating a Storage Pool

After you install your physical disks, you can concatenate their space into a storage pool, from which you can create virtual disks of any size.

To create a storage pool via Server Manager, use the following procedure.

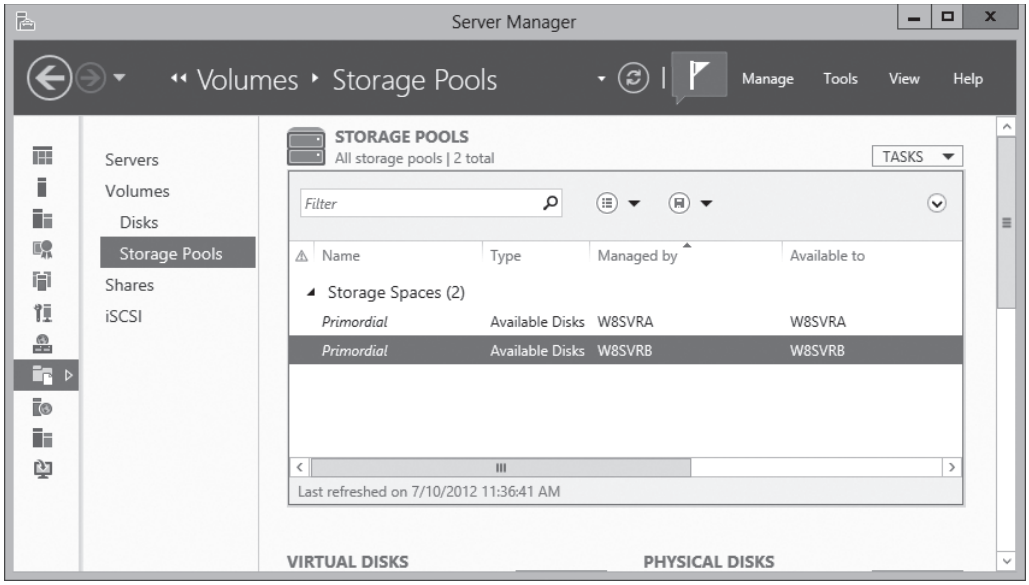
CREATE A STORAGE POOL

GET READY. Log on to Windows Server 2012 R2, using an account with Administrator privileges.

1. In the *Server Manager* window, click the File and Storage Services icon and, in the submenu that appears, click Storage Pools. The *Storage Pools* homepage appears, as shown in Figure 3-6.

Figure 3-6

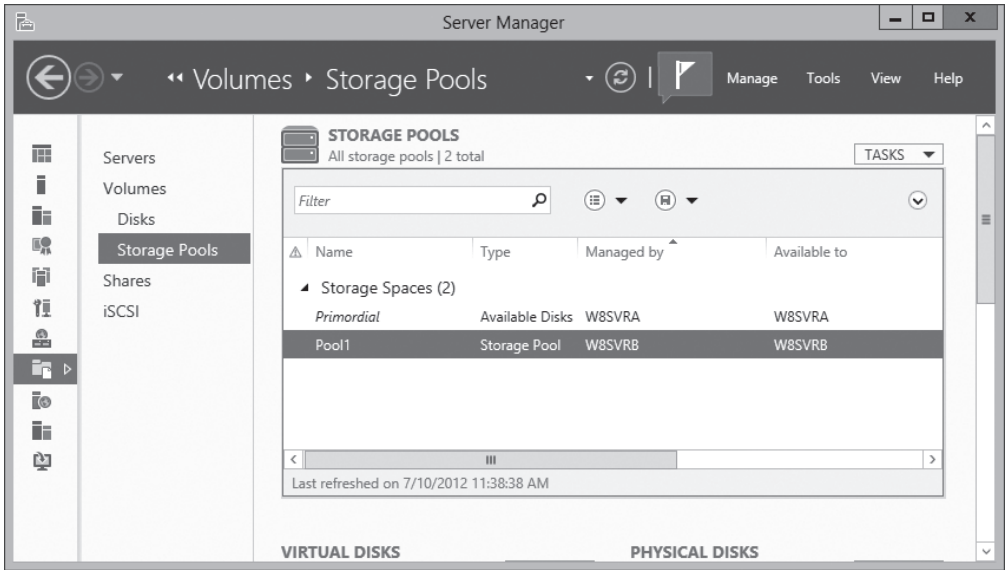
The *Storage Pools* homepage



2. In the *Storage Pools* tile, select the primordial space on the server where you want to create the pool and then, from the Tasks menu, select New Storage Pool. The *New Storage Pool Wizard* appears, displaying the *Before you begin* page.
3. Click Next. The *Specify a storage pool name and subsystem* page appears.
4. In the Name text box, type the name you want to assign to the storage pool. Then, select the server on which you want to create the pool and click Next. The *Select physical disks for the storage pool* page appears.
5. Select the check boxes for the disks you want to add to the pool and click Next. The *Confirm selections* page appears.
6. Click Create. The wizard creates the new storage pool and the *View results* page appears.
7. Click Close. The wizard closes, and the *Storage Pools* homepage lists the new pool, as shown in Figure 3-7.

Figure 3-7

A new pool on the *Storage Pools* homepage



CLOSE the *Server Manager* window.

After you create a storage pool, you can modify its capacity by adding or removing physical disks. The *Tasks* menu in the *Physical Disks* tile on the *Storage Pools* homepage contains the following options:

- **Add Physical Disk** enables you to add a physical disk to the pool, as long as it is initialized and does not contain any volumes
- **Remove Disk** removes the space provided by a physical disk from the storage pool. This option appears only if all data already has been evicted from the disk.

Creating a Simple Volume

Technically speaking, you create partitions on basic disks and volumes on dynamic disks. This is not just an arbitrary change in nomenclature. Converting a basic disk to a dynamic disk actually creates one big partition, occupying all space on the disk. The volumes you create on the dynamic disk are logical divisions within that single partition.

Windows versions prior to 2008 use the correct terminology in the Disk Management snap-in. The menus enable you to create partitions on basic disks and volumes on dynamic disks. Windows Server 2012 R2 uses the term *volume* for both disk types, and enables you to create any of the available volume types, whether the disk is basic or dynamic. If the volume type you select is not supported on a basic disk, the wizard converts it to a dynamic disk as part of the volume creation process.

Despite the menus that refer to basic partitions as volumes, the traditional rules for basic disks remain in effect. The New Simple Volume menu option on a basic disk creates up to three primary partitions. When you create a fourth volume, the wizard actually creates an extended partition and a logical drive of the size you specify. If any space remains on the disk, you can create additional logical drives in the extended partition.

To create a new simple volume on a basic or dynamic disk using the Disk Management snap-in, use the following procedure.

WARNING When you use ***DiskPart.exe***, a command-line utility included with Windows Server 2012 R2, to manage basic disks, you can create four primary partitions, or three primary partitions and one extended partition. The DiskPart.exe utility contains a superset of the commands supported by the Disk Management snap-in. In other words, DiskPart can do everything Disk Management can do, and more. However, while the Disk Management Snap-in prevents you from unintentionally performing actions that might result in data loss, DiskPart has no safeties, and thus does not prohibit you from performing such actions. For this reason, Microsoft recommends that only advanced users use DiskPart and that they use it with due caution.

CREATE A NEW SIMPLE VOLUME

GET READY. Log on to Windows Server 2012 R2, using an account with Administrator privileges.

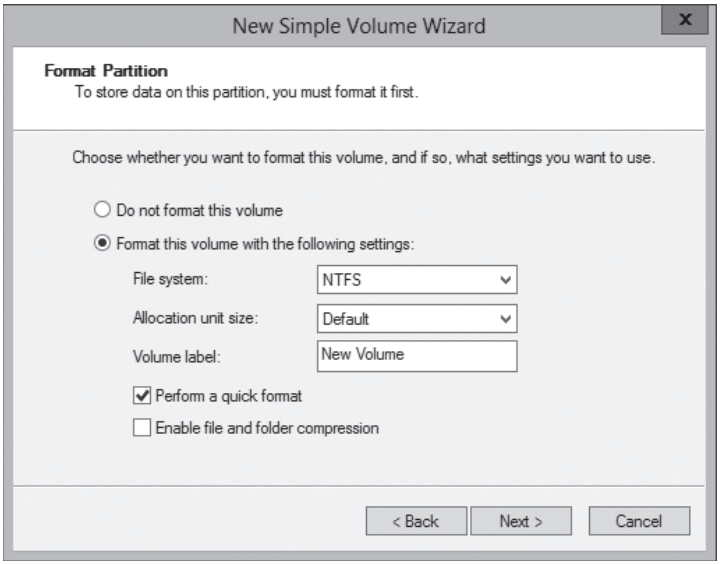
1. In the *Server Manager* window, click Tools > Computer Management.
2. In the *Computer Management* console, click Disk Management.
3. In the *Graphical View* of the *Disk Management* snap-in, right-click an unallocated disk area on which you want to create a volume. From the context menu, select New Simple Volume. The *New Simple Volume Wizard* appears.
4. Click Next to dismiss the *Welcome* page. The *Specify Volume Size* page appears.
5. Select the size for the new partition or volume, within the maximum and minimum limits stated on the page, using the Simple volume size in MB spin box, and then click Next. The *Assign Drive Letter or Path* page appears.
6. Configure one of the following options:
   - Assign the following drive letter: If you select this option, click the associated drop-down list for a list of available drive letters and select the letter you want to assign to the drive.
   - Mount in the following empty NTFS folder: If you select this option, either key the path to an existing NTFS folder or click Browse to search for or create a new folder. The folder you specify will list the entire contents of the new drive.

- Do not assign a drive letter or drive path: Select this option if you want to create the partition but are not yet ready to use it. When you do not assign a volume a drive letter or path, the drive is left unmounted and inaccessible. When you want to mount the drive for use, assign a drive letter or path to it.

7. Click Next. The *Format Partition* page appears, as shown in Figure 3-8.

Figure 3-8

The *Format Partition* page



8. Specify whether the wizard should format the volume and, if so, how. If you do not want to format the volume at this time, select the Do not format this volume option. If you do want to format the volume, select the Format this volume with the following settings option, and then configure the following associated options:
   - File system: Select the desired file system. The options available depend on the size of the volume, and can include ReFS, NTFS, exFAT, FAT32, or FAT.
   - Allocation unit size: Specify the file system's cluster size. The cluster size signifies the basic unit of bytes in which the system allocates disk space. The system calculates the default allocation unit size based on the size of the volume. You can override this value by clicking the associated drop-down list and then selecting one of the values. For example, if your client uses consistently small files, you may want to set the allocation unit size to a smaller cluster size.
   - Volume label: Specify a name for the partition or volume. The default name is New Volume, but you can change the name to anything you want.
   - Perform a quick format: When you select this option, Windows formats the disk without checking for errors. This is a faster method with which to format the drive, but Microsoft does not recommend it. When you check for errors, the system looks for and marks bad sectors on the disk so that your clients will not use those areas.
   - Enable file and folder compression: Selecting this option turns on folder compression for the disk. This option is available only for volumes being formatted with the NTFS file system.

9. Click Next. The Completing the New Simple Volume Wizard page appears.
10. Review the settings to confirm your options, and then click Finish. The wizard creates the volume according to your specifications.

CLOSE the console containing the Disk Management snap-in.

After you create a simple volume, you can use the Disk Management snap-in to modify its properties by extending or shrinking it, as described later in this lesson.

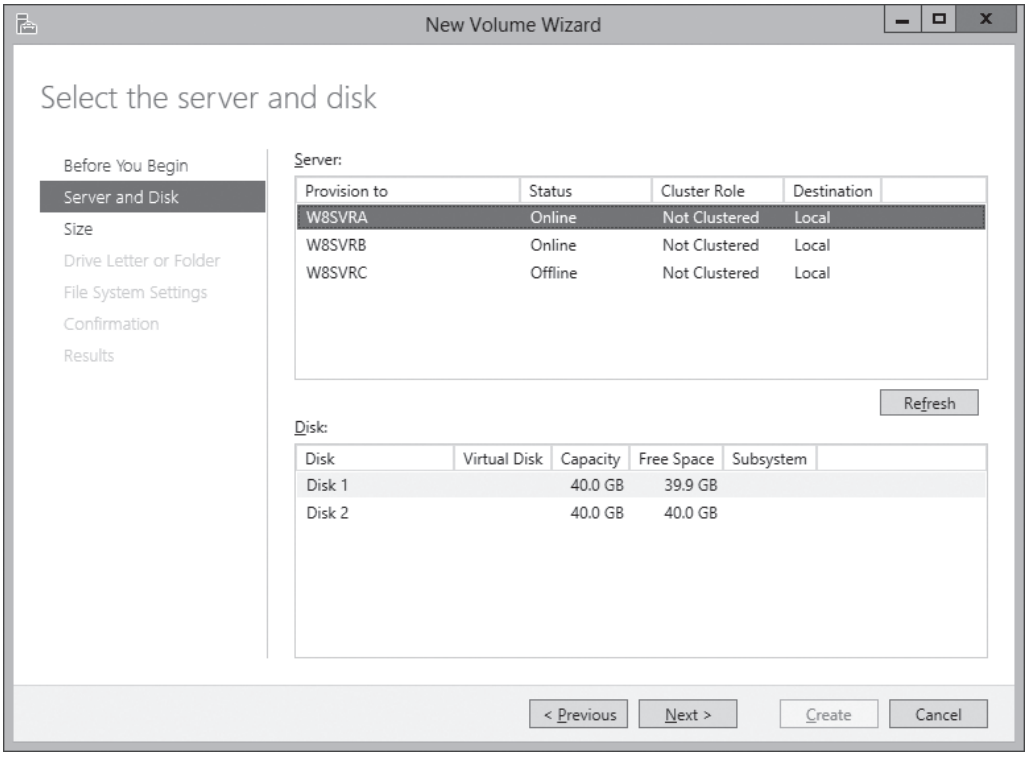
This procedure can create volumes on physical or virtual disks. You can also create simple volumes by using a similar wizard in Server Manager.

When you launch the New Volume Wizard in Server Manager, which you can do from the Volumes or Disks homepage, the options the wizard presents are virtually identical to those in the New Simple Volume Wizard in Disk Management.

The primary difference is that, like all Server Manager wizards, the New Volume Wizard includes a page that enables you to select the server and the disk on which you want to create volume, as shown in Figure 3-9. You can therefore use this wizard to create volumes on any disk, on any of your servers.

Figure 3-9

The *Select the server and disk* page in the New Volume Wizard in Server Manager



Extending and Shrinking Volumes and Disks

To extend or shrink a volume in the Disk Management snap-in, you simply right-click a volume and select *Extend Volume* or *Shrink Volume* from the context menu or from the *Action* menu.

The Disk Management snap-in extends existing volumes by expanding them into adjacent unallocated space on the same disk. When you extend a simple volume across multiple disks, the simple volume becomes a spanned volume. You cannot extend striped volumes.

In Server Manager, you can extend a simple volume using unallocated space on the same disk, but you cannot extend it to other disks to create a spanned volume.

To extend a volume on a basic disk, the system must meet the following requirements:

- A volume of a basic disk must be either unformatted or formatted with the NTFS file system.
- If you extend a volume that is actually a logical drive, the console first consumes the contiguous free space remaining in the extended partition. If you attempt to extend the logical drive beyond the confines of its extended partition, the extended partition expands to any unallocated space left on the disk.
- You can extend logical drives, boot volumes, or system volumes only into contiguous space, and only if the hard disk can be upgraded to a dynamic disk. The operating system enables you to extend other types of basic volumes into noncontiguous space but prompts you to convert the basic disk to a dynamic disk.

To extend a volume on a dynamic disk, the system must meet these requirements:

- When extending a simple volume, you can use only the available space on the same disk, if the volume is to remain simple.
- You can extend a simple volume across additional disks if it is not a system volume or a boot volume. However, after you expand a simple volume to another disk, it is no longer a simple volume; it becomes a spanned volume.
- You can extend a simple or spanned volume if it does not have a file system (a raw volume) or if you formatted it using the NTFS file system. (You cannot extend volumes using the FAT or FAT32 file systems.)
- You cannot extend mirrored or RAID-5 volumes, although you can add a mirror to an existing simple volume.

When shrinking volumes, the Disk Management snap-in frees up space at the end of the volume, relocating the existing volume's files, if necessary. The snap-in then converts that free space to new unallocated space on the disk. Server Manager cannot shrink volumes.

To shrink basic disk volumes and simple or spanned dynamic disk volumes, the system must meet the following requirements:

- The existing volume must not be full and must contain the specified amount of available free space for shrinking.
- The volume must not be a raw partition (one without a file system). Shrinking a raw partition that contains data might destroy the data.
- You can shrink a volume only if you formatted it using the NTFS file system. (You cannot shrink volumes using the FAT or FAT32 file systems.)
- You cannot shrink striped, mirrored, or RAID-5 volumes.
- You should always defragment a volume before you attempt to shrink it.

Physical disks, obviously, cannot be extended, but virtual disks can. In Server Manager, you can right-click a virtual disk and select *Extend Virtual Disk* from the context menu to display the Extend Virtual Disk dialog box.

If you elected to use thin provisioning when you created the virtual disk, you can even extend its size beyond the storage pool's current capacity. To actually store that much data on the disk, however, you must first expand the pool to provide enough space.

■ Business Case Scenario

Scenario 3-1: Planning Storage

On a new server running Windows Server 2012 R2, Morris created a storage pool that consists of two physical drives holding 1 TB each. Then he created three simple virtual disks out of the space in the storage pool. Using the Disk Management snap-in, Morris then created a RAID-5 volume out of the three virtual disks.

With this in mind, answer the following questions:

1. In what way is Morris's storage plan ineffectual at providing fault tolerance?
2. Why will adding a third disk to the storage pool fail to improve the fault tolerance of the storage plan?
3. How can Morris modify the storage plan to make it fault tolerant?

Configuring File and Share Access

LESSON 4

■ Creating Folder Shares

THE BOTTOM LINE

Sharing folders makes them accessible to network users.

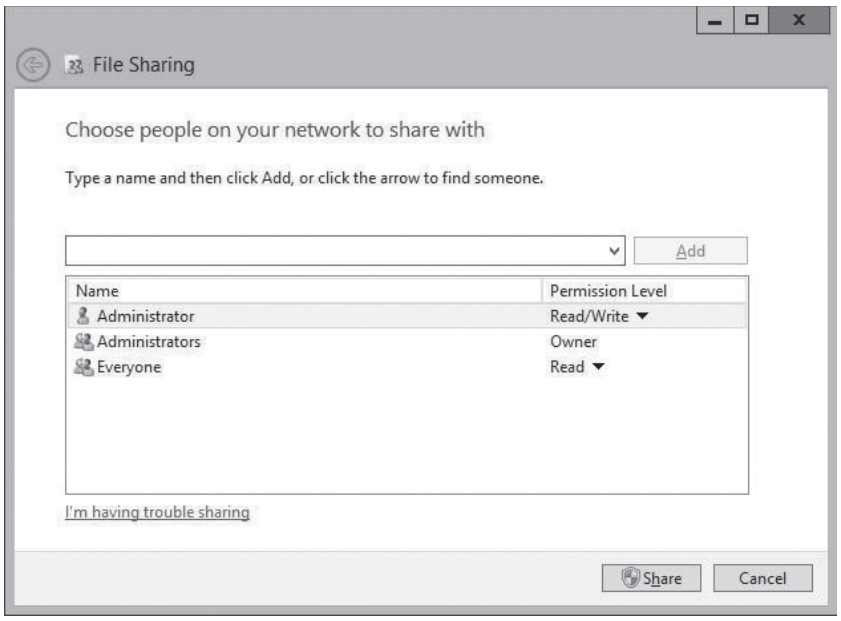
After you configure the disks on a file server, you must create shares for network users to be able to access those disks. You should have a sharing strategy in place by the time you are ready to actually create your shares. This strategy should consist of the following information:

- What folders you will share
- What names you will assign to the shares
- What permissions you will grant users to the shares
- What Offline Files settings you will use for the shares

If you are the Creator Owner of a folder, you can share it on a Windows Server 2012 R2 computer by right-clicking the folder in any File Explorer window, selecting *Share with, Specific People* from the context menu, and following the instructions in the *File Sharing* dialog box, as shown in Figure 4-1.

Figure 4-1

The *File Sharing* dialog box



This method of creating shares provides a simplified interface that contains only limited control over elements such as share permissions. You can specify only that the share users receive Read or Read/Write permissions to the share. If you are not the Creator Owner of the folder, you can access the *Sharing* tab of the folder's Properties sheet instead. Clicking the *Share* button launches the same dialog box, and clicking the *Advanced Sharing* button displays the Advanced Sharing dialog box. Clicking the Permissions button in the Advanced Sharing dialog box provides greater control over share permissions through the standard interface shown in "Setting Share Permissions," later in this lesson.

However, to take control of the shares on all your disks on all your servers and exercise granular control over their properties, use the *File and Storage Services* homepage in Server Manager.

Windows Server 2012 R2 supports two types of folder shares:

- *Server Message Blocks (SMB)* is the standard file-sharing protocol used by all versions of Windows.
- *Network File System (NFS)* is the standard file-sharing protocol used by most UNIX and Linux distributions.

When you install Windows Server 2012 R2, the setup program installs the Storage Services role service in the File and Storage Services role by default. When you create your first shared SMB folder, however, the system installs the File Server role service. To create NFS shares, you must manually install the Server for NFS role service.

To create a folder share using Server Manager, use the following procedure.

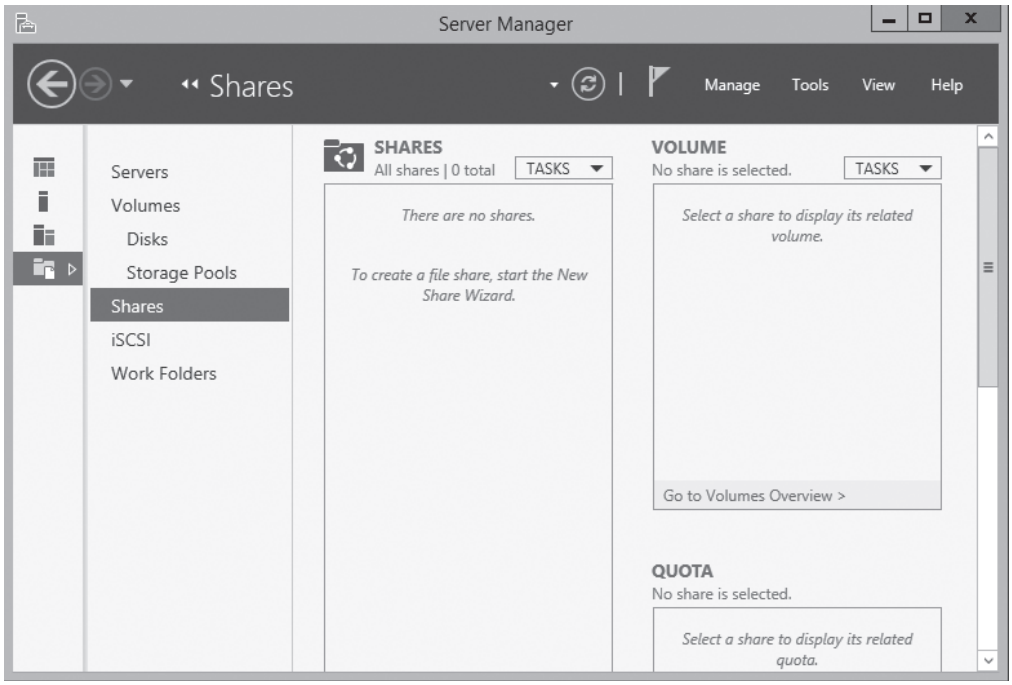
CREATE A FOLDER SHARE

GET READY. Log on to Windows Server 2012 R2, using an account with Administrator privileges. The Server Manager window appears.

1. Click the File and Storage Services icon and, in the submenu that appears, click Shares. The *Shares* homepage appears, as shown in Figure 4-2.

Figure 4-2

The Shares homepage



2. From the Tasks menu, select New Share. The *New Share Wizard* appears, displaying the *Select the profile for this share* page.
3. From the File share profile list, select one of the following options:
   - SMB Share–Quick provides basic SMB sharing with full share and NTFS permissions.
   - SMB Share–Advanced provides SMB sharing with full share and NTFS permissions and access to services provided by File Server Resource Manager.
   - SMB Share–Applications provides SMB sharing with settings suitable for Hyper-V and other applications.
   - NFS Share–Quick provides basic NFS sharing with authentication and permissions.
   - NFS Share–Advanced provides NFS sharing with authentication and permissions, plus access to services provided by File Server Resource Manager.
4. Click Next. The *Select the server and path for this share* page appears.
5. Select the server on which you want to create the share, and then either select a volume on the server or specify a path to the folder you want to share. Then click Next. The *Specify share name* page appears.
6. In the Share name text box, specify the name you want to assign to the share and click Next. The *Configure share settings* page appears.
7. Select any or all of the following options:
   - Enable access-based enumeration prevents users from seeing files and folders they do not have permission to access.
   - Allow caching of share enables offline users to access the contents of the share.
   - Enable BranchCache on the file share enables BranchCache servers to cache files accessed from this share.
   - Encrypt data access causes the server to encrypt remote file access to this share.

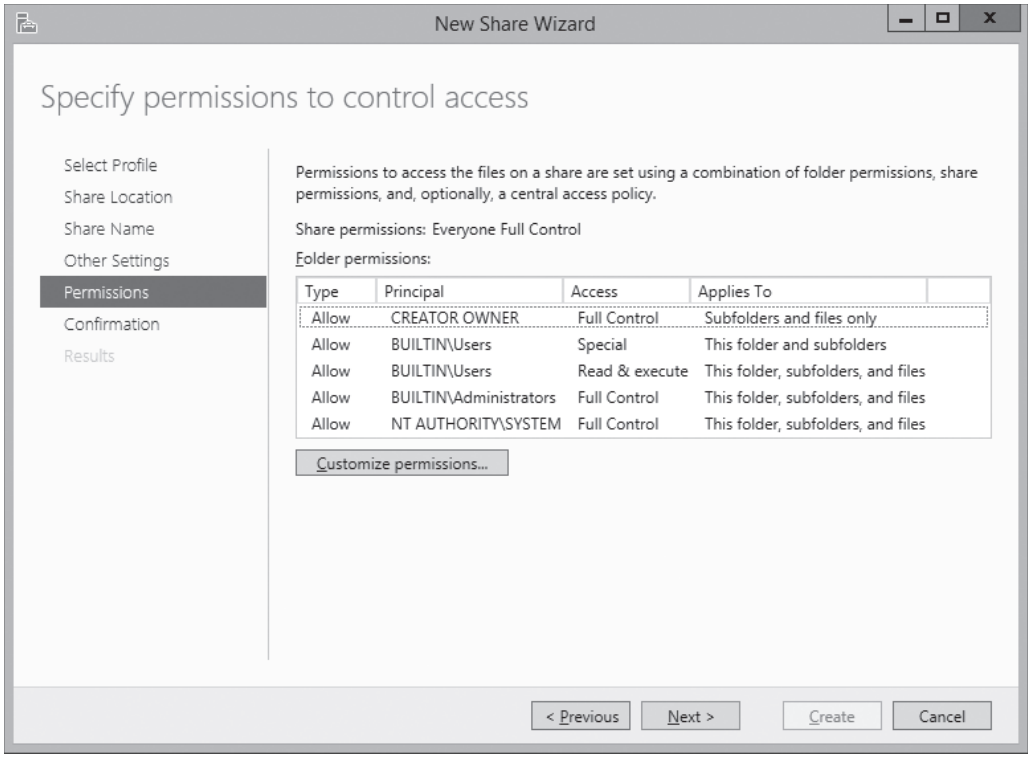
TAKE NOTE*

Offline Files, also known as client-side caching, is a Windows feature that enables client systems to maintain local copies of files they access from server shares. When a client selects the *Always available offline* option for a server-based file, folder, or share, the client system copies the selected data to the local drive and updates it regularly, so that the client user can always access it, even if the server is offline. To enable clients to use the Offline Files feature, the share must have the *Allow caching of share* check box selected. Windows Server 2012, Windows Server 2012 R2, Windows 8, and Windows 8.1 also have a new Always Offline mode for the Offline Files feature that causes clients to always use the cached copy of server files, providing better performance. To implement this mode, you must set the *Configure slow-link mode* Group Policy setting on the client to a value of 1 millisecond.

8. Click Next. The *Specify permissions to control access* page appears, as shown in Figure 4-3.

Figure 4-3

The *Specify permissions to control access* page of the New Share Wizard



9. Modify the default share and NTFS permissions as needed and click Next. The *Confirm selections* page appears.
10. Click Create. The *View results* page appears as the wizard creates the share.

CLOSE the *New Share Wizard.*

After you create a share with the wizard, the new share appears in the *Shares* tile of the *Shares* homepage in Server Manager.

You can now use the tile to manage a share by right-clicking it and opening its Properties sheet, or by clicking *Stop Sharing*. The Properties sheet for a share in Server Manager provides access to the exact same controls found on the *Specify permissions to control access* and *Configure share settings* pages in the New Share Wizard.

■ Assigning Permissions

THE BOTTOM LINE

Protect your data by controlling who can access it.

Using Windows Server 2012 R2, you can control access to a file server to provide network users with the access they need, while protecting other files against possible intrusion and damage, whether deliberate or not. To implement this access control, Windows Server 2012 R2 uses permissions.

Permissions are privileges granted to specific system entities, such as users, groups, or computers, enabling them to perform a task or access a resource. For example, you can grant a specific user permission to read a file while denying that same user the permissions needed to modify or delete the file.

Windows Server 2012 R2 has several sets of permissions that operate independently of each other. As a server administrator, you should be familiar with the operation of the following four permission systems:

- Share permissions control access to folders over a network. To access a file over a network, a user must have appropriate share permissions (and appropriate NTFS permissions, if the shared folder is on an NTFS volume).
- NTFS permissions control access to the files and folders stored on disk volumes formatted with the NTFS file system. To access a file, whether on the local system or over a network, a user must have the appropriate NTFS permissions.
- Registry permissions control access to specific parts of the Windows registry. An application that modifies registry settings or a user attempting to manually modify the registry must have the appropriate registry permissions.
- Active Directory permissions control access to specific parts of an AD DS hierarchy. Although file servers typically do not function as AD DS domain controllers, server administrators might use these permissions when servicing computers that are members of a domain.

All these permission systems operate independently of each other and sometimes combine to provide increased protection to a specific resource. For example, you might grant Ralph the NTFS permissions needed to access a spreadsheet stored on a file server volume. If Ralph sits down at the file server console and logs on as himself, he can access that spreadsheet. However, if Ralph is working at his own computer, he cannot access the spreadsheet until you create a share containing the file and grant Ralph the proper share permissions.

For network users to be able to access a shared folder on an NTFS drive, you must grant them both share permissions and NTFS permissions. As you saw earlier, you can grant these permissions as part of the share creation process, but you can also modify the permissions at any time afterward.

Understanding the Windows Permission Architecture

Permissions protect all files, folders, shares, registry keys, and AD DS objects.

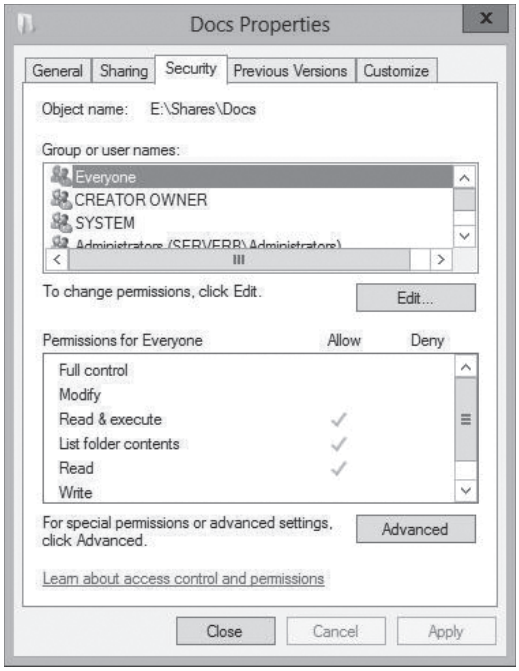
To store the permissions, each element has an ***access control list (ACL)***. An ACL is a collection of individual permissions, in the form of ***access control entries (ACEs)***. Each ACE consists of a ***security principal*** (the name of the user, group, or computer granted the permissions) and the specific permissions assigned to that security principal. When you manage permissions in any of the Windows Server 2012 R2 permission systems, you are actually creating and modifying the ACEs in an ACL.

It is important to understand that, in all Windows operating systems, permissions are stored as part of the protected element, not the security principal granted access. For example, when you grant a user the NTFS permissions needed to access a file, the ACE you create is stored in the file's ACL; it is not part of the user account. You can move the file to a different location, and its permissions go with it.

To manage permissions in Windows Server 2012 R2, you use a tab in the protected element's Properties sheet, like the one shown in Figure 4-4, with the security principals listed at the top and the permissions associated with them at the bottom. Share permissions are typically found on a *Share Permissions* tab, and NTFS permissions are located on a *Security* tab. All Windows permission systems use the same basic interface, although the permissions themselves vary. Server Manager also provides access to NTFS and share permissions, using a slightly different interface.

Figure 4-4

The Security tab of a Properties sheet



Understanding Basic and Advanced Permissions

The permissions protecting a particular system element are not like the keys to a lock, which provide either full access or no access at all. Permissions are designed to be granular, enabling you to grant specific degrees of access to security principals.

For example, you can use NTFS permissions to control not only who has access to a spreadsheet, but also the degree of access. You might grant Ralph permission to read and modify the spreadsheet, but Alice can only read it, and Ed cannot see it at all.

To provide this granularity, each Windows permission system has an assortment of permissions that you can assign to a security principal in any combination. Depending on the permission system you are working with, you might have dozens of different permissions available for a single system element.

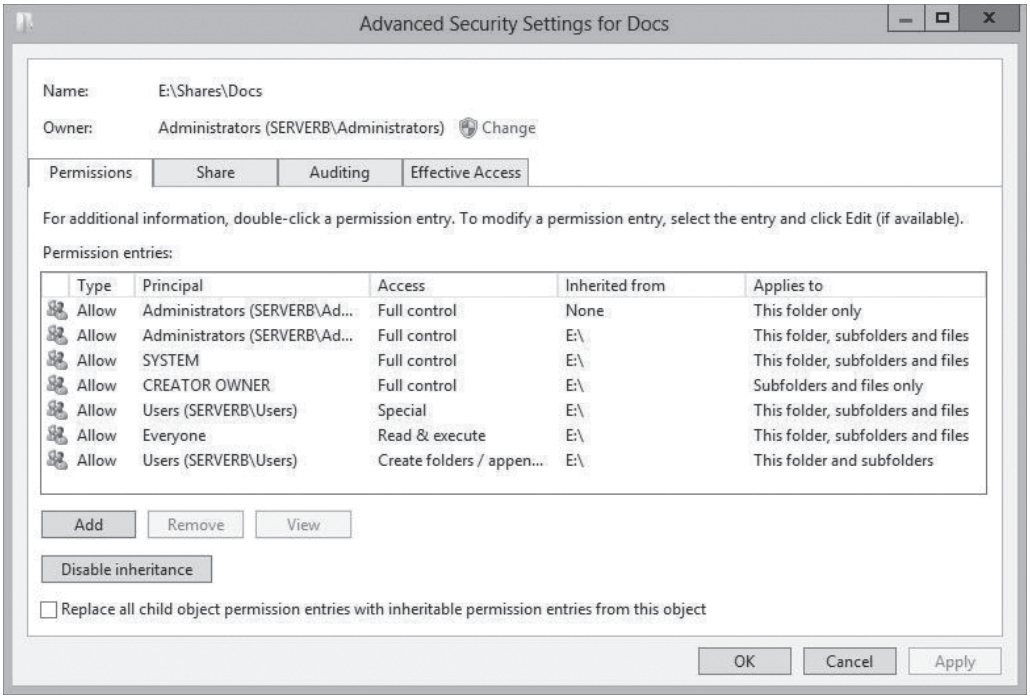
If this is all starting to sound extremely complex, don't worry. Windows provides preconfigured permission combinations suitable for most common access control chores. When you open the Properties sheet for a system element and look at its *Security* tab, the NTFS permissions you see are called ***basic permissions***. Basic permissions are actually combinations of ***advanced permissions***, which provide the most granular control over the element.

For example, the NTFS permission system has 14 advanced permissions that you can assign to a folder or file. However, it also has 6 basic permissions that are various combinations of the 14 advanced permissions. You can also assign both types of permissions in a single ACE, combining a basic permission with one or more advanced permissions, to create a customized combination. In most cases, however, you work only with basic permissions. Many administrators rarely, if ever, work directly with advanced permissions.

If you do find it necessary to work with advanced permissions directly, Windows makes it possible. After you click the *Advanced* button on the *Security* tab of any Properties sheet, you access the ACEs for the selected system element directly through an *Advanced Security Settings* dialog box (see Figure 4-5). System Manager provides access to the same dialog box through a share's Properties sheet.

Figure 4-5

The *Advanced Security Settings* dialog box



Setting Share Permissions

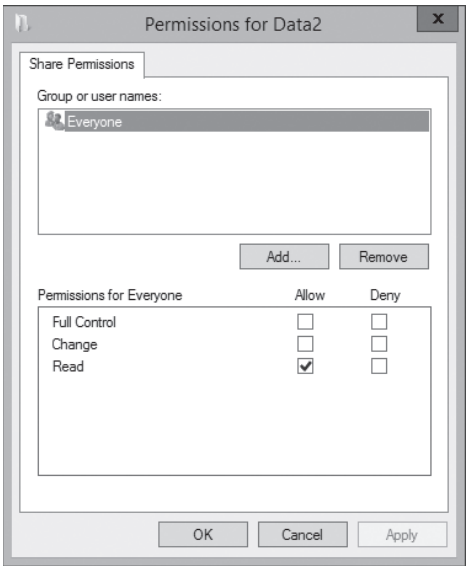
On Windows Server 2012 R2, shared folders have their own permission system, which is completely independent from the other Windows permission systems.

For network users to access shares on a file server, you must grant them the appropriate share permissions. By default, the Everyone special identity receives the Allow Read share permission to any new shares you create using File Explorer. In shares you create using Server Manager, the Everyone special identity receives the Allow Full Control share permission.

To modify the share permissions for an existing share via File Explorer, you open the Properties sheet for the shared folder, select the *Sharing* tab, and click *Advanced Sharing* and then *Permissions* to open the *Share Permissions* tab, as shown in Figure 4-6.

Figure 4-6

The Share Permissions tab for a shared folder



By using this interface, you can add security principals and allow or deny them the three share permissions listed in Table 4-1.

Table 4-1

Share Permissions and their functions

Share Permission	Allows or Denies Security Principals the Ability to:
Full Control	• Change file permissions • Take ownership of files • Perform all tasks allowed by the Change permission
Change	• Create folders • Add files to folders • Change data in files • Change file attributes • Delete folders and files • Perform all actions permitted by the Read permission
Read	• Display folder names, filenames, file data and attributes • Execute program files • Access other folders within the shared folder

To set share permissions via Server Manager while creating a share or modifying an existing one, use the following procedure.

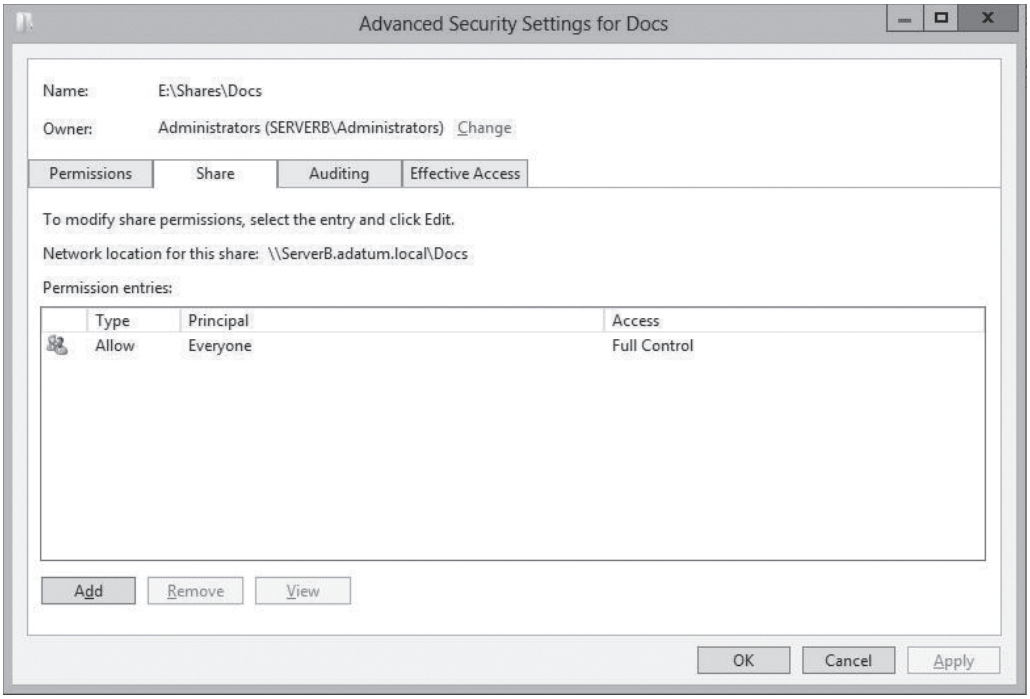
SET SHARE PERMISSIONS

GET READY. Log on to Windows Server 2012 R2, using an account with domain administrative privileges.

1. In the *Server Manager* window, click the File and Storage Services icon. In the submenu, click Shares. The *Shares* homepage appears.
2. In the *Shares* tile, right-click a share and, from the context menu, select Properties. The Properties sheet for the share appears.
3. Click Permissions. The *Permissions* page appears.
4. Click Customize Permissions. The *Advanced Security Settings* dialog box for the share appears.
5. Click the Share tab to display the interface shown in Figure 4-7.

Figure 4-7

The Share tab of the Advanced Security Settings dialog box for a share in Server Manager



6. Click Add. A *Permission Entry* dialog box for the share appears.
7. Click the Select a principal link to display the *Select User, Computer, Service Account, or Group* dialog box.
8. Type the name of or search for the security principal to which you want to assign share permissions and click OK. The *Permission Entry* dialog box displays the security principal you specified.
9. Select the type of permissions you want to assign (Allow or Deny).
10. Select the check boxes for the permissions you want to assign and click OK. The *Advanced Security Settings* dialog box displays the new access control entry you just created.
11. Click OK to close the *Advanced Security Settings* dialog box.
12. Click OK to close the share's Properties sheet.

CLOSE the *Server Manager* window.

When assigning share permissions, you must be aware that they do not combine like NTFS permissions. If you grant Alice the Allow Read and Allow Change permissions to the shared C:\Documents\Alice folder and later deny her all three permissions to the shared C:\Documents folder, the Deny permissions prevent her from accessing any files through the C:\Documents share, including those in the C:\Documents\Alice folder. However, she can still access her files through the C:\Documents\Alice share because of the Allow permissions. In other words, the C:\Documents\Alice share does not inherit the Deny permissions from the C:\Documents share.

Understanding NTFS Authorization

Most Windows installations today use the NTFS and ReFS file systems, as opposed to FAT32. One main advantage of NTFS and ReFS is that they support permissions, which FAT32 does not.

As described earlier in this lesson, every file and folder on an NTFS or ReFS drive has an ACL that consists of ACEs, each of which contains a security principal and the permissions assigned to that principal.

In the NTFS permission system, which ReFS also supports, the security principals involved are users and groups, which Windows refers to using ***security identifiers (SIDs)***. When a user attempts to access an NTFS file or folder, the system reads the user's security access token, which contains the SIDs for the user's account and all groups to which the user belongs. The system then compares these SIDs to those stored in the file or folder's ACEs, to determine what access the user should have. This process is called ***authorization***.

Assigning Basic NTFS Permissions

Most file server administrators work with basic NTFS permissions almost exclusively because they do not need to work directly with advanced permissions for most common access-control tasks.

Table 4-2 lists the basic permissions that you can assign to NTFS files or folders, and the capabilities that they grant to their possessors.

Table 4-2

NTFS Basic Permissions

Standard Permission	When Applied to a Folder, Enables a Security Principal to:	When Applied to a File, Enables a Security Principal to:
Full Control	• Modify the folder permissions • Take ownership of the folder • Delete subfolders and files contained in the folder • Perform all actions associated with all the other NTFS file permissions	• Modify the file permissions • Take ownership of the file • Perform all actions associated with all the other NTFS folder permissions
Modify	• Delete the folder • Perform all actions associated with the Write and the Read & Execute permissions	• Modify the file • Delete the file • Perform all actions associated with the Write and the Read & Execute permissions
Read and Execute	• Navigate through restricted folders to reach other files and folders • Perform all actions associated with the Read and List Folder Contents permissions	• Perform all actions associated with the Read permission • Run applications
List Folder Contents	• View the names of the files and subfolders contained in the folder	• Not applicable
Read	• See the files and subfolders contained in the folder • View the folder's ownership, permissions, and attributes	• Read the file contents • View the file's ownership, permissions, and attributes
Write	• Create new files and subfolders inside the folder • Modify the folder attributes • View the folder's ownership and permissions	• Overwrite the file • Modify the file attributes • View the file's ownership and permissions

To assign basic NTFS permissions to a shared folder, the options are essentially the same as with share permissions. You can open the folder's Properties sheet in File Explorer and select the *Security* tab, or you can open a share's Properties sheet in Server Manager, as in the following procedure.

ASSIGN BASIC NTFS PERMISSIONS

GET READY. Log on to Windows Server 2012 R2, using an account with domain administrative privileges. The Server Manager window appears.

1. Click the File and Storage Services icon and, in the submenu that appears, click Shares. The *Shares* homepage appears.
2. In the *Shares* tile, right-click a share and, from the context menu, select Properties. The Properties sheet for the share appears.
3. Click Permissions. The *Permissions* page appears.
4. Click Customize Permissions. The *Advanced Security Settings* dialog box for the share appears, displaying the *Permissions* tab, as shown in Figure 4-8. This dialog box is as close as the Windows graphical interface can come to displaying the contents of an ACL. Each line in the *Permission Entries* list is essentially an ACE and includes the following information:
   - *Type* specifies whether the entry allows or denies the permission.
   - *Principal* specifies the name of the user, group, or device principal receiving the permission.
   - *Access* specifies the name of the permission assigned to the security principal. If the entry is used to assign multiple advanced permissions, the word *Special* appears in this field.
   - *Inherited From* specifies whether the permission is inherited and, if so, from where it is inherited.
   - *Applies To* specifies whether the permission is to be inherited by subordinate objects and, if so, by which ones.

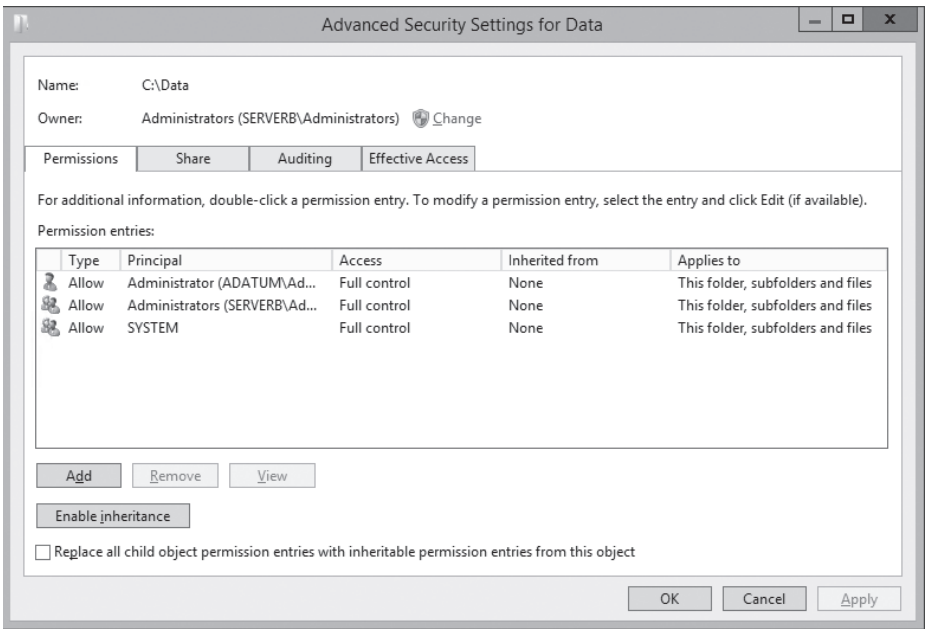


Figure 4-8

The *Advanced Security Settings* dialog box for a share in Server Manager

5. Click Add. A *Permission Entry* dialog box for the share appears.
6. Click the Select a principal link to display the *Select User, Computer, Service Account, or Group* dialog box.
7. Type the name of or search for the security principal to which you want to assign share permissions and click OK. The *Permission Entry* dialog box displays the security principal you specified.
8. From the Type drop-down list, select the type of permissions you want to assign (Allow or Deny).
9. From the Applies to drop-down list, specify which subfolders and files should inherit the permissions you are assigning.
10. Select the check boxes for the basic permissions you want to assign and click OK. The *Advanced Security Settings* dialog box displays the new access control entry you just created.
11. Click OK to close the *Advanced Security Settings* dialog box.
12. Click OK to close the Properties sheet.

CLOSE the *Server Manager* window.

■ Configuring Volume Shadow Copies

THE BOTTOM LINE

Volume Shadow Copies is a Windows Server 2012 R2 feature that enables you to maintain previous versions of files on a server, so that if users accidentally delete or overwrite a file, they can access a copy. You can implement shadow copies only for an entire volume; you cannot select specific shares, folders, or files.

To configure a Windows Server 2012 R2 volume to create shadow copies, use the following procedure.

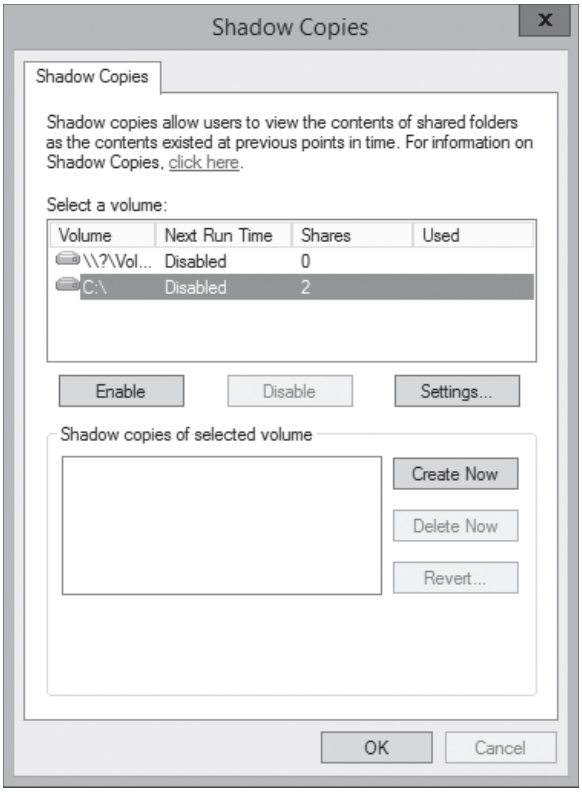
CONFIGURE SHADOW COPIES

GET READY. Log on to Windows Server 2012 R2, using an account with domain administrative privileges.

1. Click the File Explorer icon on the taskbar to display the *File Explorer* window.
2. In the Folders list, expand the Computer container, right-click a volume and, from the context menu, select Configure Shadow Copies. The *Shadow Copies* dialog box appears, as shown in Figure 4-9.

Figure 4-9

The Shadow Copies dialog box



3. In the Select a Volume box, choose the volume for which you want to enable shadow copies. By default, when you enable shadow copies for a volume, the system uses the following settings:
   - It stores the shadow copies on the selected volume.
   - It reserves a minimum of 300 MB of disk space for the shadow copies.
   - It creates shadow copies at 7:00 AM and 12:00 PM every weekday.
4. To modify the default parameters, click Settings. The *Settings* dialog box appears.
5. In the Storage Area box, specify the volume where you want to store the shadow copies. For a server operating with a high I/O load, such as a file server, Microsoft recommends that, for best performance, you create the Shadow Copies storage area on another volume, one that does not have Shadow Copies enabled. However, some third-party backup utilities require shadow copies to be stored on the same volume as the data.
6. Specify the Maximum Size for the storage area or choose the No Limit option. If the storage area fills up, the system begins deleting the oldest shadow copies, so if many large files are stored on the volume, increasing the size of the storage area can be beneficial. However, no matter how much space you allocate to the storage area, Windows Server 2012 R2 supports a maximum of 64 shadow copies for each volume, after which the system begins deleting the oldest copies.
7. Click Schedule. The *Schedule* dialog box appears. By using the controls provided, you can modify the existing shadow copies tasks, delete them, or create new ones, based on your users' needs. Scheduling shadow copies to occur too frequently can degrade server performance and cause copies to be aged out too quickly, whereas scheduling them to occur too infrequently can cause users to lose work because the most recent copy is too old.

8. Click OK twice to close the *Schedule* and *Settings* dialog boxes.
9. Click Enable. The system enables the Shadow Copies feature for the selected volume and creates the first copy in the designated storage area.

CLOSE File Explorer.

After you complete this procedure, users can restore previous versions of files on the selected volumes from the *Previous Versions* tab on any file or folder's Properties sheet.

■ Configuring NTFS Quotas

THE BOTTOM LINE

Managing disk space is a constant concern for server administrators. One way to prevent users from monopolizing large amount of storage is to implement quotas. Windows Server 2012 R2 supports two types of storage quotas. The more elaborate of the two is implemented as part of File Server Resource Manager. The second, simpler option is NTFS quotas.

NTFS quotas enable you to set a storage limit for users of a particular volume. Depending on how you configure the quota, users exceeding the limit can be denied disk space or just receive a warning. The space consumed by individuals users is measured by the size of the files they own or create.

NTFS quotas are relatively limited in that you can set only a single limit for all users of a volume. The feature is also limited in the actions it can take in response to a user exceeding the limit. The quotas in File Server Resource Manager, by contrast, are much more flexible in the nature of the limits you can set and the responses of the program, which can send e-mail notifications, execute commands, and generate reports, as well as log events.

To configure NTFS quotas for a volume, use the following procedure.

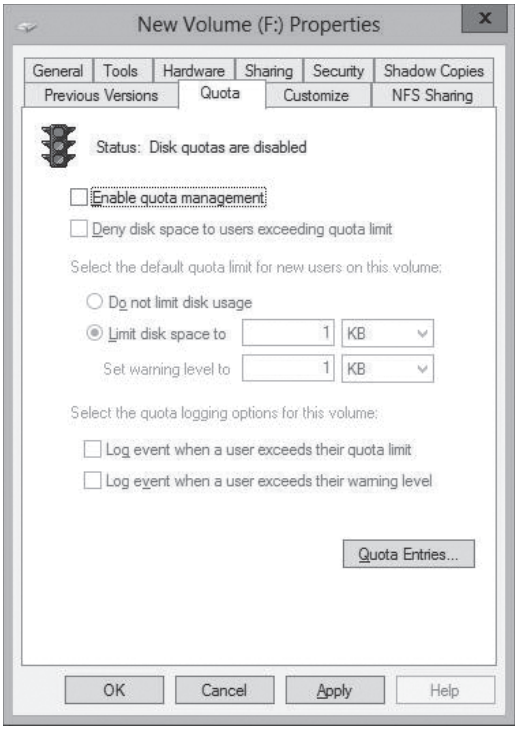
CONFIGURE NTFS QUOTAS

GET READY. Log on to Windows Server 2012 R2, using an account with domain administrative privileges.

1. Click the File Explorer icon in the taskbar. The *File Explorer* window appears.
2. In the Folders list, expand the Computer container, right-click a volume and, from the context menu, select Properties. The Properties sheet for the volume appears.
3. Click the Quota tab to display the interface shown in Figure 4-10.

Figure 4-10

The Quota tab of a volume's Properties sheet



4. Select the Enable quota management check box to activate the rest of the controls.
5. If you want to prevent users from consuming more than their quota of disk space, select the Deny disk space to users exceeding quota limit check box.
6. Select the Limit disk space to radio button and specify amounts for the quota limit and the warning level.
7. Select the Log event check boxes to control whether users exceeding the specified limits should trigger log entries.
8. Click OK to create the quota and close the Properties sheet.

CLOSE Windows Explorer.

■ Configuring Work Folders

THE BOTTOM LINE

Work Folders is a Windows Server 2012 R2 feature that enables administrators to provide their users with synchronized access to their files on multiple workstations and devices, while storing them on a network file server. The principle is roughly the same as Microsoft's OneDrive service, except that the files are stored on a private Windows server, instead of a cloud server on the Internet. This enables administrators to maintain control over the files, backing them up, classifying them, and/or encrypting them as needed.

To set up the Work Folders environment, you install the Work Folders role service under the File and Storage Services role on a server running Windows Server 2012 R2. This installs the File Server role service, if it is not installed already, and the IIS Hostable Web Core feature, which makes it possible for the server to respond to incoming HTTP requests from Work Folders clients on the network. You then create a new type of share called a sync share.

On the client side, you configure Work Folders in the Windows 8.1 Control Panel, specifying the email address of the user and the location of the Work Folders on the local disk. The system also creates a system folder called Work Folders, which appears in File Explorer and in file management dialogs. When the user saves files to the Work Folders on the client system, they are automatically synchronized with the user's folder on the Work Folders server.

Users can configure as many Work Folders clients as they need on different computers or other devices. After saving files to their Work Folders on their office workstations, for example, users can go home and find those files already synchronized to their home computers. In the same way, Work Folders can synchronize a user's files to a portable device at the office, and the user can work on them while offline during the commute home. Arriving home and connecting to the Internet, the device synchronizes the files back to the server , so that the user finds the latest versions on the office computer the next day.

Work Folders is not designed to be a collaborative tool; it is just a means synchronizing folders between multiple devices, while enabling administrators to retain control over them. It is possible to specify that Work Folders files remain encrypted during synchronization, and administrators can impose secuity policies that force the use of lock screens and mandatory data wipes for lost machines.

■ Business Case Scenarios

Scenario 4-1: Assigning Permissions

While you are working the help desk for a corporate network, a user named Leo calls to request access to the files for Trinity, a new classified project. The Trinity files are stored in a shared folder on a Windows Server 2012 R2 workgroup file server, which is locked in a secured underground data storage facility in New Mexico. After verifying that he has the appropriate security clearance for the project, you create a new group on the file server called TRINITY_USERS and add Leo's user account to that group. Then, you add the TRINITY_USER group to the access control list for the Trinity folder on the file server, and assign the group the following NTFS permissions:

- Allow Modify
- Allow Read & Execute
- Allow List Folder Contents
- Allow Read
- Allow Write

Sometime later, Leo calls you to tell you that he is able to access the Trinity folder and read the files stored there, but he has been unable to save changes back to the server. What is the most likely cause of the problem?

Scenario 4-2: Accessing Orphaned Files

Libby, a new hire in the IT department, approaches you, her supervisor, ashen-faced. A few minutes earlier, the president of the company called the help desk and asked Libby to give his new assistant the permissions needed to access his personal budget spreadsheet. As she was attempting to assign the permissions, she accidentally deleted the BUDGET_USERS group from the spreadsheet's access control list. Libby is terrified because that group was the only entry in the file's ACL. Now, no one can access the spreadsheet file, not even the president or the Administrator account. Is there any way to gain access to the file, and if so, how?

Configuring Print and Document Services

LESSON 5

■ Deploying a Print Server

THE BOTTOM LINE

Like the file-sharing functions discussed in previous lessons, print device sharing is one of the most basic applications for which local area networks were designed.

Installing, sharing, monitoring, and managing a single network print device is relatively simple, but when you are responsible for dozens or even hundreds of print devices on a large enterprise network, these tasks can be overwhelming.

Understanding the Windows Print Architecture

You need to understand the terms that Microsoft uses when referring to the various components of the network printing architecture.

Printing in Microsoft Windows typically involves these four components:

- A ***print device*** is the actual hardware that produces hard-copy documents on paper or other print media. Windows Server 2012 R2 supports both *local print devices* directly attached to computer ports and *network interface print devices* connected to the network, either directly or through another computer.
- In Windows, a ***printer*** is the software interface through which a computer communicates with a print device. Windows Server 2012 R2 supports numerous physical interfaces, including Universal Serial Bus (USB), IEEE 1394 (FireWire), parallel (LPT), serial (COM), Infrared Data Access (IrDA), Bluetooth ports, and network printing services such as lpr, Internet Printing Protocol (IPP), and standard TCP/IP ports.
- A ***print server*** is a computer (or standalone device) that receives print jobs from clients and sends them to print devices locally attached or connected to the network.
- A ***printer driver*** is a device driver that converts the print jobs generated by applications into an appropriate string of commands for a specific print device. Printer drivers are designed for specific print devices and provide applications with access to all a print device's features.

TAKE NOTE*

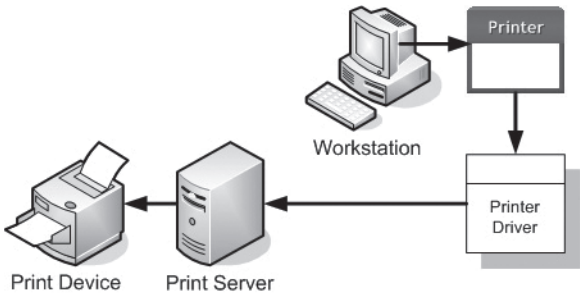
Printer and *print device* are the most commonly misused terms of the Windows printing vocabulary. Obviously, many sources use *printer* to refer to the printing hardware. However, in Windows, printer and print device are not equivalents. For example, you can add a printer to a Windows Server 2012 R2 computer without a physical print device being present. The computer can then host the printer, print server, and printer driver. These three components enable the computer to process the print jobs and store them in a print queue until the print device is available.

UNDERSTANDING WINDOWS PRINTING

These four components—print device, printer, print server, and printer driver—work together to process the print jobs produced by Windows applications and turn them into hard-copy documents, as shown in Figure 5-1.

Figure 5-1

The Windows print architecture



Before you can print documents in Windows, you must install at least one printer. To install a printer in Windows, you must do the following:

- Select the print device's specific manufacturer and model.
- Specify the port (or other interface) the computer will use to access the print device.
- Supply a printer driver specifically created for that print device.

When you print a document in an application, you select the destination printer for the print job.

The printer is associated with a printer driver that takes the commands generated by the application and converts them into a ***printer control language (PCL)***, a language understood by the printer. PCLs can be standardized, like the PostScript language, or they can be proprietary languages developed by the print device manufacturer.

The printer driver enables you to configure the print job to use the print device's various capabilities. These capabilities are typically incorporated into the printer's Properties sheet. For example, your word-processing application does not know if your print device is color, monochrome, or supports duplex printing; the printer driver provides support for print device features such as these.

After the printer processes a print job, it stores the job in a print queue, known as a ***spooler***. Depending on the arrangement of the printing components, the spooled jobs might be in a PCL format, ready to go to the print device, or in an interim format, in which case the printer driver must process the spooled jobs into the PCL format before sending them to the device. If other jobs are waiting to be printed, a new job might wait in the spooler for some time. When the server finally sends the job to the print device, the device reads the PCL commands and produces the hard-copy document.

Sharing a Printer

Using Windows Server 2012 R2 as a print server can be simple or complex, depending on how many clients the server has to support and how much printing they do.

For a home or small business network, in which a handful of users need occasional access to the printer, no special preparation is necessary. However, if the computer must support heavy printer use, you might need the following hardware upgrades:

- **Additional system memory:** Processing print jobs requires system memory, just like any other application. If you plan to run heavy print traffic through a Windows Server 2012 R2 server, in addition to other roles or applications, make sure that the computer has sufficient memory to support all its functions.
- **Additional disk space:** When a print device is busy, the print server spools additional incoming print jobs temporarily on a hard drive until the print device is free to receive them. Depending on the amount of print traffic and the types of print jobs, the print server might require a substantial amount of temporary storage for this purpose.
- **Make the computer a dedicated print server:** In addition to memory and disk space, using Windows Server 2012 R2 as a print server requires processor clock cycles, just like any other application. On a server handling heavy print traffic, other roles and applications are likely to experience substantial performance degradation. If you need a print server to handle heavy traffic, consider dedicating the computer to print server tasks only and deploying other roles and applications elsewhere.

On a Windows Server 2012 R2 computer, you can share a printer as you are installing it or at any time afterward. On older printers, initiate the installation process by launching the *Add Printer Wizard* from the *Devices and Printers* control panel. However, most of the print devices on the market today use either a USB connection to a computer or an Ethernet or wireless connection to a network.

In the case of a USB-connected printer, you plug the print device into a USB port on the computer and turn on the device to initiate the installation process. Manual intervention is only required when Windows Server 2012 R2 does not have a driver for the print device.

For network-attached print devices, an installation program supplied with the product locates the print device on the network, installs the correct drivers, creates a printer on the computer, and configures the printer with the proper IP address and other settings.

After you install the printer on the Windows Server 2012 R2 computer that functions as your print server, you can share it with your network clients.

Managing Printer Drivers

Printer driver components enable your computers to manage the capabilities of your print devices. When you install a printer on a server running Windows Server 2012 R2, you install a driver that other Windows computers also can use.

The printer drivers you install on Windows Server 2012 R2 are the same drivers that Windows workstations and other server versions use, with one stipulation. As a 64-bit platform, Windows Server 2012 R2 uses 64-bit device drivers, which are suitable for other computers running 64-bit versions of Windows. If you have 32-bit Windows systems on your network, however, you must install a 32-bit driver on the server for those systems to use.

The *Additional Drivers* dialog box, accessible from the *Sharing* tab of a printer's Properties sheet, enables you to install drivers for other processor platforms. However, you must install those drivers from a computer running on the alternative platform.

In other words, to install a 32-bit driver for a printer on a server running Windows Server 2012 R2, you must access the printer's Properties sheet from a computer running 32-bit version of Windows. You can do this by accessing the printer directly through the network using Windows Explorer, or by running the Print Management snap-in on the 32-bit system and using it to manage your Windows Server 2012 R2 print server.

Using Remote Access Easy Print

When a Remote Desktop Services client connects to a server, it runs applications using the server's processor(s) and memory. However, if that client wants to print a document from one of those applications, it wants the print job to go to the print device connected to the client computer.

Remote Desktop Easy Print is the component that enables Remote Desktop clients to print to their local print devices. Easy Print takes the form of a printer driver installed on the server, along with the Remote Desktop Session Host role service.

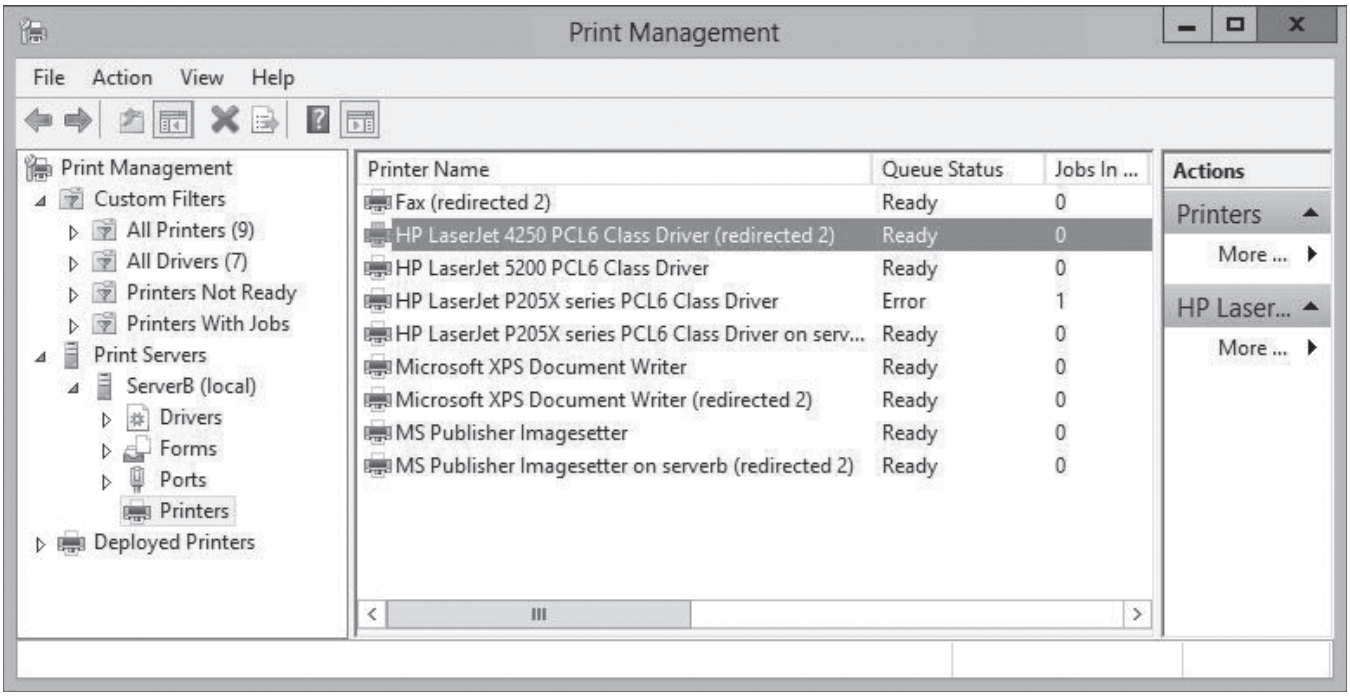
The Remote Desktop Easy Print driver appears in the Print Management snap-in automatically but it is not associated with a particular print device. Instead, the driver functions as a redirector, enabling the server to access the printers on the connected clients.

On Windows Server 2012 R2, Easy Print requires no configuration other than the allowance of Remote Desktop connections or the installation of the Remote Desktop Services role. However, as soon as it is operational, it provides the server administrator with additional access to the printers on the Remote Desktop clients.

When a Remote Desktop client connects to a server via the Remote Desktop Connection program or the RD Web Access site, the printers installed on the client system are redirected to the server and appear in the Print Management snap-in as redirected server printers, as shown in Figure 5-2.

Figure 5-2

Printers redirected by Easy Print on a Remote Desktop server



A client running an application on the server can therefore print to a local print device via the redirected printer. You can also open the Properties sheet for the redirected printer in the usual manner and manipulate its settings.

Configuring Printer Security

Like folder shares, clients must have the proper permissions to access a shared printer.

Printer permissions are much simpler than NTFS permissions; they dictate whether users are allowed to use the printer, manage documents submitted to the printer, or manage the properties of the printer itself. To assign permissions for a printer, use the following procedure.

ASSIGN PRINTER PERMISSIONS

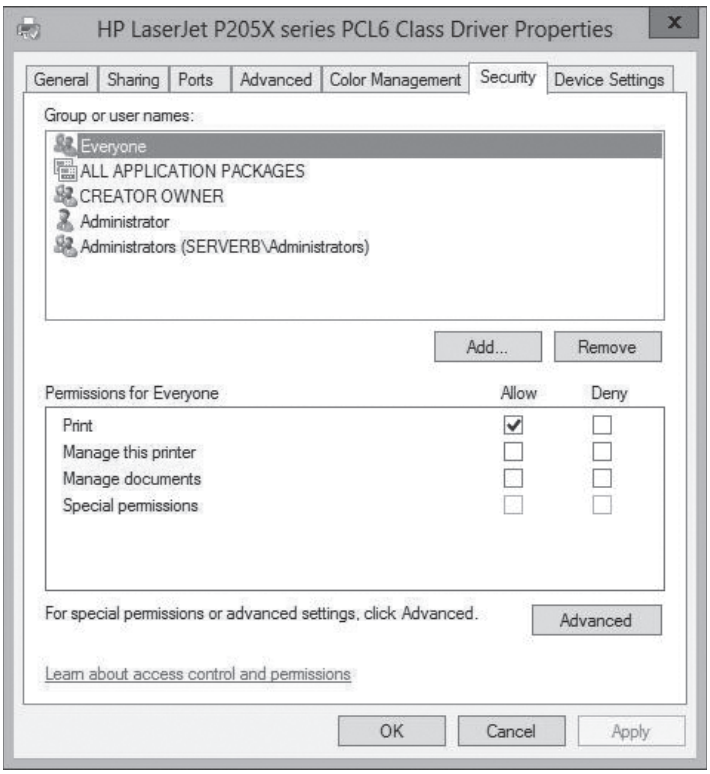
GET READY. Log on to Windows Server 2012 R2 using a domain account with Administrator privileges.

1. Open the Control Panel and select Hardware> Devices and Printers. The *Devices and Printers* window appears.
2. Right-click one of the printer icons in the window and, from the context menu, select Printer Properties. The printer's Properties sheet appears.

   Click the Security tab, as shown in Figure 5-3. The top half of the display lists all the security principals now possessing permissions to the selected printer. The bottom half lists the permissions held by the selected security principal.

Figure 5-3

The Security tab of a printer's Properties sheet



3. Click Add. The *Select Users, Computers, Service Accounts, or Groups* dialog box appears.

TAKE NOTE*

This procedure assumes that the Windows Server 2012 R2 computer is a member of an Active Directory domain. When you assign printer permissions on a standalone server, you select local user and group accounts to be the security principals that receive the permissions.

4. In the Enter the object names to select text box, type a user or group name, and then click OK. The user or group appears in the *Group or user names* list of the printer's properties sheet.
5. Select the security principal you added, and then select or clear the check boxes in the bottom half of the properties sheet to Allow or Deny the user any of the basic permissions.
6. Click OK to close the Properties sheet.

CLOSE the control panel.

Like NTFS permissions, printer permissions come in two types: basic and advanced. Each of the three basic permissions consists of a combination of advanced permissions, as listed in Table 5-1.

Table 5-1
Basic Printer Permissions

Permission	Capabilities	Advanced Permissions	Default Assignments
Print	• Connect to a printer • Print documents • Pause, resume, restart, and cancel the user's own documents	• Print • Read Permissions	Assigned to the Everyone special identity
Manage this printer	• Cancel all documents • Share a printer • Change printer properties • Delete a printer • Change printer permissions	• Print • Manage Printers • Read Permissions • Change Permissions • Take Ownership	Assigned to the Administrators group
Manage documents	• Pause, resume, restart, and cancel all users' documents • Control job settings for all documents	• Manage Documents • Read Permissions • Change Permissions • Take Ownership	Assigned to the Creator Owner special identity

Managing Printers

Users with the Allow Manage This Printer permission can go beyond manipulating queued documents; they can reconfigure the printer itself. Managing a printer refers to altering the operational parameters that affect all users and controlling access to the printer.

Generally, most software-based tasks that fall under the category of managing a printer are those you perform once while setting up the printer for the first time. Day-to-day printer management is more likely to involve physical maintenance, such as clearing print jams, reloading paper, and changing toner or ink cartridges. However, the following sections examine some of the printer manager's typical configuration tasks.

SETTING PRINTER PRIORITIES

In some cases, administrators with the Manage This Printer permission might want to give certain users in your organization priority access to a print device so that when print traffic is heavy, their jobs are processed before those of other users. To do this, you must create multiple printers, associate them with the same print device, and then modify their priorities, as described in the following procedure.

SET A PRINTER'S PRIORITY

GET READY. Log on to Windows Server 2012 R2 using an account with the Manage This Printer permission.

1. Open the Control Panel and select Hardware> Devices and Printers. The *Devices and Printers* window appears.
2. Right-click one of the printer icons and then, from the context menu, select Printer Properties. The Properties sheet for the printer appears.
3. On the *Advanced* tab set the Priority spin box to a number representing the highest priority you want to set for the printer. Higher numbers represent higher priorities. The highest possible priority is 99.
4. On the *Security* tab, add the users or groups that you want to provide with high-priority access to the printer and assign the Allow Print permission to them.
5. Revoke the Allow Print permission from the Everyone special identity.
6. Click OK to close the Properties sheet.
7. Create an identical printer, using the same printer driver and pointing to the same print device. Leave the Priority setting to its default value of 1 and leave the default permissions in place.
8. Rename the printers, specifying the priority assigned to each one.

CLOSE the control panel.

Inform the privileged users that they should send their jobs to the high-priority printer. All jobs sent to that printer are processed before those sent to the other, lower-priority printer.

SCHEDULING PRINTER ACCESS

Sometimes, you might want to limit certain users' access to a printer to specific times of the day or night. For example, your organization might have a color laser printer that the company's graphic designers use during business hours, but you permit other employees to use it after 5:00 PM. To do this, you associate multiple printers with a single print device, much as you did to set different printer priorities.

After creating two printers, both pointing to the same print device, you configure their scheduling using the following procedure.

SCHEDULE PRINTER ACCESS

GET READY. Log on to Windows Server 2012 R2 using an account with the Manage Printer permission. When the logon process is completed, close the *Initial Configuration Tasks* window and any other windows that open.

1. Open the Control Panel and select Hardware> Devices and Printers. The *Devices and Printers* window appears.
2. Right-click one of the printer icons and then, from the context menu, select Printer Properties. The Properties sheet for the printer appears.
3. On the *Advanced* tab, select the Available from radio button and then, in the two spin boxes provided, select the range of hours you want the printer to be available.
4. On the Security tab, add the users or groups that you want to provide with access to the printer during the hours you selected and grant them the Allow Print permission.
5. Revoke the Allow Print permission from the Everyone special identity.
6. Click OK to close the Properties sheet.

CLOSE the control panel.

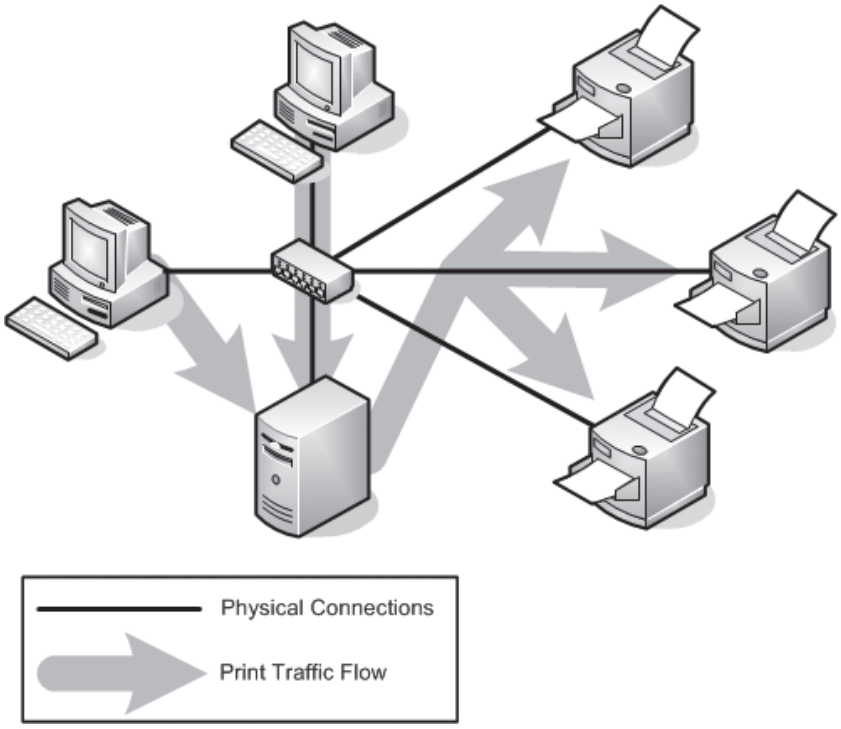
The two printers are now available only during the hours you have specified.

CREATING A PRINTER POOL

As mentioned earlier, a printer pool increases the production capability of a single printer by connecting it to multiple print devices. When you create a printer pool, the print server sends each incoming job to the first print device it finds that is not busy. This effectively distributes the jobs among the available print devices, as shown in Figure 5-4, providing users with more rapid service.

Figure 5-4

Printer pooling shares print jobs among multiple print devices



To configure a printer pool, use the following procedure.

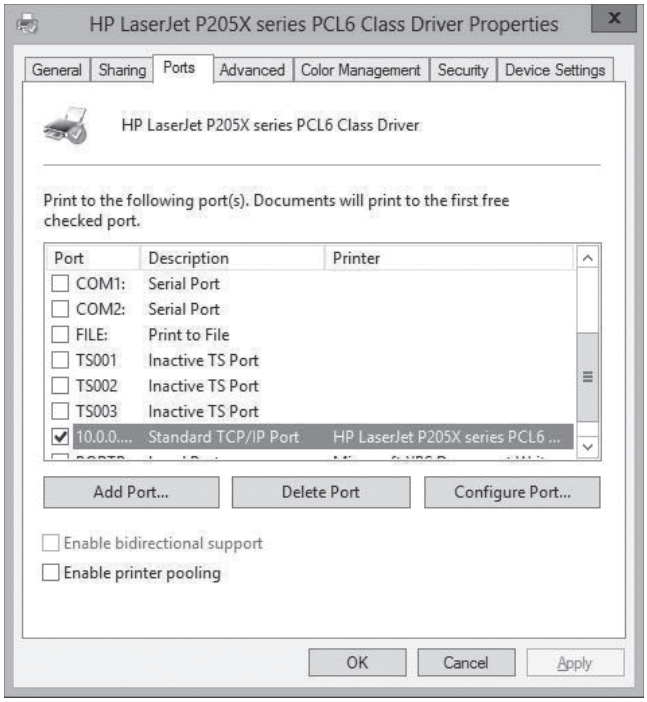
CREATE A PRINTER POOL

GET READY. Log on to Windows Server 2012 R2 using an account with the Manage Printer permission. When the logon process is completed, close the Initial *Configuration Tasks* window and any other windows that open.

1. Open the Control Panel and select Hardware> Devices and Printers. The *Devices and Printers* window appears.
2. Right-click one of the printer icons and then, from the context menu, select Printer Properties. The Properties sheet for the printer appears.
3. On the Ports tab, select all ports to which the print devices are connected (see Figure 5-5).

Figure 5-5

The Ports tab of a printer's Properties sheet



4. Select the Enable printer pooling check box, and then click OK.

CLOSE the control panel.

To create a printer pool, you must have at least two identical print devices, or at least print devices that use the same printer driver. The print devices must be in the same location, because you cannot tell which print device will process a given document. You must also connect all print devices in the pool to the same print server. If the print server is a Windows Server 2012 R2 computer, you can connect the print devices to any viable ports.

■ Using the Print and Document Services Role

THE BOTTOM LINE

All printer sharing and management capabilities discussed in the previous sections are available on any Windows Server 2012 R2 computer in its default installation configuration. However, installing the Print and Document Services role on the computer provides additional tools that are particularly useful to administrators involved with network printing on an enterprise scale.

When you install the Print and Document Services role using Server Manager's Add Roles and Features Wizard, a *Select role services* page appears, enabling you to select from the options listed in Table 5-2.

Table 5-2

Role Service Selections for the Print Services Role

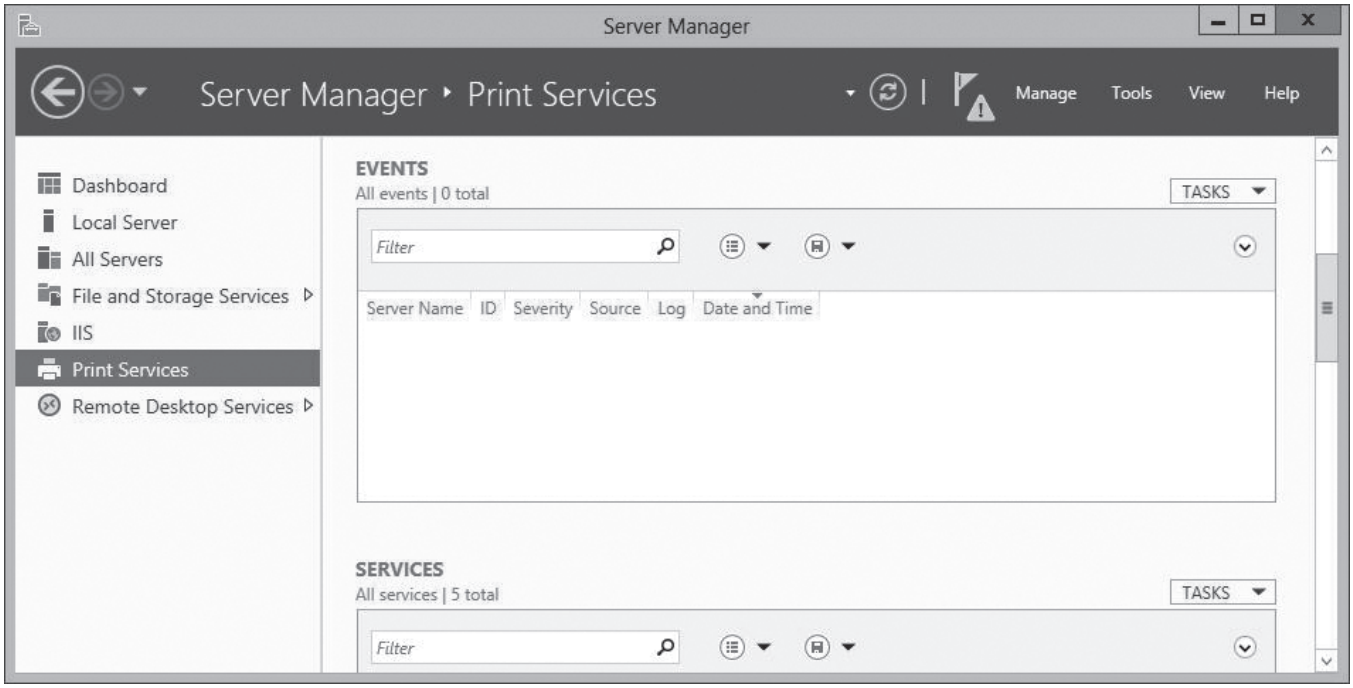
Role Service	System Services Installed	Description
Print Server	Print Spooler (Spooler)	• Installs the Print Management console for Microsoft Management Console (MMC), which enables administrators to deploy, monitor, and manage printers throughout the enterprise. • Is the only required role service when you add the Print Services role.
Distributed Scan Server	Distributed Scan Server (ScanServer)	• Enables the computer to receive documents from network-based scanners and forward them to the appropriate users.
Internet Printing	• World Wide Web Publishing Service (w3svc) • IIS Admin Service (iisadmin)	• Creates a website that enables users on the Internet to send print jobs to shared Windows printers.
LPD Service	TCP/IP Print Server (LPDSVC)	• Enables UNIX clients running the LPR (line printer remote) program to send their print jobs to Windows printers.

To install the Internet Printing role service, you must also install the Web Server (IIS) role with certain specific role services. The *Add Roles and Features Wizard* enforces these dependencies by displaying an *Add features that are required for Internet Printing?* message box when you select the Internet Printing role service. Clicking *Add Features* causes the wizard to select the exact role services within Web Server (IIS) role that the Internet Printing service needs.

As always, Windows Server 2012 R2 adds a new icon to the Server Manager navigation pane when you install a role. The Print Services homepage contains a filtered view of print-related event log entries, a status display for the role-related system services and role services, and performance counters, as shown in Figure 5-6.

Figure 5-6

The Print Services node in Server Manager



Using the Print Management Console

The Print Management snap-in for MMC, an administrative tool, consolidates the controls for the printing components throughout the enterprise into a single console. With this tool, you can access the print queues and Properties sheets for all network printers in the enterprise, deploy printers to client computers via Group Policy, and create custom views that simplify the process of detecting print devices that need attention due to errors or depleted consumables.

Windows Server 2012 installs the Print Management console when you add the Print and Document Services role to the computer. You can also install the console without the role by adding the Print and Document Services Tools feature, found under *Remote Server Administration Tools* > *Role Administration Tools* in Server Manager.

When you launch the Print Management console, the default display includes in the scope (left) pane the nodes listed in Table 5-3.

Table 5-3

Print Management Nodes

Node	Description
Custom Filters	Contains composite views of all printers hosted by the print servers listed in the console, regulated by customizable filters
Print Servers	Lists all print servers that you have added to the console, and all drivers, forms, ports, and printers for each print server
Deployed Printers	Lists all printers you have deployed with Group Policy through the console

The following sections demonstrate some administration tasks you can perform with the Print Management console.

ADDING PRINT SERVERS

By default, the Print Management console displays only the local machine in its list of print servers. Each print server has four nodes beneath it, listing the drivers, forms, ports, and printers associated with that server.

To manage other print servers and their printers, you must first add them to the console, using the following procedure.

ADD A PRINT SERVER

GET READY. Log on to Windows Server 2012 R2 using a domain account with Administrator privileges.

1. In the *Server Manager* window, click Tools > Print Management. The *Print Management* console appears.
2. Right-click the Print Servers node and, from the context menu, click Add/Remove Servers. The *Add/Remove Servers* dialog box appears.
3. In the *Specify Print Server* section, click Browse. The *Select Print Server* dialog box appears.
4. Select the print server you want to add to the console and click Select Server. The server you selected appears in the *Add servers* box in the *Add/Remove Servers* dialog box.
5. Click Add to List. The server you selected appears in the *Print Servers* list.
6. Click OK. The server appears under the *Print Servers* node.

CLOSE the Print Management console.

You can now manage the printers associated with the server you have added to the console.

VIEWING PRINTERS

One major problem for printing on large enterprise networks is keeping track of dozens or hundreds of print devices, all in frequent use, and all needing attention regularly. Whether the maintenance required is a major repair, replenishing ink or toner, or just filling the paper trays, print devices cannot get the attention they need until an administrator is aware of the problem.

The Print Management console provides many ways to view the printing components associated with the network's print servers. To create views, the console takes the complete list of printers and applies various filters to it, selecting which printers to display. Under the Custom Filters node are four default filters, as listed in Table 5-4.

Table 5-4
Default Filters

Filter	Description
All Printers	Contains a list of all the printers hosted by all the print servers added to the console
All Drivers	Contains a list of all printer drivers installed on all print servers added to the console
Printers Not Ready	Contains a list of all printers that are not reporting a Ready status
Printers With Jobs	Contains a list of all printers that currently have jobs waiting in the print queue

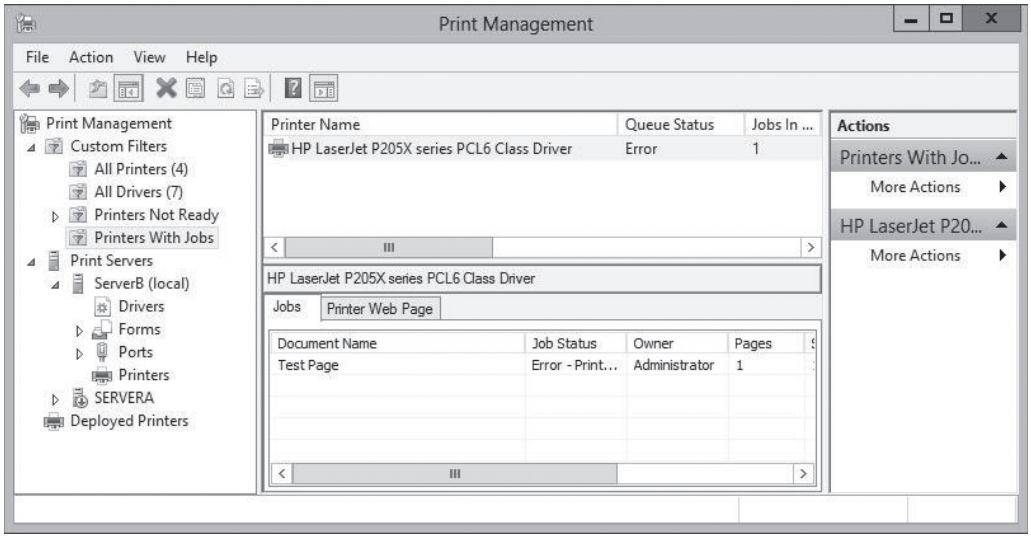
Views such as Printer Not Ready are a useful way for you to identify printers that need attention, without having to browse individual print servers or search through a long list of every printer on the network. In addition to these defaults, you can create your own custom filters.

MANAGING PRINTERS AND PRINT SERVERS

After you use filtered views to isolate the printers you want to examine, selecting a printer displays its status, the number of jobs currently in its print queue, and the name of the print server hosting it. If you right-click the filter in the scope pane and, from the context menu, select *Show Extended View*, an additional pane appears containing the contents of the selected printer's queue (see Figure 5-7). You can manipulate the queued jobs just as you would from the print queue window on the print server console.

Figure 5-7

The Print Management console's extended view



The Print Management console also enables you to access the configuration interface for any printer or print server listed in any of its displays. Right-clicking a printer or print server anywhere in the console interface, and selecting *Properties* from the context menu, displays the same Properties sheet that you would see on the print server computer itself. You can then configure printers and print servers without having to travel to the site of the print server or establish a Remote Desktop connection to the print server.

DEPLOYING PRINTERS WITH GROUP POLICY

Configuring a print client to access a shared printer is a simple matter of browsing the network or the AD DS tree and selecting the printer. However, when you have to configure hundreds or thousands of print clients, the task becomes more complicated. AD DS helps simplify the process of deploying printers to large numbers of clients.

Publishing printers in the AD DS database enables users and administrators to search for printers by name, location, or model (if you populate the *Location* and *Model* fields in the printer object). To create a printer object in the AD DS database, you can either select the *List in the directory* check box while sharing the printer, or right-click a printer in the *Print Management* console and, from the context menu, select *List in Directory*.

To use AD DS to deploy printers to clients, you must configure the appropriate policies in a Group Policy Object (GPO). You can link a GPO to any domain, site, or organizational unit (OU) in the AD DS tree. When you configure a GPO to deploy a printer, all users or computers in that domain, site, or OU receive the printer connection by default when they log on.

■ Business Case Scenarios

Scenario 5-1: Enhancing Print Performance

You are a desktop support technician for a law firm with a group of 10 legal secretaries who provide administrative support to the attorneys. The secretaries use a single, shared, high-speed laser printer connected to a dedicated Windows Server 2012 R2 print server. They regularly print multiple copies of large documents, and although the laser printer is fast, it runs constantly. Sometimes, the secretaries have to wait 20 minutes or more after submitting a print job for their documents to reach the top of the queue. The office manager has offered to purchase additional printers for the department. However, the secretaries are accustomed to simply clicking the Print button, and don't like the idea of having to examine multiple print queues to determine which one has the fewest jobs before submitting a document. What can you do to provide the department with a printing solution that will enable the secretaries to utilize additional printers most efficiently?

Scenario 5-2: Troubleshooting Printer Delays

One of your small business clients has a print device connected to a server running Windows Server 2012 R2. He has shared the printer so that the other network users can access it. Often, the other users print large documents that take a long time to print, but sometimes your client and other users have important documents that need to be printed before any long documents that are waiting in the printer queue. What would you suggest to this user?

6

LESSON

Configuring Servers for Remote Management

■ Using Server Manager for Remote Management

THE BOTTOM LINE

Windows Server 2012 R2 facilitates remote server management, so that you do not need to work at the server console. This capability conserves server resources that can be devoted to applications.

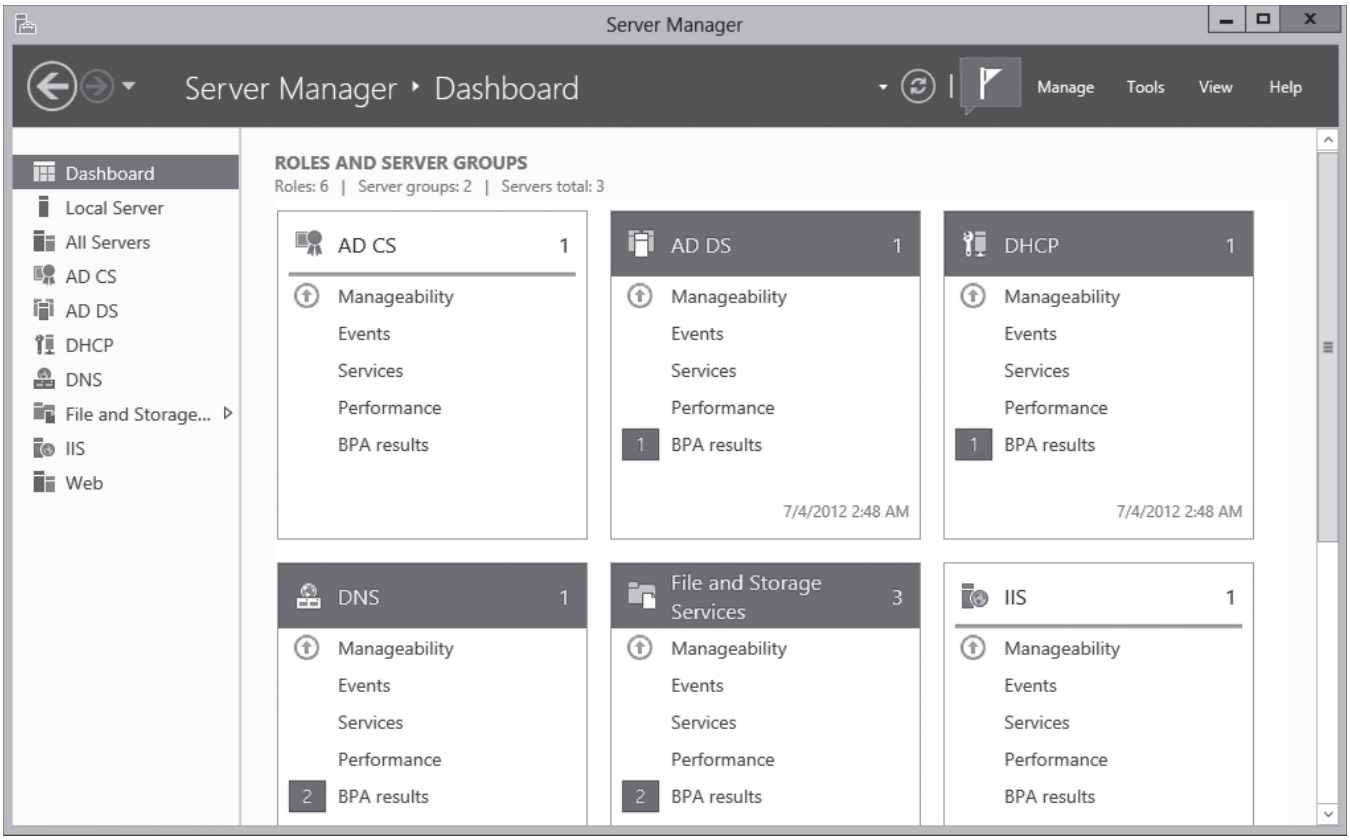
Since Windows Server 2003, Server Manager has been the primary server administration tool for Windows Server. The most obvious improvement to the Server Manager tool in Windows Server 2012 R2 is the capability to perform administrative tasks on remote servers, as well as on the local system.

After you log on to a GUI installation of Windows Server 2012 R2 with an administrative account, Server Manager loads automatically, displaying the Welcome tile.

The Server Manager interface contains a navigation pane on the left with icons representing various views of server resources. Selecting an icon displays a homepage in the right pane, which contains tiles with information about the resource. The Dashboard page, which appears by default, contains, in addition to the Welcome tile, thumbnails that summarize the other views available in Server Manager, as shown in Figure 6-1. The other views include Local Server, All Servers, and server groups and role groups.

Figure 6-1

Dashboard thumbnails in Server Manager



Adding Servers

The primary difference between the Windows Server 2012 R2 Server Manager and previous versions is the capability to add and manage multiple servers at once.

Although only the local server appears in Server Manager when you first run it, you can add other servers, enabling you to manage them together. The servers you add can be physical or virtual, and can run any version of Windows Server since Windows Server 2003. After you add servers to the interface, you can create groups containing collections of servers, such as the servers at a particular location or those performing a particular function. These groups appear in the navigation pane, enabling you to administer them as a single entity.

To add servers in Server Manager, use the following procedure.

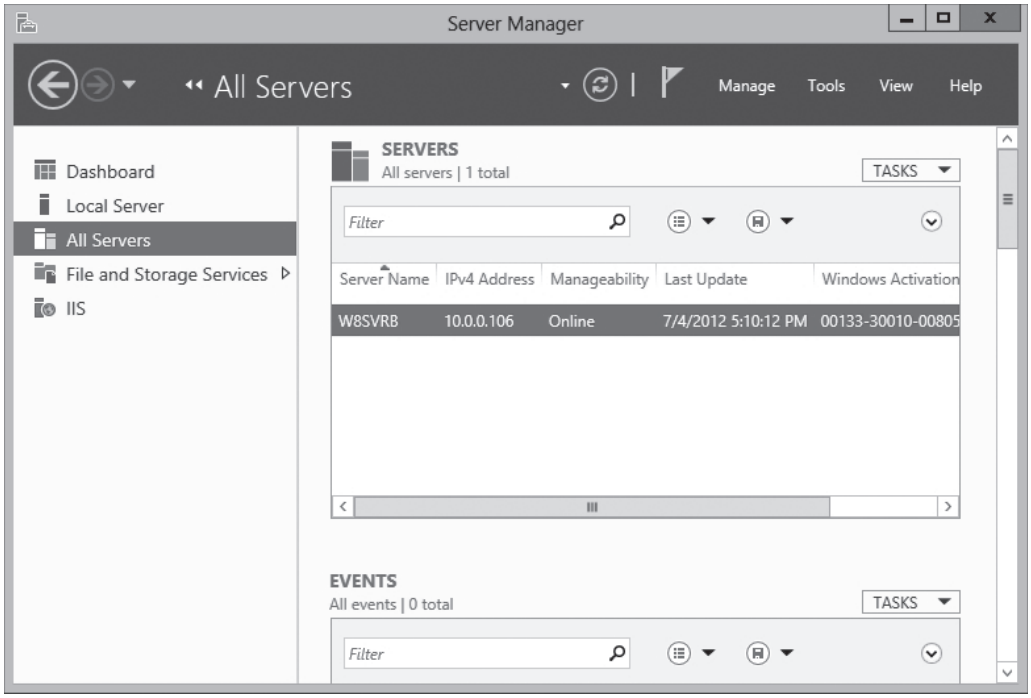
ADD SERVERS IN SERVER MANAGER

GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges. The Server Manager window appears.

1. In the navigation pane, click the All Servers icon. The All Servers homepage appears, as shown in Figure 6-2.

Figure 6-2

The All Servers homepage in Server Manager



2. From the Manage menu, select Add Servers. The Add Servers dialog box appears.
3. Select one of the following tabs to specify how you want to locate servers to add:
   - **Active Directory** enables you to search for computers running specific operating systems in specific locations in the local Active Directory Domain Services (AD DS) domain.
   - **DNS** enables you to search for servers in your configured Domain Name System (DNS) server.
   - **Import** enables you to supply a text file containing the names or IP addresses of the servers you want to add.
4. Initiate a search or upload a text file to display a list of available servers.
5. Select the servers you want to add and click the right arrow button to add them to the Selected list.
6. Click OK. The servers you selected are added to the All Servers homepage.

CLOSE the Server Manger console.

After you add remote servers to the Server Manager interface, they appear on the All Servers homepage. You can access them in various ways, depending on the Windows version the remote server is running.

ADDING WORKGROUP SERVERS

The procedure in the previous section assumes that the computer running Server Manager and the managed servers are members of an AD DS domain. You can usually add workgroup servers to Server Manager, but the system's attempts to access the remote servers fail with a "Credentials not valid" error.

Why is that? AD DS systems authenticate by using the Kerberos protocol, but Windows workgroup computers use an alternative authentication protocol called NTLM (NT LAN Manager). Essentially, the remote server tries to log on to the workgroup server and fails.

For the authentication to succeed, you must add the name of the workgroup server to the TrustedHosts list on the computer running Server Manager, by using a Windows PowerShell command with the following syntax:

```
Set-Item wsman:\localhost\Client\TrustedHosts
<servername> -Concatenate -Force
```

Managing Non-Domain Joined Servers

When you add servers that are members of an Active Directory Domain Services (AD DS) domain to the Server Manager interface, Windows Server 2012 R2 uses the standard Kerberos authentication protocol and your current logged on user credentials when connecting to the remote systems. You can also add servers that are not joined to an AD DS domain, but obviously, the system cannot authenticate using an AD DS account.

To manage a non-domain joined server using Server Manager and a server named MGMT01, you must first complete the following tasks:

- Supply administrative credentials for the non-domain joined server.
- Add the non-domain joined server to the MGMT01's WS-Management TrustedHosts list on the computer running Server Manager.

To add non-domain joined servers to Server Manager, you must use the DNS or Import option in the Add Servers Wizard. After creating the server entries, you must right-click each one and select Manage As from the context menu. This displays a Windows Security dialog box, in which you can supply credentials for an account with administrative privileges on the remote server.

Domain membership automatically establishes a trust relationship among the computers in the domain. To manage computers that are not in the same domain, you must establish that trust yourself by adding the computers you want to manage to the TrustedHosts list on the computer running Server Manager.

The TrustedHosts list exists on a logical drive called WSMan:; the path to the list itself is WSMan:\localhost\Client\TrustedHosts. To add a computer to the list, you use the Set-Item cmdlet in Windows PowerShell. After opening a Windows PowerShell session with administrative privileges on the computer running Server Manager, use the following command to add the servers you want to manage to the list:

```
Set-Item WSMan:\localhost\Client\TrustedHosts –value
<servername> -force
```

Managing Windows Server 2012 R2 Servers

After you add servers running Windows Server 2012 R2 to Server Manager, you can immediately use the Add Roles and Features Wizard to install roles and features on any server you add.

You can also perform other administrative tasks, such as configure NIC teaming and restart the server, because Windows Remote Management (WinRM) is enabled by default on Windows Server 2012 R2. ***WinRM*** is a Windows feature that enables administrators to execute management commands and scripts on remote computers, using a communications protocol called WS-Management Protocol.

CONFIGURING WINRM

WinRM enables you to manage a computer from a remote location using tools based on Windows Management Instrumentation (WMI) and Windows PowerShell. If the default WinRM setting has been modified, or if you want to change it manually, you can do so through the Server Manager interface.

On the Local Server homepage, the Properties tile contains a Remote management indicator that specifies the server's current WinRM status. To change the WinRM state, click the Remote management hyperlink to open the Configure Remote Management dialog box. Clearing the *Enable remote management of this server from other computers* check box disables WinRM, and selecting the check box enables it.

CONFIGURING WINDOWS FIREWALL

However, if you attempt to launch Microsoft Management Console (MMC) snap-ins targeting a remote server (such as the Computer Management console), you will receive an error because of the default Windows Firewall settings on Windows Server 2012 R2. MMC uses the Distributed Component Object Model (DCOM) for remote management, rather than WinRM, and these settings are not enabled by default.

To address this problem, you must enable the following inbound Windows Firewall rules on the remote server you want to manage:

- COM+ Network Access (DCOM-In)
- Remote Event Log Management (NP-In)
- Remote Event Log Management (RPC)
- Remote Event Log Management (RPC-EPMAP)

To modify the firewall rules on the remote system, you can use one of the following methods:

- Open the Windows Firewall with Advanced Security MMC snap-in on the remote server (if it is a Full GUI installation).
- Run the `Netsh AdvFirewall` command from an administrative command prompt.
- Use the NetSecurity module in Windows PowerShell.
- Create a Group Policy object containing the appropriate settings and apply it to the remote server.

For remote management solutions, the Group Policy method provides distinct advantages. Not only does it enable you to configure the firewall on the remote system without accessing the server console directly, it also can configure the firewall on Server Core installations without working from the command line. Finally, and possibly most important for large networks, you can use Group Policy to configure the firewall on all servers you want to manage at once.

To configure Windows Firewall settings using Group Policy, use the following procedure. This procedure assumes that the server is a member of an AD DS domain and has the Group Policy Management feature installed.

CONFIGURE WINDOWS FIREWALL WITH GROUP POLICY

GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges. The Server Manager window appears.

1. Open the Group Policy Management console and create a new Group Policy object (GPO), giving it a name such as "Server Firewall Configuration."
2. Open the GPO you created using the Group Policy Management Editor.
3. Browse to the Computer Configuration\Policies\Windows Settings\Security Settings\Windows Firewall with Advanced Security\Inbound Rules node, as shown in Figure 6-3.

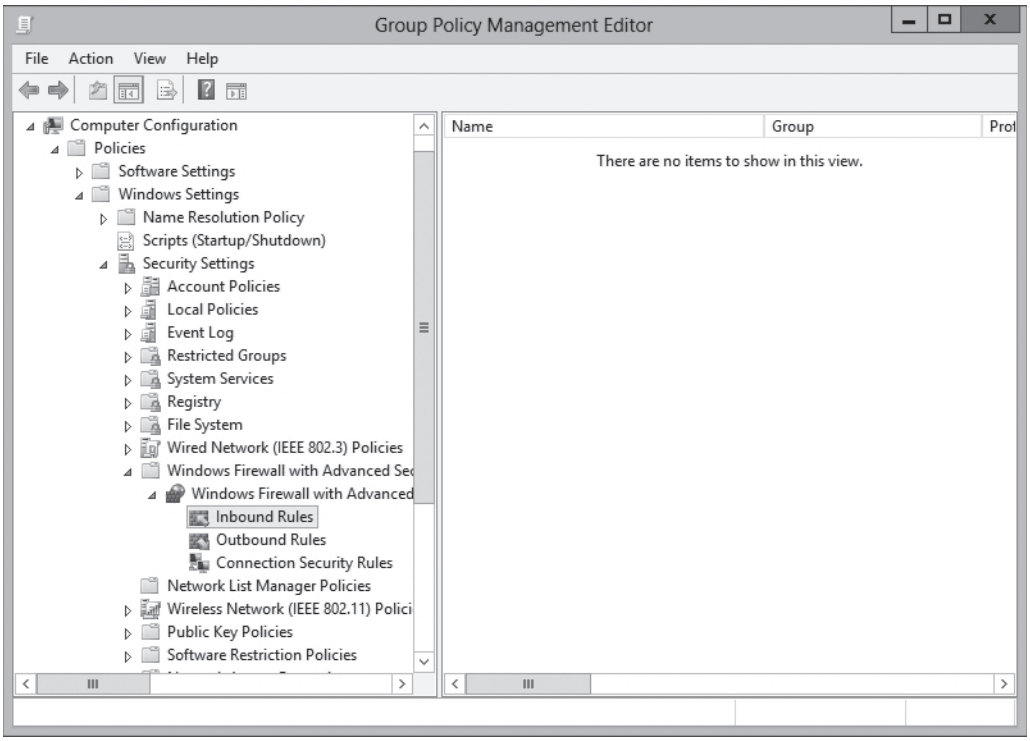


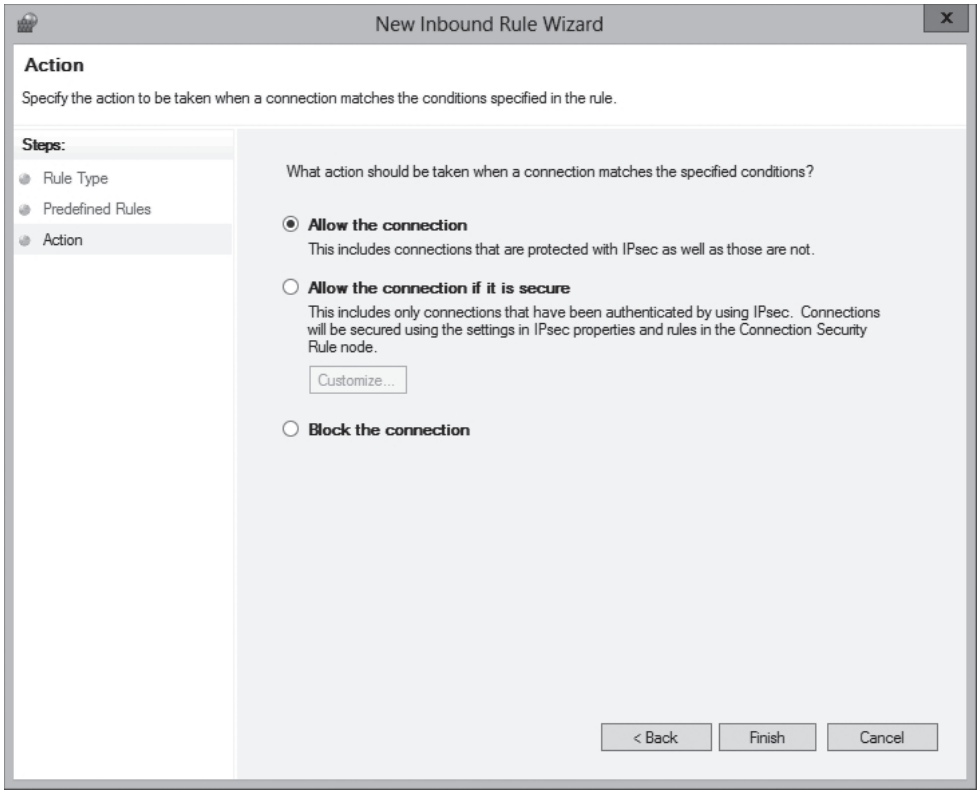
Figure 6-3

The *Windows Firewall with Advanced Security Inbound Rules* node in a Group Policy object

4. Right-click Inbound Rules and, from the context menu, select New Rule. The New Inbound Rule Wizard appears, displaying the Rule Type page.
5. Select the Predefined option and, in the drop-down list, select COM+ Network Access and click Next. The Predefined Rules page appears.
6. Click Next. The Action page appears, as shown in Figure 6-4.

Figure 6-4

The Action page of the New Inbound Rule Wizard



7. Leave the Allow the connection option selected and click Finish. The rule appears in the Group Policy Management Editor console.
8. Open the New Inbound Rule Wizard again.
9. Select the Predefined option and, in the drop-down list, select Remote Event Log Management and click Next. The Predefined Rules page appears, displaying the three rules in the Remote Event Log Management group.
10. Leave the three rules selected and click Next. The page appears.
11. Leave the *Allow the connection* option selected and click Finish. The three rules appear in the Group Policy Management Editor console.
12. Close the Group Policy Management Editor.
13. In the Group Policy Management console, link the Server Firewall Configuration GPO you just created to your domain.

CLOSE the Group Policy Management console.

The settings in the GPO you created deploy to your remote servers the next time they recycle or restart, and you can use MMC snap-ins, such as Computer Management and Disk Management, on them.

Managing Down-Level Servers

The Windows Firewall rules to enable for remote servers running Windows Server 2012 R2 are also disabled by default on computers running earlier versions of Windows Server, so you need to enable them there as well.

Unlike Windows Server 2012 R2, however, earlier versions of the operating system also lack the WinRM support needed for them to be managed using the new Server Manager.

By default, after you add servers running Windows Server 2008 or Windows Server 2008 R2 to the Windows Server 2012 R2 Server Manager, they appear with a manageability status that reads "Online – Verify WinRM 3.0 service is installed, running, and required firewall ports are open."

To add WinRM support to servers running Windows Server 2008 or Windows Server 2008 R2, you must download and install the following updates:

- .NET Framework 4.0
- Windows Management Framework 3.0

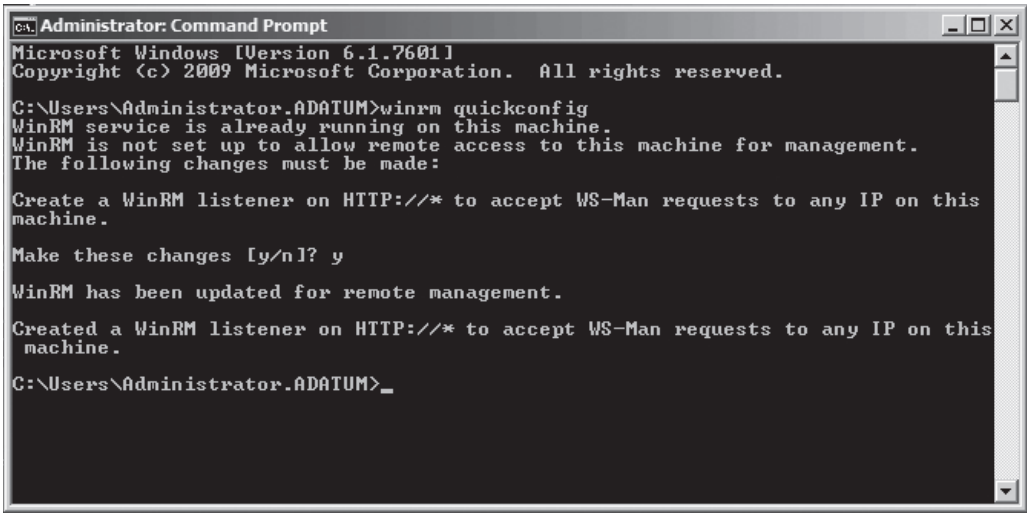
These updates are available from the Microsoft Download Center at Microsoft's website.

After you install the updates, the system automatically starts the Windows Remote Management service, but you must complete the following tasks on the remote server:

- Enable the Windows Remote Management (HTTP-In) rules in Windows Firewall.
- Create a WinRM listener by running the `winrm quickconfig` command at a command prompt with administrative privileges, as shown in Figure 6-5.

Figure 6-5

Creating a WinRM listener



- Enable the COM+ Network Access and Remote Event Log Management rules in Windows Firewall, as described in the previous section.

After installing the previous updates, you still have limitations to the management tasks you can perform on down-level servers from a remote location. For example, you cannot use the Add Roles and Features Wizard in Server Manager to install roles and features on down-level servers. These servers do not appear in the server pool on the *Select destination server* page.

However, you can use Windows PowerShell to install roles and features on servers running Windows Server 2008 and Windows Server 2008 R2 remotely, as in the following procedure.

INSTALL A FEATURE ON A DOWN-LEVEL SERVER

GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges. The Server Manager window appears.

1. Open a Windows PowerShell session with administrative privileges.
2. Establish a Windows PowerShell session with the remote computer by using the following command:

   ```
   Enter-PSSession <remote server name> -credential
   <user name>
   ```

3. Type the password associated with the user name you specified and press Enter.
4. Display a list of the roles and features on the remote server by using the following command:

   ```
   Get-WindowsFeature
   ```

5. Using the short name of the role or service as it appears in the Get-WindowsFeature display, install the component using the following command:

   ```
   Add-WindowsFeature <feature name>
   ```

6. Close the session with the remote server by using the following command:

   ```
   Exit-PSSession
   ```

CLOSE the Windows PowerShell window.

Working with Remote Servers

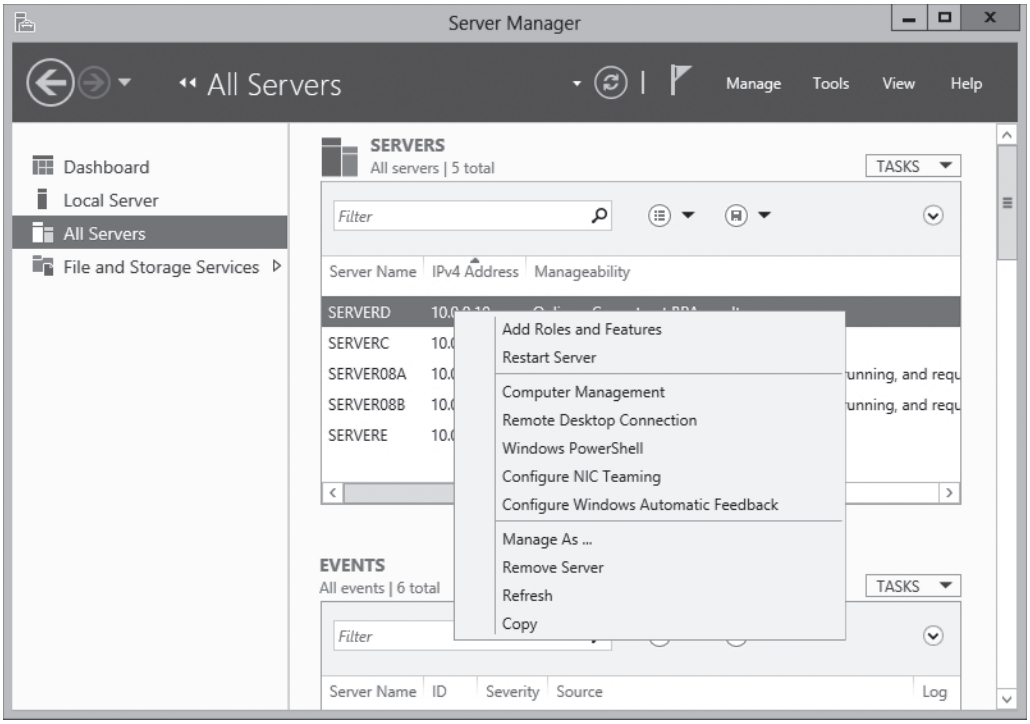
THE BOTTOM LINE After you add remote servers to Server Manager, you can access them using various remote administration tools.

Server Manager provides three basic methods for addressing remote servers, as follows:

- **Contextual tasks:** When you right-click a server in a Servers tile, anywhere in Server Manager, you see a context menu that provides access to tools and commands pointed at the selected server, as shown in Figure 6-6. Some are commands that Server Manager executes on the remote server, such Restart Server and Windows PowerShell. Others launch tools on the local system and direct them at the remote server, such as Microsoft Management Console snap-ins and the Install Roles and Features Wizard. Still others modify Server Manager itself, by removing servers from the interface. Other contextual tasks sometimes appear in the Tasks menus for specific panes.

Figure 6-6

Contextual tasks in Server Manager



- **Non-contextual tasks:** The menu bar at the top of the Server Manager console provides access to internal tasks, such as launching the Add Server and Install Roles and Features Wizards, as well as the Server Manager Properties dialog box, in which you can specify the console's refresh interval.
- **Non-contextual tools:** The console's Tools menu provides access to external programs, such as MMC snap-ins and the Windows PowerShell interface, that are directed at the local system.

■ Business Case Scenarios

Scenario 6-1: Managing Servers

Ralph is responsible for the 24 servers running a particular application, which are scattered all over his company's enterprise network. Ralph wants to use Server Manager on his Windows 8 workstation to manage those servers and monitor the events that occur on them. To do this, he must enable the incoming COM+ Network Access and Remote Event Log Management rules in Windows Firewall on the servers.

Because he can't travel to the locations of all the servers, and many of the sites do not have trustworthy IT personnel, Ralph has decided to use Group Policy to configure Windows Firewall on all of the servers. The company's Active Directory Domain Services tree is organized geographically, which means that Ralph's servers are located in many different OUs, all under one domain.

How can Ralph use Group Policy to deploy the required Windows Firewall rule settings to his 24 servers, and only those servers?

Scenario 6-2: Installing Windows PowerShell Web Access

You need a method to remotely manage a few servers from any client within the enterprise. You want to avoid any method that requires additional client software except a web browser. What will you use? Give an outline of tasks.

7

LESSON

Creating and Configuring Virtual Machine Settings

■ Virtualizing Servers

THE BOTTOM LINE

The concept of virtualizing servers has, in the past several years, grown from a novel experiment to a convenient lab and testing tool, as well as to a legitimate deployment strategy for production servers. Windows Server 2012 R2 includes the ***Hyper-V*** role, which enables you to create virtual machines, each of which runs in its own isolated environment. ***Virtual machine (VMs)*** are self-contained units that you can easily move from one physical computer to another, greatly simplifying the process of deploying network applications and services.

Server virtualization in Windows Server 2012 R2 is based on a module called a ***hypervisor***. Sometimes called a ***virtual machine monitor (VMM)***, the hypervisor is responsible for abstracting the computer's physical hardware and creating multiple virtualized hardware environments, called virtual machines (VMs). Each VM has its own (virtual) hardware configuration and can run a separate copy of an operating system. Therefore, with sufficient physical hardware and the correct licensing, a single computer running Windows Server 2012 R2 with the Hyper-V role installed can support multiple VMs, which you can manage as though they were standalone computers.

Virtualization Architectures

Virtualization products can use several different architectures to share a computer's hardware resources among several virtual machines.

The earlier types of virtualization products, including Microsoft Windows Virtual PC and Microsoft Virtual Server, require a standard operating system installed on a computer. This becomes the "host" operating system. Then you install the virtualization product, which adds the hypervisor component. The hypervisor essentially runs as an application on the host operating system, as shown in Figure 7-1, and enables you to create as many virtual machines as the computer has hardware to support.

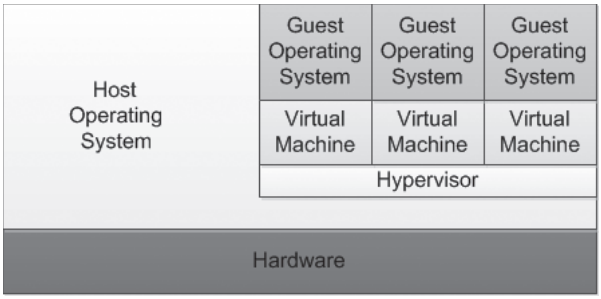


Figure 7-1

A hybrid VMM sharing hardware access with a host operating system

This arrangement, in which the hypervisor runs on top of a host operating system, is called a *Type II virtualization*. By using the Type II hypervisor, you create a virtual hardware environment for each virtual machine. You can specify how much memory to allocate to each VM, create virtual disk drives by using space on the computer's physical drives, and provide access to peripheral devices. You then install a "guest" operating system on each virtual machine as though you were deploying a new computer. The host operating system then shares access to the computer's processor with the hypervisor, with each taking the clock cycles it needs and passing control of the processor back to the other.

Type II virtualization can provide adequate virtual machine performance, particularly in classroom and laboratory environments, but it does not provide performance equivalent to separate physical computers. Therefore, it is not recommended for high-traffic servers in production environments.

The virtualization capability built into Windows Server 2012 R2, called Hyper-V, uses a different type of architecture. Hyper-V uses *Type I virtualization*, in which the hypervisor is an abstraction layer that interacts directly with the computer's physical hardware—that is, without an intervening host operating system. The term *hypervisor* is intended to represent the next level beyond the term *supervisor*, concerning responsibility for allocating a computer's processor clock cycles.

The hypervisor creates individual environments called *partitions*, each of which has its own operating system installed and accesses the computer's hardware via the hypervisor. Unlike Type II virtualization, no host operating system shares processor time with the hypervisor. Instead, the hypervisor designates the first partition it creates as the parent partition and all subsequent partitions as child partitions, as shown in Figure 7-2.

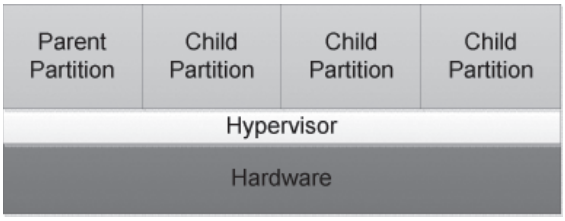


Figure 7-2

A Type 1 VMM, with the hypervisor providing all hardware access

The parent partition accesses the system hardware through the hypervisor, just as the child partitions do. The only difference is that the parent runs the virtualization stack, which creates and manages the child partitions. The parent partition is also responsible for the subsystems that directly affect the performance of the computer's physical hardware, such as Plug and Play, power management, and error handling. These subsystems run in the operating systems on the child partitions as well, but they address only virtual hardware, whereas the parent, or root, partition handles the real thing.

Installing Hyper-V

THE BOTTOM LINE

As soon as you have the appropriate hardware and the required licenses, you can add the Hyper-V role to Windows Server 2012 R2 via Server Manager, just as you would any other role.

Adding the Hyper-V role installs the hypervisor software, and, in the case of a full GUI installation, the management tools. The primary tool for creating and managing virtual machines and their components on Hyper-V servers is the Hyper-V Manager console. Hyper-V Manager provides you with a list of all the virtual machines on Windows Server 2012 R2 systems and enables you to configure both the server environments and those of the individual VMs. Windows PowerShell also includes a set of Hyper-V cmdlets that enable you to exercise complete control over VMs using that interface.

Microsoft recommends that you do not install other roles with Hyper-V. Any other roles that you need the physical computer to perform are better off implemented within one of the virtual machines you create with Hyper-V. You also might want to consider installing Hyper-V on a computer using the Server Core installation option to minimize the overhead expended on the partition. As with other roles, installing Hyper-V on Server Core excludes the management tools, which you must install separately as a feature.

Before you can install the Hyper-V role on a server running Windows Server 2012 R2, you must have appropriate hardware, as follows:

- A *64-bit processor* that includes hardware-assisted virtualization. This type of virtualization is available in processors that include a virtualization option, such of Intel Virtualization Technology (Intel VT) or AMD Virtualization (AMD-V) technology.
- A *system BIOS* that supports the virtualization hardware and on which the virtualization feature has been enabled.
- Hardware-enforced *Data Execution Prevention (DEP)*, which Intel describes as eXecuted Disable (XD) and AMD describes as No eXecute (NS). CPUs use this technology to segregate areas of memory for either storage of processor instructions or for storage of data. Specifically, you must enable Intel XD bit (execute disable bit) or AMD NX bit (no execute bit).

Using Hyper-V Manager

THE BOTTOM LINE

After you install the Hyper-V role and restart the computer, you can begin to create virtual machines and deploy operating systems on them.

The primary graphical tool for creating and managing virtual machines is the Hyper-V Manager console, which you can access from the *Tools* menu in Server Manager.

As with most of the Windows Server 2012 R2 management tools, including Server Manager itself, you can use the Hyper-V Manager console to create and manage virtual machines on multiple servers, enabling administrators to exercise full control over their servers from a central location.

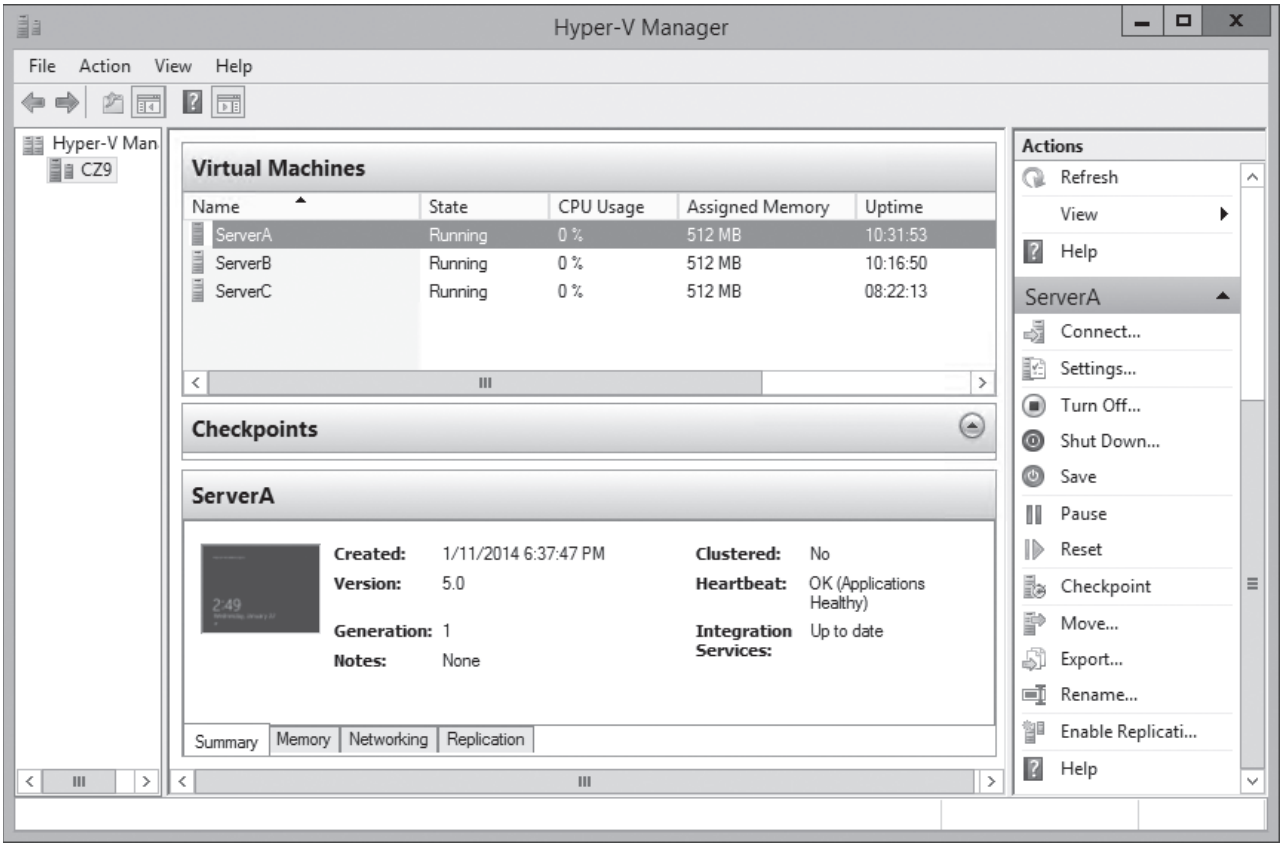
To run Hyper-V Manager on a server that does not have the Hyper-V role, you must install the Hyper-V Management Tools feature. You also can find these tools in the Remote Server Administration Tools package.

After you install and launch the Hyper-V Manager console, you can add servers to the display by right-clicking the *Hyper-V Manager* node in the left pane and selecting *Connect to Server* from the shortcut menu. In the *Select Computer* dialog box that appears, you can type or browse to the name of a Hyper-V server.

The Hyper-V Manager console lists all the virtual machines on the selected server, along with status information about each one (see Figure 7-3).

Figure 7-3

The Hyper-V Manager console



Creating a Virtual Machine

After installing Hyper-V and configuring Hyper-V Manager, you are ready to create virtual machines and install the operating system on each virtual machine that you create.

By using Hyper-V Manager, you can create new virtual machines and define the hardware resources that the system should allocate to them. In the settings for a particular virtual machine, depending on the physical hardware available in the computer and the limitations of the guest operating system, you can specify the number of processors and the amount of memory a virtual machine should use, install virtual network adapters, and create virtual disks using various technologies, including storage area networks (SANs).

By default, Hyper-V stores the files that make up virtual machines in the folders you specified on the *Default Stores* page during the role installation. Each virtual machine uses the following files:

- A virtual machine configuration (.vmc) file in XML format that contains the virtual machine configuration information, including all settings for the virtual machine
- One or more virtual hard disk (.vhd or .vhdx) files to store the guest operating system, applications, and data for the virtual machine

A virtual machine may also use a saved-state (.vsv) file, if the machine has been placed into a saved state.

Creating Generation 1 and Generation 2 VMs

In Windows Server 2012 R2, the Hyper-V implementation includes a new type of virtual machine, which it refers to as Generation 2. The VM type created by all previous versions is called Generation 1. When you create a new virtual machine in the Hyper-V manager, the New Virtual Machine Wizard includes a new page on which you specify whether you want to create a Generation 1 or Generation 2 VM. The New-VM cmdlet in Windows PowerShell also enables you to include a new –Generation parameter.

Generation 1 VMs are designed to emulate the hardware found in a typical computer, and to do this, they use drivers for specific devices, such as an AMI BIOS, an S3 graphics adapter, and an Intel chipset and network adapter. Generation 1 VMs that you create with Windows Server 2012 R2 Hyper-V are completely compatible with all previous Hyper-V versions.

Generation 2 VMs use synthetic drivers and software-based devices instead, and provide advantages that include the following:

- UEFI boot Instead of using the traditional BIOS, Generation 2 VMs support Secure Boot, using the Universal Extensible Firmware Interface (UEFI), which requires a system to boot from digitally signed drivers and enables them to boot from drives larger than 2 TB, with GUID partition tables.
- SCSI disks Generation 2 VMs omit the IDE disk controller used by Generation 1 VMs to boot the system and use a high-performance virtual SCSI controller for all disks, enabling the VMs to boot from VHDX files and support hot disk adds and removes.

The end result is a Generation 2 virtual machine that deploys much faster than its Generation 1 counterparts, and performs better as well. The limitations, however, are that Generation 2 VMs can only run the following guest operating systems:

- Windows Server 2012
- Windows Server 2012 R2
- Windows 8 64-bit
- Windows 8.1 64-bit

Configuring Guest Integration Services

In some cases, certain Hyper-V guest operating system features do not function properly using the OS's own device drivers. Hyper-V, therefore, includes a software package called ***guest integration services***, which you can install on your virtual machines for compatibility purposes.

Some functions provided by the guest integration services package are as follows:

- **Operating system shutdown** enables the Hyper-V Manager console to remotely shut down a guest operating system in a controlled manner, eliminating the need for you to log on and manually shut the system down.
- **Time synchronization** enables Hyper-V to synchronize the operating system clocks in parent and child partitions.
- **Data Exchange** enables the operating systems on parent and child partitions to exchange information, such as OS version information and fully qualified domain names.
- **Heartbeat** implements a service in which the parent partition sends regular heartbeat signals to the child partitions, which are expected to respond in kind. A failure of a child partition to respond indicates that the guest OS has frozen or malfunctioned.
- **Backup** allows backup of Windows virtual machines using Volume Shadow Copy Services.

The Windows Server 2012, Windows Server 2012 R2, and Windows 8, and Windows 8.1 operating systems have the latest guest integration services software built in, so you do not need to install the package on VMs running those operating systems as guests. Earlier versions of Windows, however, have previous versions of the guest integration services package that need to be upgraded, and some Windows versions do not include the package at all.

To upgrade the guest integration services on a Windows guest OS, use the following procedure.

INSTALL GUEST INTEGRATION SERVICES

GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges.

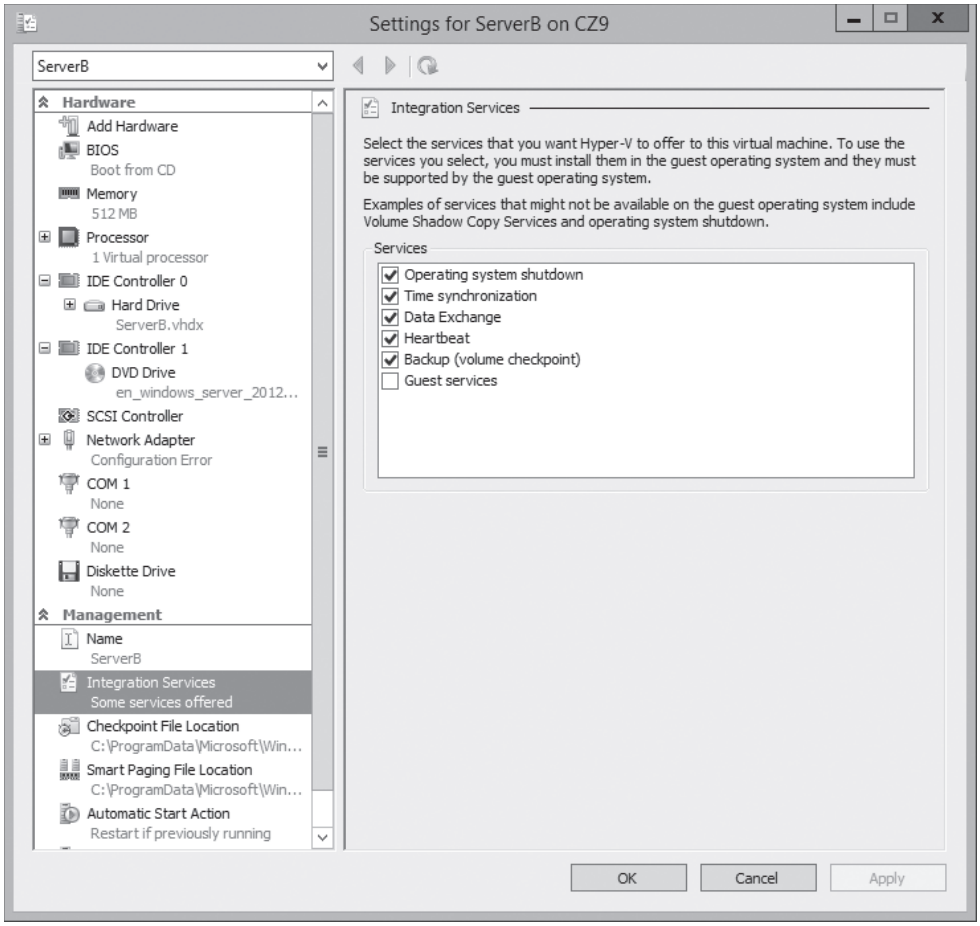
1. From the Tools menu of the *Server Manager* window, choose Hyper-V Manager. The *Hyper-V Manager* console appears.
2. In the left pane, select a Hyper-V server.
3. In the Actions pane, start the virtual machine on which you want to install the guest integration services and click Connect. A *Virtual Machine Connection* window appears.
4. From the Action menu of the Virtual Machine Connection window, choose Insert Integration Services Setup Disk. Hyper-V mounts an image of the guest integration services disk to a virtual disk drive and displays an *Autoplay* window.
5. Click Install Hyper-V Integration Services. A message box appears, asking you to upgrade the existing installation.
6. Click OK. The system installs the package and prompts you to restart the computer.
7. Click Yes to restart the computer.

After you install or upgrade the guest integration services, you can enable or disable each individual function by opening the *Settings* dialog box for the virtual machine and selecting the *Integration Services* page, as shown in Figure 7-4.

Now, you are ready to configure and manage the virtual machine as though you were working on a physical server. You can modify the network configuration, enable remote desktop, load the appropriate roles and features, and install applications.

Figure 7-4

Integration Services settings for a virtual machine



Using Enhanced Session mode

In previous versions of Hyper-V, when you open a Virtual Machine Connection window in the Hyper-V Manager console, you receive mouse and keyboard connectivity, plus a limited cut and paste functionality. To obtain any further access, such as audio or print functionality, you could establish a Remote Desktop Services connection to the VM, but this requires the computers to be connected to the same network, which is not always possible.

Starting in Windows Server 2012 R2, Hyper-V supports an enhanced session mode that enables the Virtual Machine Connection window to redirect any of the following local resources to VMs running Windows Server 2012 R2 or Windows 8.1:

- Display configuration
- Audio
- Printers
- Clipboard
- Smart cards
- USB devices
- Drives
- Supported Plug and Play devices

The enhanced session mode works by establishing a Remote Desktop Protocol connection between the host computer and the VM, but it does not require a standard network path

because it uses the VMBus instead. VMBus is a high-speed conduit between the various partitions running on a Hyper-V server.

Enhanced session mode is enabled by default in Windows 8.1, but in Windows Server 2012 R2, you must enable it on the Enhanced Session Mode Policy page of the Hyper-V Settings dialog box.

Allocating Memory

Dynamic memory enables Hyper-V to adjust the amount of RAM allocated to virtual machines, depending on their ongoing requirements.

Some computer components can be virtualized. You can take some disk space and create a virtual hard drive out of it, and you can take an image file and create a virtual DVD drive. You can also create virtual network interface adapters and other components, which appear like the real thing in a VM. System memory is different, however; it has no substitute, so all Hyper-V can do is take the physical memory installed in the computer and allocate it among the various virtual machines.

When you create a virtual machine with the *New Virtual Machine Wizard*, you specify how much memory the VM should use on the *Assign Memory* page. Obviously, the amount of memory available for use is based on the physical memory installed in the computer.

After you create the virtual machine, you can modify the amount of memory allocated to it by shutting down the VM, opening its *Settings* dialog box, and changing the *Startup RAM* setting on the *Memory* page, as shown in Figure 7-5. This enables you to experiment with various amounts of memory and dial in the optimum performance level for the system.

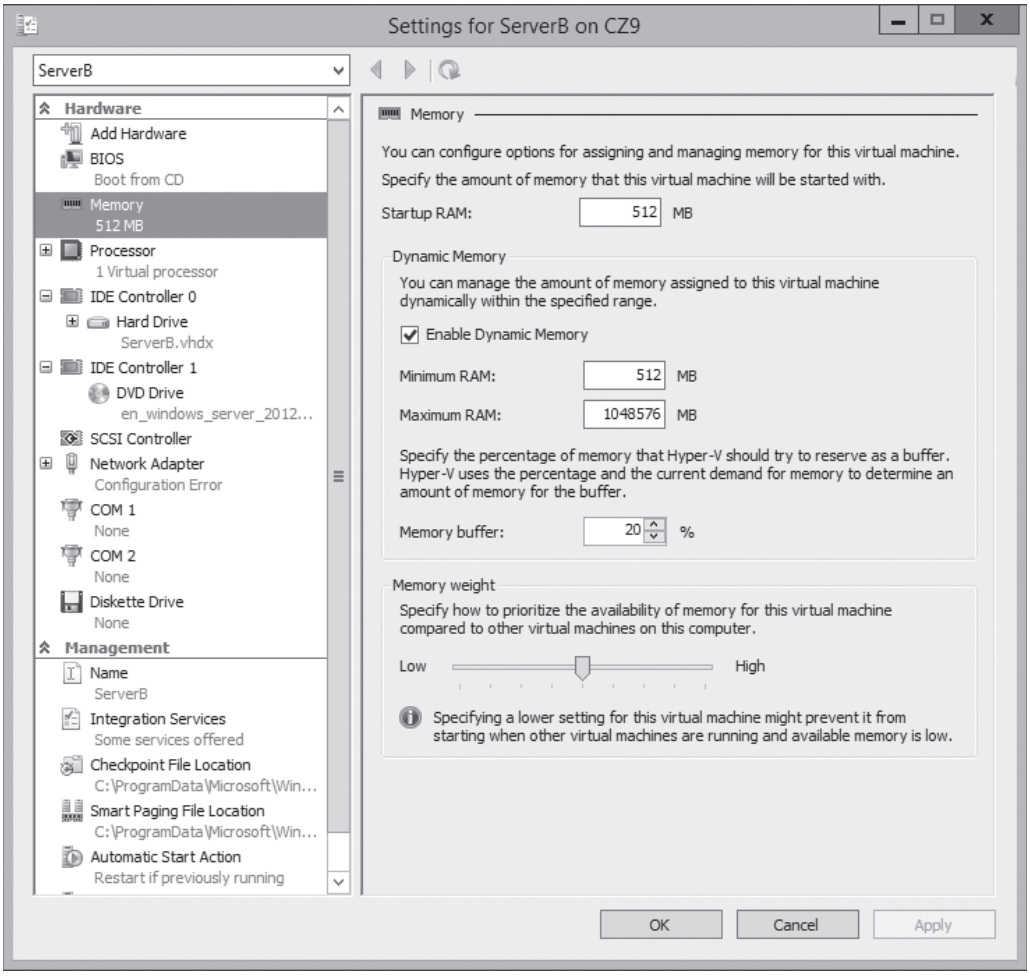


Figure 7-5

Memory settings for a virtual machine

USING DYNAMIC MEMORY

In the first versions of Hyper-V, shutting down the virtual machine was the only way to modify its memory allocation. In the Windows Server 2012 R2 version, however, you can use a feature called dynamic memory to reallocate memory automatically to the VM from a shared memory pool as its demands change. If a virtualized server starts to experience larger amounts of client traffic, for example, Hyper-V can increase the memory allocated to the system, and then reduce it when the traffic subsides.

To use dynamic memory, you must enable it by selecting the *Enable Dynamic Memory* check box on the VM's *Memory* page of the *Settings* dialog box, and then configure the following settings:

- **Startup RAM** specifies the amount of memory that you want to allocate to the VM when it starts. When you are using dynamic memory, this value can be the minimum amount of memory needed to boot the system.
- **Minimum RAM** specifies the smallest amount of memory the VM can use at any time. Operating systems can conceivably require more memory to start up than they do to run, so this value can be smaller than the *Startup RAM* value.
- **Maximum RAM** specifies the largest amount of memory that the VM can use at any time. The value can range from a low equal to the *Startup RAM* value to a high of 64 GB.
- **Memory buffer** contains a percentage that Hyper-V uses to calculate how much memory to allocate to the VM, compared to its actual utilization, as measured by performance counters. For example, with the *Memory buffer* value set to 20%, a VM with applications and operating system that consume 1 GB of memory receives a dynamic allocation of 1.2 GB.
- **Memory weight** contains a relative value that specifies the priority of this VM, compared to the other VMs on the same computer. When the physical memory in the computer is insufficient to allocate the full buffered amount specified for each VM, the VMs with the highest memory weight settings receive priority.

In addition to configuring the virtual machine settings, the guest operating system on the virtual machine must have the Windows Server 2012 R2 guest integration services to use dynamic memory.

The *Hyper-V* console also enables you to monitor the current memory allocation for each virtual machine. By selecting a VM and clicking the *Memory* tab on the bottom pane, you can see the system's current Assigned Memory and other statistics.

CONFIGURING SMART PAGING

Windows Server 2008 R2 Hyper-V introduced dynamic memory, but Windows Server 2012 and Windows Server 2012 R2 improve on the concept by adding the *Minimum RAM* setting. This enables Hyper-V to reduce the memory used by a virtual machine to a level lower than that needed to start the system, reclaiming that memory for other uses.

The problem with having minimum RAM values that are lower than the startup RAM values is that the supply of physical memory can become depleted with too many VMs running simultaneously at their minimum RAM values. If this occurs, a VM that has to restart might be unable to do so because not enough free memory is available to increase its memory allocation from its minimum RAM value to its startup RAM value.

To address this possibility, Hyper-V includes a feature called ***smart paging***. If a VM has to restart and not enough memory is available to allocate its startup RAM value, the system uses hard disk space to make up the difference and begins paging memory contents to disk.

Disk-access rates are far slower than memory-access rates, of course, so smart paging incurs a severe performance penalty, but the paging occurs only for as long as it takes to restart the VM and return it to its minimum RAM allocation.

Hyper-V uses smart paging only in highly specific conditions, such as when a VM must be restarted, no free memory is available, and the memory needed cannot be freed up by other means.

You can use the *Smart Paging File Location* page in a VM's *Settings* dialog box to specify a location for the paging file. Selecting the fastest possible hard drive is recommended.

■ Configuring Resource Metering

THE BOTTOM LINE Resource metering is a Windows PowerShell-based feature in Windows Server 2012 R2 Hyper-V that enables you to document virtual machine usage via a variety of criteria.

Organizations might want to track the use of virtual machines for various reasons. For large corporations, it might be a matter of internal accounting and controlling ongoing expenses, such as wide area network (WAN) bandwidth. For service providers, it might be necessary to bill customers based on the VM resources they use.

Resource metering uses Windows PowerShell cmdlets to track various performance metrics for individual VMs, including the following:

- CPU utilization
- Minimum/maximum/average memory utilization
- Disk space utilization
- Incoming/outgoing network traffic

Resource metering statistics remain consistent, even when you transfer VMs between host systems using live migration or move virtual hard disk files between VMs.

To use resource metering, you must first enable it for the specific VM that you want to monitor, using the *Enable-VMResourceMetering* cmdlet with the following syntax:

```
Enable-VMResourceMetering –VMName <name>
```

After you enable metering, you can display a statistical report at any time by using the *Measure-VM* cmdlet with the following syntax, as shown in Figure 7-6:

```
Measure-VM –VMName <name>
```

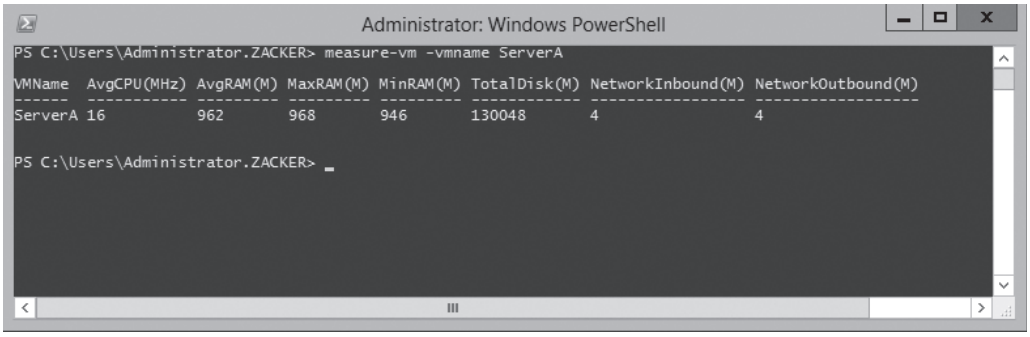


Figure 7-6

Displaying metering data with Windows PowerShell

In addition to metering resources for entire virtual machines, you can also create resource pools that help you monitor specific VM components, such as processors, memory, network adapters, or virtual hard disks. You create a resource pool using the *New-VMResourcePool* cmdlet and then enable metering for the pool using *Enable-VMResourceMetering*.

By using techniques such as pipelining, you can use the resource metering cmdlets to gather data on virtual machine performance and export it to applications or data files.

Using Remote FX

RemoteFX is a Windows Server feature that enables remote computers to connect Hyper-V guest VMs with an enhanced desktop experience, including the following:

- Graphics adapter virtualization With an appropriate graphics adapter installed in the server, client systems can offload some of the graphics processing tasks they would normally perform locally. By rendering graphics on a high-performance adapter installed on the host server and sending the resulting bitmaps to the client in a highly-compressed format, RemoteFX conserves resources on the client system, as well as network bandwidth.
- USB redirection Enables USB devices connected to client computers to be redirected to the host server, including printer, scanner, interface, storage, and audio devices.
- Intelligent encoding and decoding The RemoteFX server can encode data using the system's CPU, GPU, or dedicated hardware. In the same way, the client system can decode incoming data using its CPU, GPU, or a hardware decoder. The systems can select different codecs, depending on the data type.

RemoteFX relies on the Remote Desktop Protocol (RDP) for client/server communications, just like Remote Desktop Services, but uses these new features to provide capabilities such as multitouch support and media redirection.

Setting up the host side of RemoteFX consists of the following basic steps.

1. Install the Hyper-V role on the host server.
2. Install a RemoteFX-approved server graphics adapter and an appropriate Windows Display Driver Model (WDDM) driver on the host server.
3. Install the Remote Desktop Virtualization Host role feature for the Remote Desktop Services role and restart the server.
4. In Hyper-V Manager, open the Hyper-V Settings dialog box and, under Physical GPUs, select the graphics adapter you installed and select the Use this GPU with RemoteFX checkbox.

To set up the client side of RemoteFX, complete the following basic steps.

1. Install Windows 8 Enterprise or Windows 8.1 Enterprise on a virtual machine.
2. Enable Remote Desktop on the virtual machine.
3. Install Hyper-V Integration Services on the virtual machine.
4. In Hyper-V Manager, open the Settings dialog box for the VM and, under Add Hardware, select RemoteFX 3D Video Adapter and specify the screen resolution and number of monitors you plan to use.
5. Turn on the VM and log on using RDP.

■ Business Case Scenarios

Scenario 7-1: Isolating Server Applications

You have two network accounting applications, neither of which is processor hungry. Both of these applications must be kept totally isolated from each other and from all other applications. Both applications will access a centralized database server. What server configuration solution do you recommend?

Scenario 7-2: Configuring Virtual Machine Memory

Alice has a computer running Windows Server 2012 R2 with 8 GB of memory installed, which she has configured as a Hyper-V server. After creating eight VMs with the New Virtual Machine Wizard, each with a startup RAM value of 1,024 MB, Alice is having trouble getting all eight VMs to boot. What settings can she modify to resolve the problem without changing the startup RAM value?

8

LESSON

Creating and Configuring Virtual Machine Storage

Working with Virtual Disks

THE BOTTOM LINE

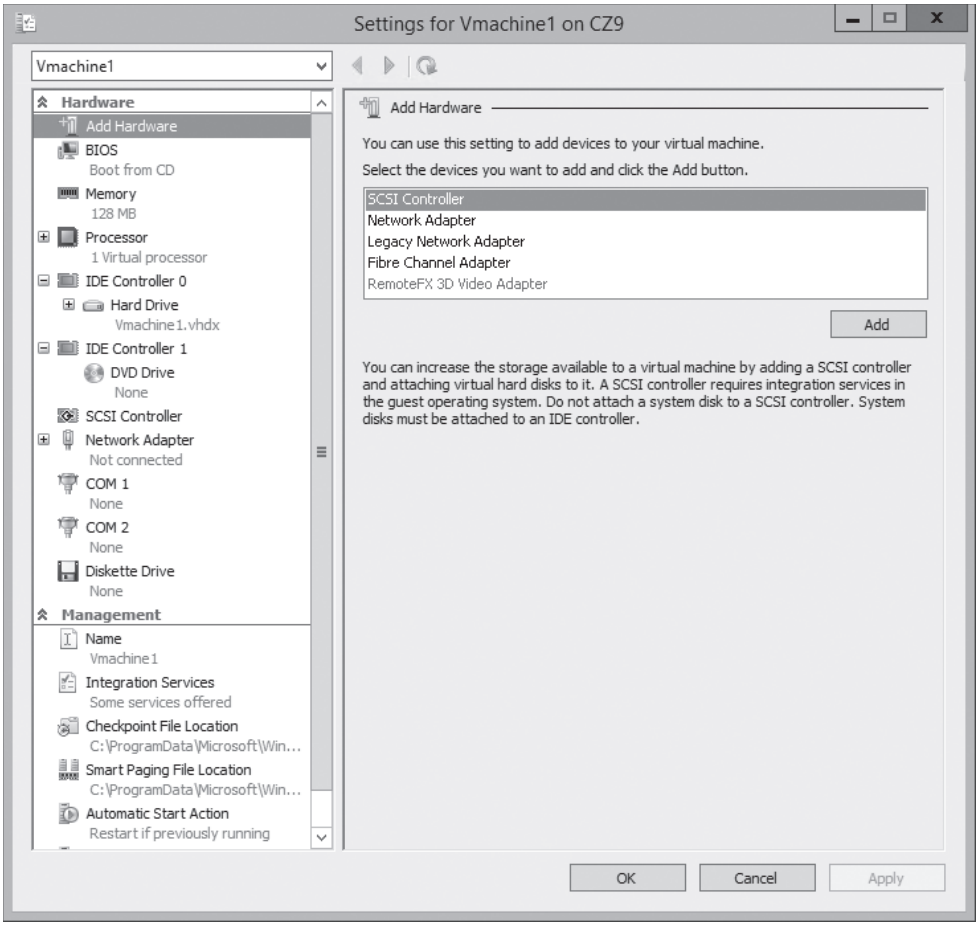
When you create a virtual machine (VM) in Windows Server 2012 R2 Hyper-V, you emulate all the standard components that you typically find in a physical computer. When you virtualize memory, as discussed in Lesson 7, "Creating and Configuring Virtual Machine Settings," you take a portion of the physical memory in the computer and dedicate it to a VM. The same is true with hard disk space. Hyper-V uses a specialized ***virtual hard disk (VHD)*** format to package part of the space on a physical disk and make it appear to the virtual machine as though it is physical hard disk drive.

When you create a new virtual machine in Hyper-V by using the New Virtual Machine Wizard, the wizard creates a virtual storage subsystem that consists of two IDE (Integrated Drive Electronics) controllers and one SCSI (Small Computer Systems Interface) controller, as shown in Figure 8-1. The IDE controllers host the virtual machine's system drive and its DVD drive. As with their physical equivalents, each IDE controller can host two devices, so you can create two additional virtual drives and add them to the system.

The SCSI controller, in the default Generation 1 VM configuration, is unpopulated, and you can create additional drives and add them to that controller to provide the VM with additional storage. In a Generation 2 VM, the system and DVD drives are connected to the default SCSI controller and there is no IDE alternative. In a VM of either generation, you can also create additional SCSI controllers and add drives to them. By creating multiple drives and controllers, Hyper-V makes it possible to construct virtual storage subsystems that emulate almost any physical storage solution you might devise.

Figure 8-1

The default VM drive controller configuration



Understanding Virtual Disk Formats

Windows Server 2012 R2 Hyper-V supports the original VHD disk image file and the new VHDX format.

The original VHD format was created by a company called *Connectix* for its Virtual PC product. Microsoft later acquired the product and used the VHD format for all its subsequent virtualization products, including Hyper-V. There are three types of VHD files, as follows:

- **Fixed hard disk image:** This image file is a specified size in which all the disk space required to create the image is allocated during its creation. Fixed disk images can be considered wasteful in terms of storage, because they can contain large amounts of empty space, but they are also efficient from a processing standpoint, because there is no overhead due to dynamic expansion.
- ***Dynamic hard disk* image:** This image file has a specified maximum size, which starts out small and expands as needed to accommodate the data the system writes to it.
- ***Differencing hard disk* image:** This child image file is associated with a specific parent image. The system writes all changes made to the data on the parent image file to the child image, to facilitate a rollback at a later time.

VHD images are limited to maximum size of 2 terabytes (TB) and are compatible with all versions of Hyper-V, as well as Microsoft's Type 2 hypervisor products, such as Virtual Server and Virtual PC. Windows Server 2012 introduced an updated version of the format, which uses a VHDX filename extension.

VHDX image files can be as large as 64 TB, and they also support 4 KB logical sector sizes, to provide compatibility with new 4 KB native drives. VHDX files can also use larger block sizes (up to 256 MB), which enable you to fine-tune the performance level of a virtual storage subsystem to accommodate specific applications and data file types. However, VHDX files are not backwards compatible and can be read only by Windows Server 2012 R2 Hyper-V servers. If migrating your virtual machines from Windows Server 2012 R2 to an older version of Hyper-V is a remote possibility, you should continue using the VHD file format.

Creating Virtual Disks

Windows Server 2012 R2 Hyper-V provides several ways to create virtual disk files. You can create them as part of a virtual machine, or create them later and add them to a VM.

The graphical interface in Hyper-V Manager provides access to most of the VHD parameters, but the new Windows PowerShell cmdlets included in Windows Server 2012 R2 provide the most granular control over the disk image format.

CREATING A VIRTUAL DISK WITH A VM

The New Virtual Machine Wizard includes a Connect Virtual Hard Disk page, with which you can add a single disk to your new VM. The options for this disk are limited and consist of the following:

- **Create a virtual hard disk** enables you to specify the name, location, and size of a new virtual hard disk, but you can create only a dynamically expanding disk using the VHDX format.
- **Use an existing virtual hard disk** enables you to specify the location of an existing VHD or VHDX disk, which the VM presumably uses as its system disk.
- **Attach a virtual hard disk later** prevents the wizard from adding virtual disks to the VM configuration. The assumption is that you will manually add a disk later, before you start the virtual machine.

The object of this wizard page is to create the disk on which you will install the VM's operating system, or select an existing disk on which an OS is already installed. The disk the wizard creates is always a dynamically expanding one connected to IDE Controller 0 on a Generation 1 VM or the SCSI Controller on a Generation 2 VM.

CREATING A NEW VIRTUAL DISK

You can create a virtual hard disk file at any time, without adding it to a virtual machine, by using the *New Virtual Hard Disk Wizard in Hyper-V Manager*.

To create a new virtual disk, use the following procedure.

CREATE A NEW VIRTUAL DISK

GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges.

1. From the Tools menu in the *Server Manager* window, select Hyper-V Manager. The *Hyper-V Manager* console appears.
2. In the left pane, select a Hyper-V server.
3. From the Action menu, select New > Hard Disk. The *New Virtual Hard Disk Wizard* appears, displaying the *Before You Begin* page.
4. Click Next. The *Choose Disk Format* page appears.

5. Select one of the following disk format options and click Next. The *Choose Disk Type* page appears.
   - VHD creates an image no larger than 2 TB, using the highly compatible VHD format.
   - VHDX creates an image up to 64 TB in size, using the new VHDX format.
6. Select one of the following disk type options and click Next. The *Specify Name and Location* page appears.
   - Fixed size creates a disk of a specific size, allocating all the space at once.
   - Dynamically expanding creates a disk that grows to the maximum size you specify as you add data.
   - Differencing creates a child drive that contains changes made to a specified parent drive.
7. Specify a filename for the disk image in the Name text box and, if desired, specify a location for the file other than the server default. Then, click Next. The *Configure Disk* page appears.
8. Select and configure one of the following options and click Next. The *Completing the New Virtual Hard Disk Wizard* page appears.
   - Create a new blank virtual hard disk specifies the size (or the maximum size) of the disk image file to create.
   - Copy the contents of the specified physical disk enables you to select one of the physical hard disks in the computer and copy its contents to the new disk image.
   - Copy the contents of the specified virtual hard disk enables you to select an existing virtual disk file and copy its contents to the new disk image.
9. Click Finish.

The wizard creates the new image disk and saves it to the specified location.

CREATING DIFFERENCING DISKS

A differencing disk enables you to preserve an existing virtual disk image file in its original state, while mounting it in an OS and even modifying its contents. For example, when building a laboratory setup, you can create a baseline system by installing a clean copy of an OS on a new virtual disk and configure the environment to your needs. Then, you can create a new child-differencing disk, using your baseline image as the parent. All subsequent changes you make to the system are written to the differencing disk, whereas the parent remains untouched. You can experiment on the test system, knowing that you can revert back to your baseline configuration by creating a new differencing disk.

You can create multiple differencing disks that point to the same parent image, enabling you to populate a lab network with as many virtual machines as you need, without having to repeatedly install the OS and while saving on disk space.

To create a cloned version of a baseline installation with a differencing disk, use the following procedure.

1. **Install and configure the baseline virtual machine:** Create a new virtual machine with a new disk image file and install a guest OS on it. Configure the OS as needed and install any roles, features, applications, or services you need.
2. **Generalize the parent image:** Open an elevated command prompt on the baseline system and run the *Sysprep.exe* utility. Sysprep configures the system to assign itself a new, unique security ID (SID) the next time the computer starts. This enables you to create multiple cloned systems from a single disk image.
3. **Create a parent disk image:** After you generalize the baseline installation, you no longer need the original virtual machine. You can delete everything except the VHD or VHDX file

containing the disk image. This file becomes your parent image. Open the Properties sheet for the image file and set the read-only flag, to ensure that the baseline does not change.

4. **Create a differencing disk:** Using the *New Virtual Hard Disk Wizard* or the *New-VHD* cmdlet for Windows PowerShell, create a new differencing disk, pointing to the baseline image you created and prepared previously as the parent image.
5. **Create a cloned virtual machine:** Create a new virtual machine and, on the *Connect Virtual Hard Disk* page, attach the differencing disk you created to it, by using the *Use an existing virtual hard disk* option.

You can then create additional cloned VMs, with differencing disks that use the same parent. Each one can function independently, and the parent disk will remain unchanged.

When you create a differencing drive by using the *New Virtual Hard Disk Wizard*, selecting the *Differencing* option on the *Choose Disk Type* page causes the *Configure Disk* page to appear. In the *Location* text box, you must specify the name of the file to use as the parent image.

In the same way, if you create the differencing disk by using Windows PowerShell, you must run the New-VHD cmdlet with the `-Differencing` parameter and the `-ParentPath` parameter, specifying the location of the parent disk.

Configuring Pass-Through Disks

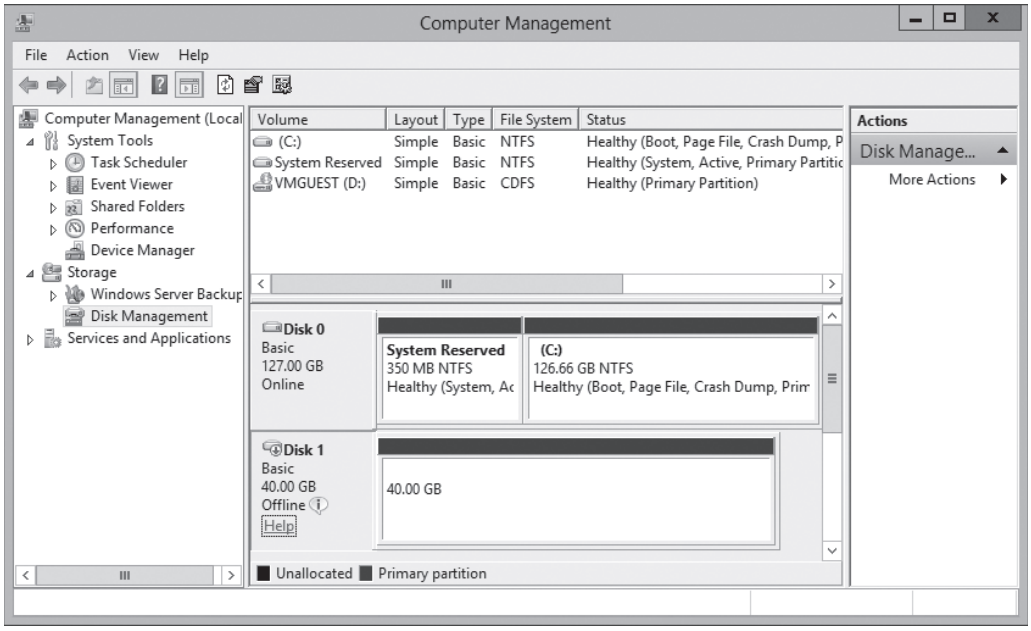
So far, this lesson focusses primarily on virtual hard disks—that is, areas of space on a physical disk drive allocated for use by virtual machines. However, it is also possible for VMs to access physical disks directly.

A ***pass-through disk*** is a type of virtual disk that points not to an area of space on a physical disk, but to a physical disk drive itself, installed on the host computer. When you add a hard drive to any of the controllers in a virtual machine, you can select a physical hard disk, as opposed to virtual one.

To add a physical hard disk to a virtual machine, however, the VM must have exclusive access to it. That is, you must first take the disk offline in the parent OS, by using the *Disk Management* snap-in, as shown in Figure 8-2, or the *Diskpart.exe* utility. After the disk is offline, it is available for selection in the *Physical hard disk* drop-down list.

Figure 8-2

An offline disk in the Disk Management snap-in



Modifying Virtual Disks

Windows Server 2012 R2 and Hyper-V provide several ways for you to manage and manipulate virtual hard disk images without mounting them in a virtual machine.

After you create a virtual hard disk, whether you attach it to a virtual machine or not, you can manage it by using the *Edit Virtual Hard Disk Wizard* in *Hyper-V Manager*. To edit an existing VHD or VHDX file, use the following procedure.

EDIT A VIRTUAL DISK

GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges.

1. From the Tools menu in the *Server Manager* window, select Hyper-V Manager. The *Hyper-V Manager* console appears.
2. In the left pane, select a Hyper-V server.
3. In the Actions pane, select Edit Disk. The *Edit Virtual Hard Disk Wizard* appears, displaying the *Before You Begin* page.
4. Click Next. The *Locate Disk* page appears.
5. Type or browse to the name of the VHD or VHDX file you want to open and click Next. The *Choose Action* page appears.
6. Select one of the following functions and click Next. The *Completing the Edit Virtual Hard Disk Wizard* page appears.
   - Compact reduces the size of a dynamically expanding or differencing disk by deleting empty space, while leaving the disk's capacity unchanged.
   - Convert changes the type of format of a disk by copying the data to a new disk image file.
   - Expand increases the capacity of the disk by adding empty storage space to the image file.
   - Shrink reduces the capacity of the disk by deleting empty storage space from the file.
   - Merge combines the data on a differencing disk with the parent disk to form a single composite image file.
7. Complete any new pages presented by the wizard as a result of your selection and click Finish.

The options that appear on the wizard's Choose Action page depend on the current status of the image file you select. For example, the Merge option only appears if you choose a differencing disk, and the Shrink option does not appear unless there is free space in the file that the wizard can delete.

In addition to these disk editing functions provided by Hyper-V Manager, it is also possible to use the Disk Management snap-in to mount a VHD or VHDX file as a drive and access its contents, just as if it was a physical disk.

Creating Checkpoints

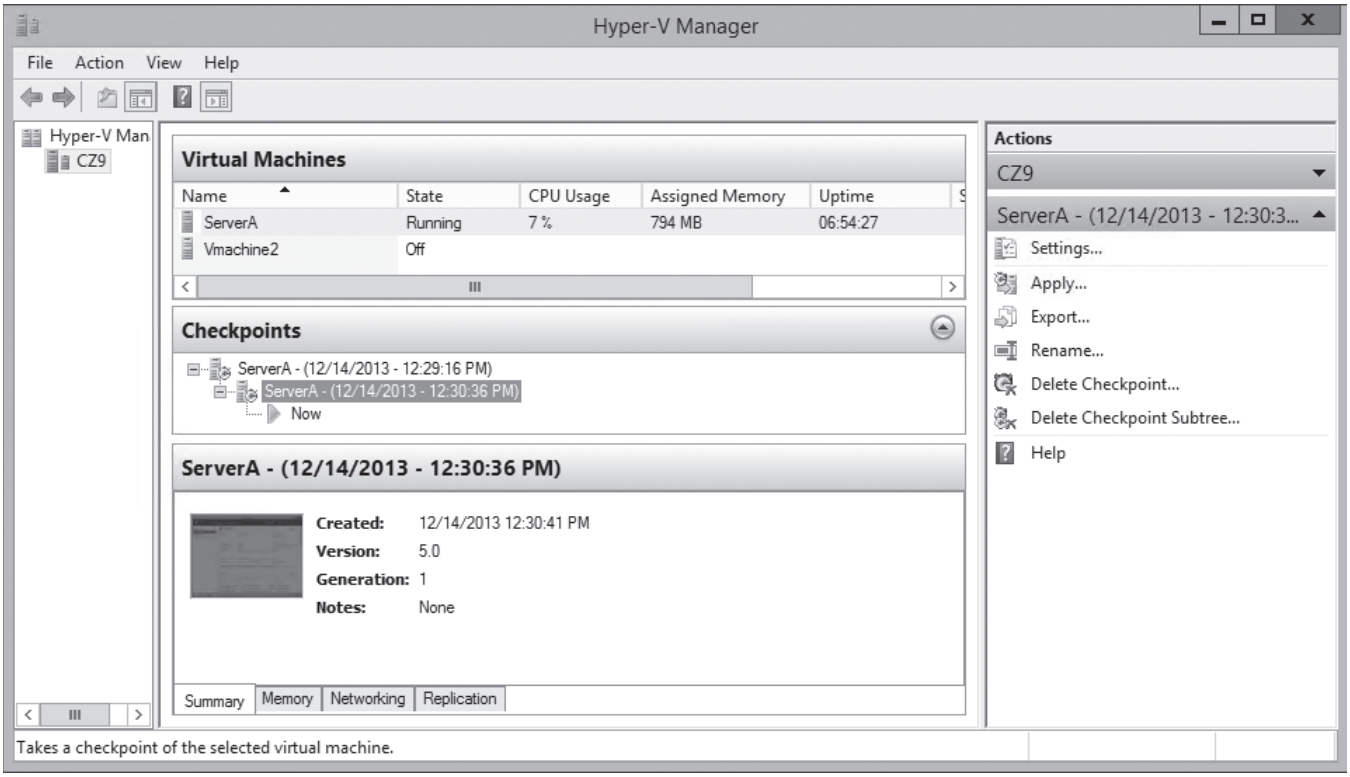
In Hyper-V, a ***checkpoint*** is a captured image of the state, data, and hardware configuration of a virtual machine at a particular moment in time.

Creating checkpoints is a convenient way for you to revert a virtual machine to a previous state at will. For example, if you create a checkpoint just before applying a system update, and the update is somehow problematic, you can apply the checkpoint and return the VM to the state it was in before you applied the update.

Creating a checkpoint is as simple as selecting a running virtual machine in Hyper-V Manager and selecting Checkpoint from the Actions pane. The system creates a checkpoint file, with an AVHD or AVHDX extension, in the same folder as the virtual hard disk file, and adds the checkpoint to Hyper-V Manager display, as shown in Figure 8-3.

Figure 8-3

A checkpoint in Hyper-V Manager



Checkpoints is a useful tool for you implementing a test environment in Hyper-V, but this tool is not recommended for heavy use in production environments. Apart from consuming disk space, the presence of checkpoints can reduce the overall performance of a virtual machine's disk subsystem.

Configuring Storage Quality of Service

Because it is common for there to be more than one virtual hard disk hosted by a single physical hard disk, it is possible for one virtual disk to monopolize the input/output capacity of a physical disk, causing the other virtual disks to slow down. To help prevent this, Hyper-V in Windows Server 2012 R2 enables you to control the Quality of Service (QoS) for a given virtual hard disk.

To use QoS in Windows Server 2012 R2, you must first install the Hyper-V role. QoS management in Hyper-V takes the form of controls that enables you to specify the minimum and maximum input/output operations per second (IOPS) for a disk. The system can generate notifications for administrators when a disk's IOPS falls below a specified minimum requirement. To configure storage QoS, you open the Settings dialog box for a VM, expand a hard drive component, and select Advanced Features to display the Advanced Features page.

After selecting the Enable Quality of Service Management checkbox, you can specify minimum and maximum IOPS values for the disk, to throttle its throughout in 8 KB increments.

■ Connecting to a SAN

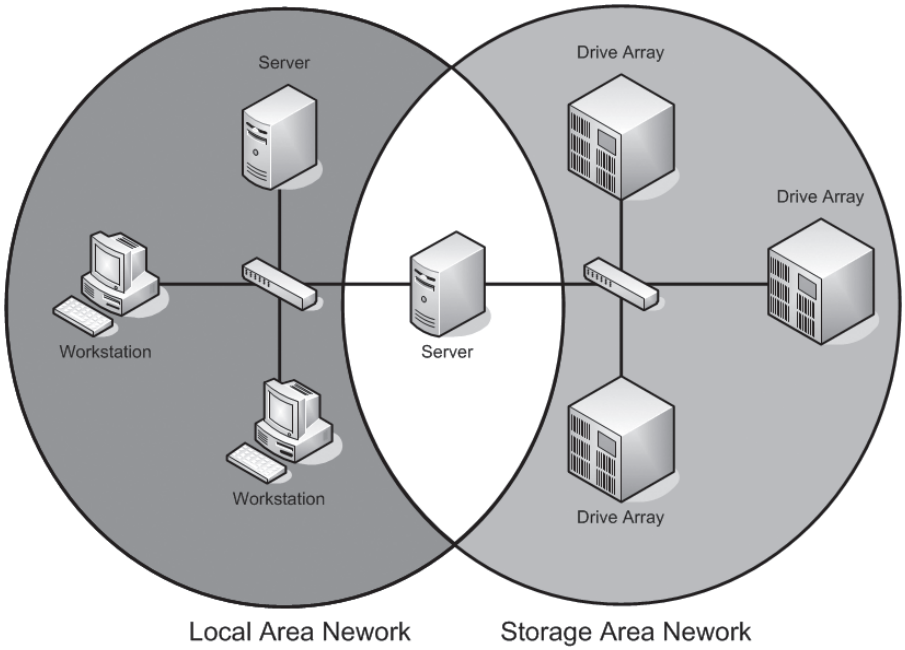
THE BOTTOM LINE

At its most basic level, a ***storage area network (SAN)*** is a network dedicated to high-speed connections between servers and storage devices.

Rather than installing disk drives into servers, or connecting them by using an external SCSI bus, a SAN consists of one or more drive arrays equipped with network interface adapters, which you connect to your servers by using standard twisted pair or fiber optic network cables. A SAN-connected server, therefore, has a minimum of two network adapters, one for the standard LAN connection and one for the SAN, as shown in Figure 8-4.

Figure 8-4

A server connected to a SAN



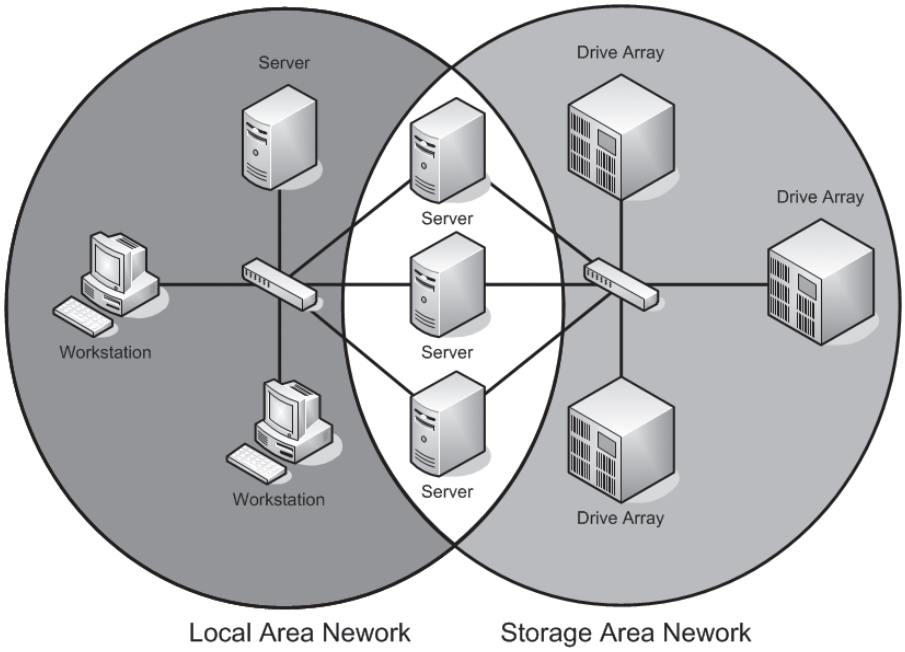
The advantages of SANs are many. By connecting the storage devices to a network rather than to the servers themselves, you avoid the limitations imposed by the maximum number of devices you can connect directly to a computer. SANs also provide added flexibility in their communications capabilities. Because any device on a SAN can communicate with any other device on the same SAN, high-speed data transfers can occur in any of the following ways:

- **Server to storage:** Servers can access storage devices over the SAN just as if they were connected directly to the computer.
- **Server to server:** Servers can use the SAN to communicate directly with each other at high speeds, to avoid flooding the LAN with traffic.
- **Storage to storage:** Storage devices can communicate among themselves without server intervention, such as to perform backups from one medium to another or to mirror drives on different arrays.

Although a SAN is not a high availability technology, you can make it into one by connecting redundant servers to the same network, as shown in Figure 8-5, enabling them to access the same data storage devices. If one server fails, another can assume its roles by accessing the same data, which is called *server clustering*.

Figure 8-5

Multiple servers connected to a SAN



Because they use standard networking technologies, SANs can also greatly extend the distances between servers and storage devices. You can design a SAN that spans different rooms, different floors, or even different buildings, just as you would with a standard computer network.

Using Fibre Channel

Fibre Channel is a high-speed serial networking technology that was originally designed for use with supercomputers, but which is now associated primarily with storage area networking.

Fibre Channel is a versatile technology, supporting various network media, transmission speeds, topologies, and upper-level protocols. Its primary disadvantage is that it requires specialized hardware that can be extremely expensive.

Installing a Fibre Channel SAN means building a new network with its own special medium, switches, and network interface adapters. In addition to the hardware costs, which can easily be 10 times that of a traditional Ethernet network, you should consider installation and maintenance expenses. Fibre Channel is a rather esoteric technology, with relatively few experts in the field. To install and maintain a Fibre Channel SAN, an organization must either hire experienced staff or train existing personnel on the new technology.

Connecting Virtual Machines to a SAN

In the past, the specialized networking technologies used to build Fibre Channel SANs made it difficult to use them with virtualized servers. However, since the Windows Server 2012 implementation, Hyper-V has supported the creation of virtual Fibre channel adapters.

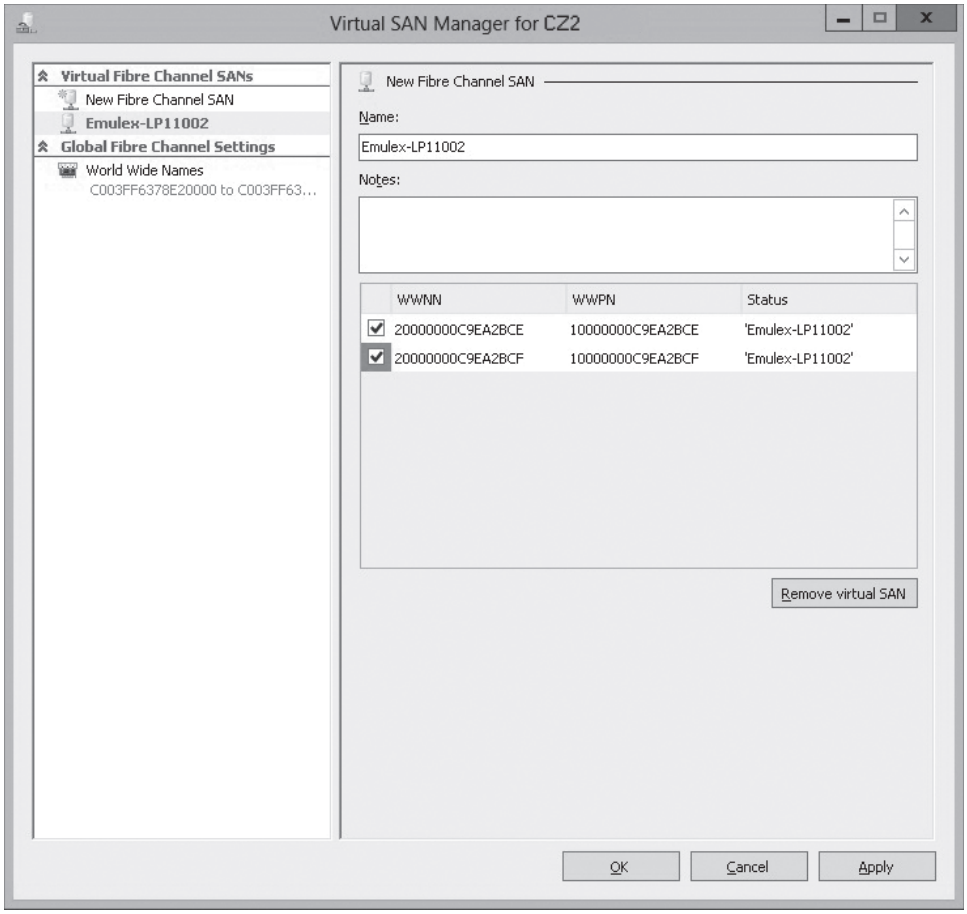
A Hyper-V Fibre Channel adapter is essentially a pass-through device that enables a virtual machine to access a physical Fibre Channel adapter installed in the computer, and through that, the external resources connected to the SAN. With this capability, applications running on virtual machines can access data files stored on SAN devices, and you can use VMs to create server clusters with shared storage subsystems.

To support virtual Fibre Channel connectivity, the physical Fibre Channel host bus adapter(s) in the host computer must have drivers that explicitly support virtual Fibre Channel. This support is relatively rare, but more manufacturers are expected to update their drivers to provide the necessary support. Your SAN must also be able to address its connected resources using logical unit numbers (LUNs).

Assuming you have the appropriate hardware and software installed on the host commuter, you implement the Fibre Channel capabilities in Hyper-V by first creating a virtual SAN. You do this by using the *Virtual SAN Manager*, accessible from *Hyper-V Manager*. When you create the virtual SAN, the World Wide Node Names (WWNNs) and World Wide Port Names (WWPNs) of your host bus adapter appear, as shown in Figure 8-6.

Figure 8-6

WWNNs and WWPNs in a virtual SAN



The next step is to add a Fibre Channel adapter to a virtual machine from the *Add Hardware* page in the *Settings* dialog box. Then, the virtual SAN you created previously is available in the *Fibre Channel Adapter* page. Hyper-V virtualizes the SAN and makes the WWNNs and WWPNs available to the VM.

■ Business Case Scenarios

Scenario 8-1: Creating Differencing Disks

To conduct multiple tests, you require several VMs with the same baseline installation. You decide to employ differencing disks to create your VMs. Walk through the steps.

Scenario 8-2: Modifying Virtual Disks

You need to modify an existing VHD file. How do you proceed?

LESSON 9

Creating and Configuring Virtual Networks

■ Using Virtual Networking

THE BOTTOM LINE

Networking is a critical part of creating a virtual machine (VM) infrastructure. Depending on your network plan, the virtual machines you create on a Windows Server 2012 R2 Hyper-V server can require communication with other virtual machines, with the computers on your physical network, and/or with the Internet.

When you build a network out of physical computers, you install a network interface adapter in each one and connect it to a hardware switch. The same principle is true in a Hyper-V environment, except that you use virtual components rather than physical ones. Each virtual machine you create has at least one virtual network adapter, and you can connect that adapter to a virtual switch. This enables you to connect the virtual machines on your Hyper-V server in various network configurations that either include or exclude the systems on your physical network.

Therefore, Hyper-V creates a pool of 256 MAC addresses during the installation. You can create multiple virtual switches on a Hyper-V server and multiple network adapters in each virtual machine. Multiple virtual switches and network adapters enable you to create a flexible networking environment, which is suitable for anything from a laboratory or classroom network to a production environment. In addition, Windows Server 2012 R2 can create extensions for virtual switches, so that software developers can enhance their capabilities.

Creating Virtual Switches

A ***virtual switch***, like its physical counterpart, is a device that functions at layer 2 of the Open Systems Interconnect (OSI) reference model. A switch has a series of ports, each of which is connected to a computer's network interface adapter. Any computer connected to the switch can transmit data to any other computer connected to the same switch.

Unlike physical switches, the virtual switches created by Hyper-V can have an unlimited number of ports, so you don't need to connect switches together or use uplinks and crossover circuits.

CREATING A NEW VIRTUAL SWITCH

Hyper-V in Windows Server 2012 R2 supports three types of switches, which you must create in the *Virtual Switch Manager* before you can connect virtual machines to them.

To create a new virtual switch, use the following procedure.

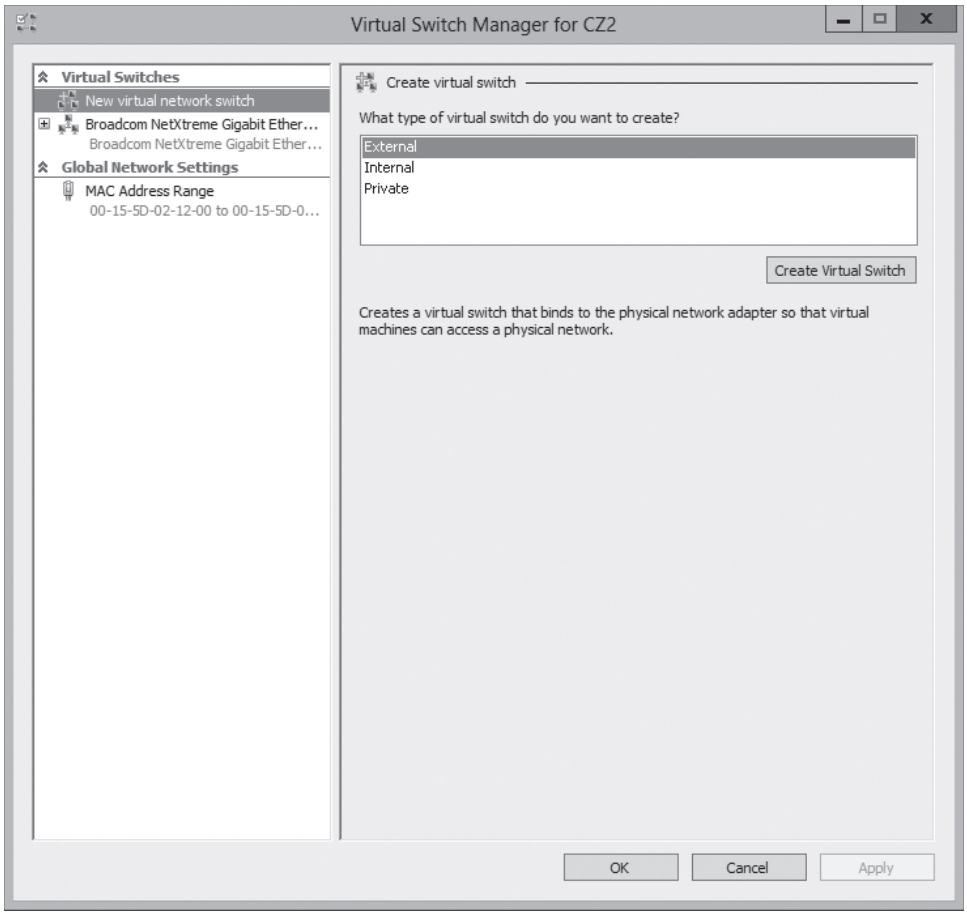
CREATE A NEW VIRTUAL SWITCH

GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges.

1. From the Tools menu in the *Server Manager* window, select Hyper-V Manager. The *Hyper-V Manager* console appears.
2. In the left pane, select a Hyper-V server.
3. From the Actions pane, select Virtual Switch Manager. The *Virtual Switch Manager* dialog box for the Hyper-V server appears, as shown in Figure 9-1.

Figure 9-1

The Virtual Switch Manager dialog box



4. In the Create virtual switch box, select one of the following switch types and click Create Virtual Switch. The *Virtual Switch Properties* page appears.
   - External: The virtual switch is bound to networking protocol stack in the host operating system and connected to a physical network interface adapter

in the Hyper-V server. Virtual machines running on the server's parent and child partitions can access the physical network to which the physical adapter is connected.

- Internal: An ***internal network switch*** is bound to a separate instance of the networking protocol stack in the host operating system, independent from the physical network interface adapter and its connected network. Virtual machines running on the server's parent and child partitions can access the virtual network implemented by the virtual switch, and the host operating system on the parent partition can access the physical network through the physical network interface adapter, but the virtual machines on the child partitions cannot access the physical network through the physical adapter.
- Private: A ***private network switch*** exists only in the Hyper-V server and is accessible only to the virtual machines running on the child partitions. The host operating system on the parent partition can still access the physical network through the physical network interface adapter, but it cannot access the virtual network created by the virtual switch.

5. Configure the following options, if desired:
   - Allow management operating system to share this network adapter: Selected by default when you create an external virtual switch, clearing this check box excludes the host operating system from the physical network, while allowing access to the child virtual machines.
   - Enable single root I/O virtualization (SR-IOV): This option enables you to create an external virtual switch that is associated with a physical network adapter capable of supporting SR-IOV. This option is only available when creating a new virtual switch; you cannot modify an existing switch to use this option.
   - Enable virtual LAN identification for management operating system: If your host computer is connected to a physical switching infrastructure that uses virtual LANs (VLANs) to create separate subnets, you can select the check box and enter a VLAN identifier to associate the virtual switch with a particular VLAN on your physical network.
6. Click OK. The new virtual switch appears in the left pane, in the list of virtual switches.

You can proceed to create additional virtual switches as needed. You can create only one switch for each physical network adapter in the computer, but you can create multiple internal or private switches, to create as many virtual networks as you need.

CONFIGURING MAC ADDRESSES

Every network interface adapter has a Media Access Control (MAC) address (sometimes called a *hardware address*) that uniquely identifies the device on the network. On physical network adapters, the MAC is assigned by the manufacturer and permanently entered in the adapter's firmware. The MAC address is a 6-byte hexadecimal value—the first 3 bytes is an organizationally unique identifier (OUI) that specifies the manufacturer, and the last 3 bytes identifies the adapter itself.

The MAC address is essential to the operation of a LAN, so the virtual network adapters on a Hyper-V server require them. The server has at least one real MAC address, provided in its physical network adaptor, but Hyper-V cannot use that one address for all of the virtual adapters connecting virtual machines to the network.

To provide MAC addresses for the virtual adapters, Hyper-V creates a pool of addresses and assigns addresses from this pool to virtual machines as you create them. To view or modify the MAC address pool for the Hyper-V server, you open the *Virtual Switch Manager* and select *MAC Address Range* under *Global Network Settings*.

The first 3 bytes of MAC address range are always 00-15-5D, which is an OUI registered by Microsoft. The fourth and fifth bytes of the MAC address are the last 2 bytes of the IP address assigned to the server's physical network adapter, converted to hexadecimal notation. The sixth and last byte of the MAC address contains the range of values from 00 to FF, which provides 256 possible addresses.

The Hyper-V server assigns the MAC addresses to the network adapters in virtual machines as you create the adapters. The adapters retain their MAC addresses permanently, or until the adapter is removed from the virtual machine. The server reclaims any unused addresses and reuses them.

The default pool of 256 addresses is expected to be sufficient for most Hyper-V virtual machine configurations, but if it is not, you can modify the *Minimum* and *Maximum* values to enlarge the pool. To prevent address duplication, you should change the second to last byte only, by making it into a range of addresses like the last byte.

WARNING When you modify the MAC address pool, and you have other Hyper-V servers on your network, you must be careful not to create the opportunity for duplicate MAC addresses to occur, or networking problems can result.

For example, the range illustrated in the figure provides 256 addresses with the following values:

```
00-15-1D-02-12-00 to 00-15-1D-02-12-FF
```

modifying only the least significant digit, as in the following values, increases the pool from 256 to 4,096.

```
00-15-1D-02-10-00 to 00-15-1D-02-1F-FF
```

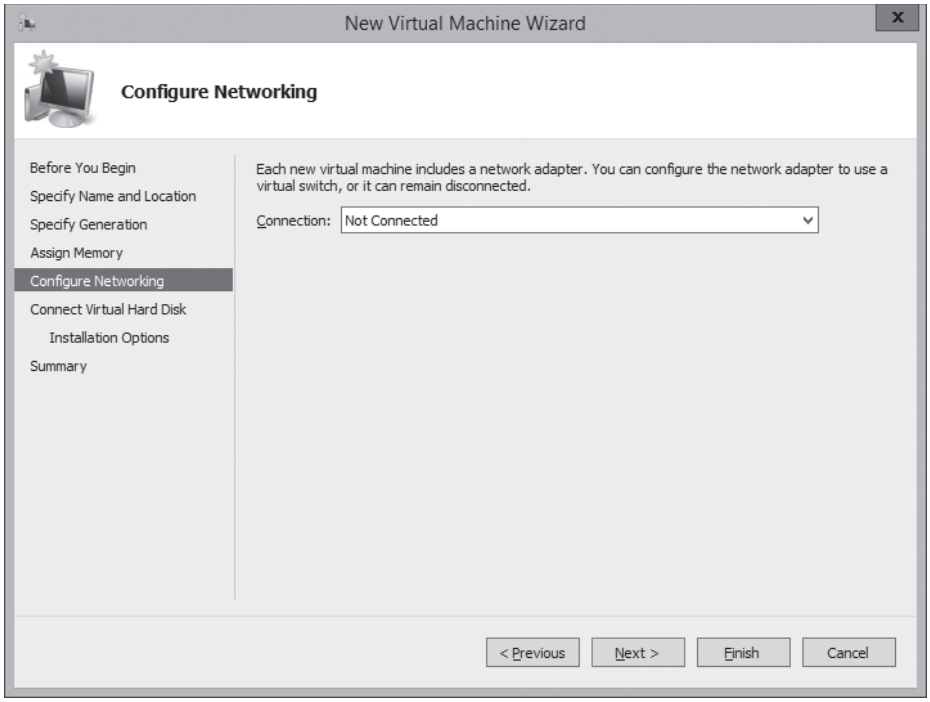
Creating Virtual Network Adapters

After you create virtual switches in Hyper-V Manager, you can connect virtual machines to them by creating and configuring virtual network adapters.

When you create a new virtual machine, the default configuration includes one virtual network adapter. The *New Virtual Machine Wizard* includes a *Configure Networking* page, as shown in Figure 9-2, on which you can select one of the virtual switches you created.

Figure 9-2

The *Configure Networking* page in the New Virtual Machine Wizard



If you create only the default external virtual switch when installing Hyper-V, then connecting a virtual machine to the switch joins the system to the physical network.

USING SYNTHETIC ADAPTORS AND EMULATED ADAPTERS

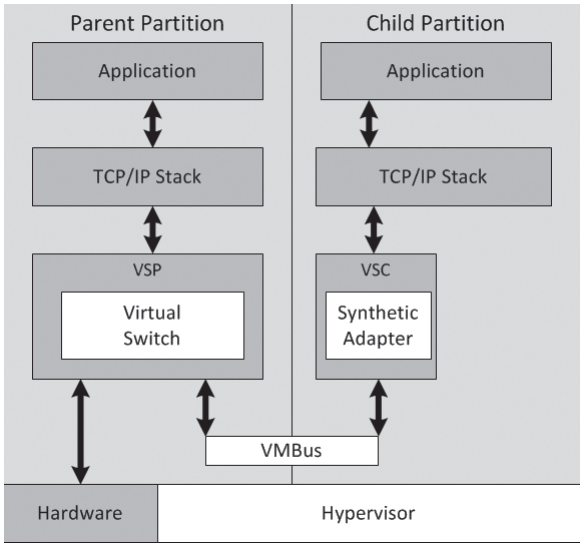
Selecting the *Network Adapter* option in the *Add Hardware* page creates what is known in Hyper-V terminology as a *synthetic network adapter*. Hyper-V supports two types of network and storage adapters: synthetic and emulated (sometimes called *legacy*).

A ***synthetic adapter*** is a virtual device that does not correspond to a real-world product. Synthetic devices in a virtual machine running on a child partition communicate with the parent partition using a high-speed conduit called the ***VMBus***.

The virtual switches you create in Hyper-V reside in the parent partition and are part of a component called the network ***Virtualization Service Provider (VSP)***. The synthetic network adapter in the child partition is a ***Virtualization Service Client (VSC)***. The VSP and the VSC are both connected to the VMBus, which provides interpartition communications, as shown in Figure 9-3. The VSP, in the parent partition, provides the VSC, in the child partition, with access to the physical hardware in the host computer, that is, the physical network interface adaptor.

Figure 9-3

Synthetic network adapters communicate using the VMBUS

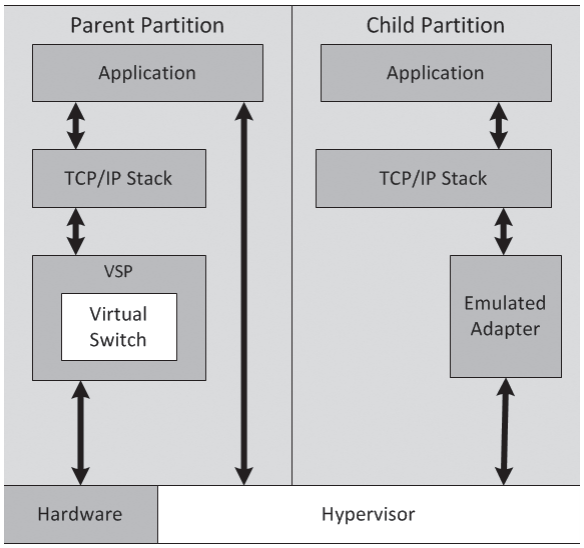


Because they have access to the hardware through the VMBus, synthetic adapters provide a much higher level of performance than the alternative—an emulated adapter. Synthetic adapters are implemented as part of the guest integration services package that run on supported guest operating systems. The one main drawback of synthetic network adapters is that they are not operational until the operating system is loaded on the virtual machine.

An ***emulated adapter*** (sometimes called a ***legacy adapter***) is a standard network adapter driver that communicates with the parent partition by making calls directly to the hypervisor, which is external to the partitions, as shown in Figure 9-4. This communication method is substantially slower than the VMBus used by the synthetic network adapters, and is therefore less desirable.

Figure 9-4

Emulated network adapters communicate using the hypervisor



To install an emulated adapter, you use the same procedure described previously, except that you select *Legacy Network Adapter* in the *Add Hardware* list. Unlike synthetic adapters, emulated adapters load their drivers before the operating system, so you can boot the virtual machine using the Preboot eXecution Environment (PXE) and deploy an operating system over the network.

This is one of the only scenarios in which an emulated adapter is preferable to a synthetic adapter. The other is when you install an operating system on your virtual machines that does not have a guest integration services package available for it.

CONFIGURING HARDWARE ACCELERATION SETTINGS

Some physical network interface adapters have features that are designed to improve performance by offloading certain functions from the system processor to components built into the adapter itself. Hyper-V includes support for some of these features, as long as the hardware in the physical network adapter supports them properly.

When you expand a network adapter in the Settings dialog box of a VM, you gain access to the Hardware Acceleration page. On this page, you can configure the following hardware acceleration settings:

- Enable virtual machine queue: Virtual machine queue (VMQ) is a technique that stores incoming packets intended for virtual machines in separate queues on the physical network adapter and delivers them directly to the VMs, bypassing the processing normally performed by the virtual switch on the parent partition.
- Enable IPsec task offloading: This setting uses the components on the network adapter to perform some of the cryptographic functions required by IPsec. You can also specify maximum number of security associations you want the adapter to be able to calculate.
- Single-root I/O virtualization: This setting enables the virtual adapter to use the SR-IOV capabilities of the physical adapter.

CONFIGURING ADVANCED NETWORK ADAPTER FEATURES

The Advanced Features page provides additional options for supporting network adapter capabilities, as follows:

- Static MAC address: By default, virtual network adapters receive a dynamically assigned MAC address from the Hyper-V server. However, you can also opt to create a static MAC address, by using this option. The only requirement is that no other adapter, virtual or physical, on the same network uses the same address.
- Enable MAC address spoofing: When enabled, the port in the virtual switch to which the virtual network adapter is connected can send and receive packets that contain any MAC address. The virtual switch port can also learn of new MAC addresses and add them in its forwarding table.
- Enable DHCP guard: This option prevents the adapter from processing messages sent by rogue DHCP servers.
- Port mirroring mode: This option enables the adapter to forward all the packets it receives over the network to another virtual adapter for analysis using an application such as Network Monitor.
- NIC teaming: This option enables the adapter to add its bandwidth to other adapters in the same guest operating system in a NIC teaming arrangement.

Configuring NIC Teaming in a Virtual Network Environment

As explained in objective 1.2, "Configuring Servers," NIC teaming is a Windows feature that enables administrators to join multiple network adapters into a single virtual adapter, for performance enhancement or fault tolerance purposes. Hyper-V virtual machines can also take advantage of NIC teaming, but they are limited to teams of only two, as opposed to the host operating system, which can have teams of up to 64 NICs.

To use NIC teaming in Hyper-V, you must complete three basic tasks, as follows:

- Create the NIC team in the Windows Server 2012 R2 R2 host operating system.
- In Hyper-V Manager, create an external virtual switch using the NIC team.
- Configure the virtual network adapter in a VM to connect to the virtual switch representing the NIC team.

CREATING THE NIC TEAM

NIC teams must consist of physical network interface adapters, so before you can use a NIC team in a virtual machine, you must create it in the host operating system. After installing two NICs in the host computer, you can create a NIC team with Server Manager in the usual manner, using the settings shown in Figure 9-5. Creating the team installs the Microsoft Network Adapter Multiplexor Driver, which appears as one of the components of the network connection representing the team.

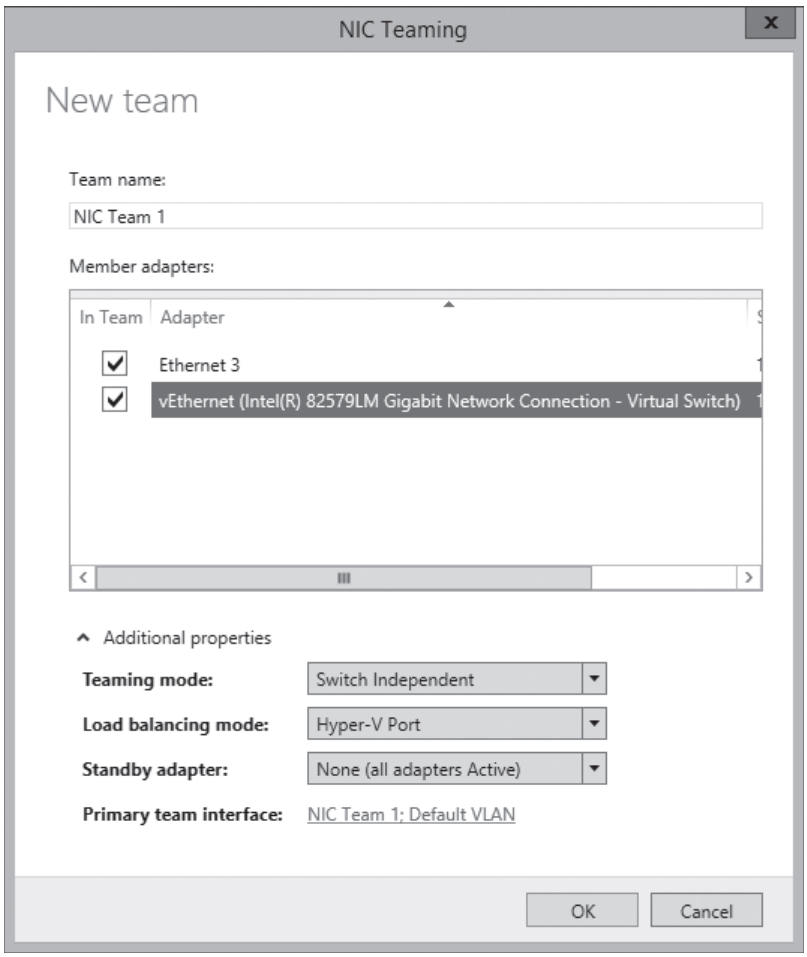


Figure 9-5

The NIC Teaming dialog box

CREATING THE TEAM VIRTUAL SWITCH

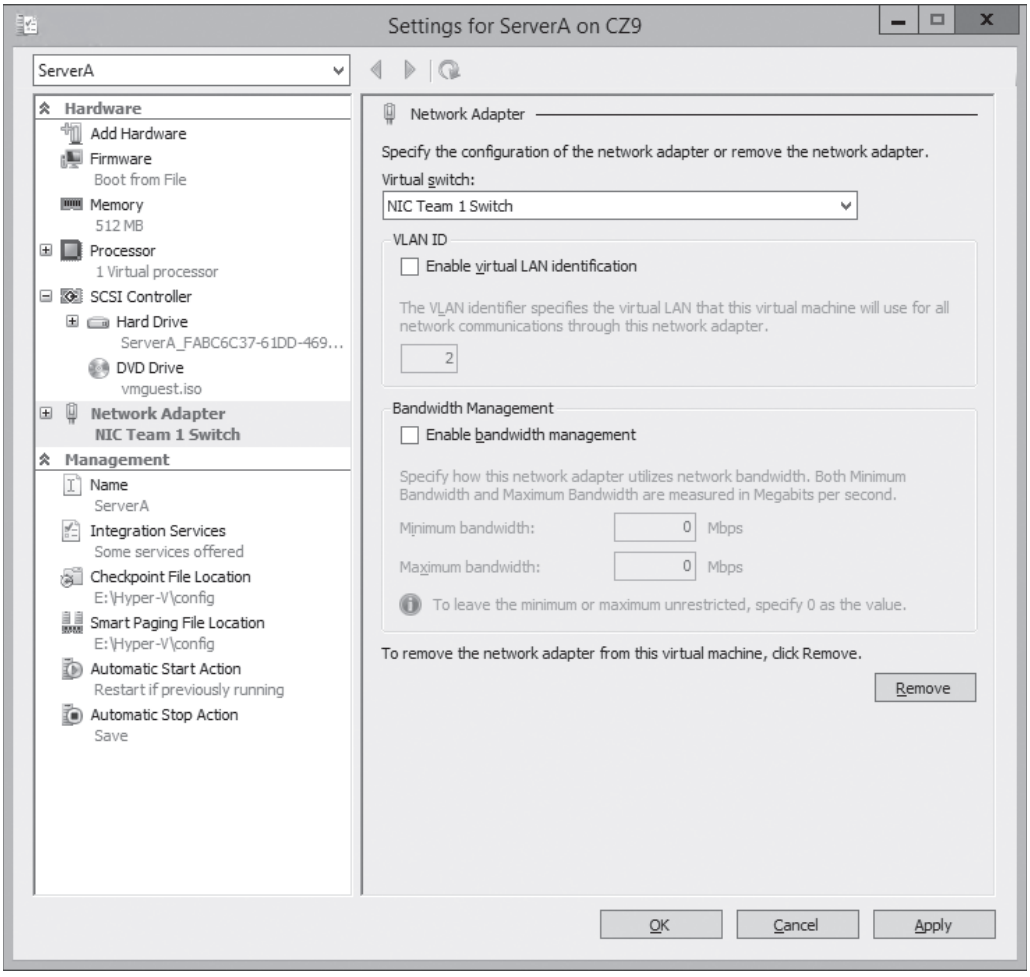
Once you have created the NIC team, you can open the Virtual Switch Manager and create a new virtual switch by selecting the External network option and choosing Microsoft Network Adapter Multiplexor Driver from the drop-down list.

CONFIGURING A NIC TEAM VIRTUAL NETWORK ADAPTER

To configure a virtual machine to use a NIC team, you must use the Settings dialog box to modify the properties for a virtual network adapter, configuring it to use the team switch you created in the previous section, as shown in Figure 9-6.

Figure 9-6

The Network Adapter settings for a NIC team adapter



Finally, you must open the Advanced Features page for the network adapter and select the Enable the network adapter to be part of a team in the guest operating system checkbox. At this point, the NIC team is operational for the virtual machine. You can unplug one of the network cables, and the system will maintain its connection to the network.

Creating Virtual Network Configurations

Hyper-V makes it possible to extend virtually any existing physical network configuration into its virtual space, or create a completely separated and isolated network within the Hyper-V environment.

The basic default configuration of a Hyper-V virtual machine connects its network adapter to an external virtual switch, thus attaching the guest operating system on the virtual machine to the outside network. The virtual machine can then take advantage of services running on the outside network and send traffic through routers to other networks, including the Internet.

This type of arrangement can enable you to consolidate many physical servers into virtual machines on a single Hyper-V server, by providing them all with access to the entire network. There is no distinction here between the physical network and the virtual one in the Hyper-V space.

CREATING AN ISOLATED NETWORK

For testing and evaluation purposes, or for classroom situations, you might create isolated network environments. By creating internal or private virtual switches, you can create a network that exists only within the Hyper-V space, with or without the parent partition included.

An isolated network such as this suffers from the weaknesses of its strengths. If you want to install the guest operating systems using Windows Deployment Services or configure the virtual machines using DHCP, you must install and configure these services on your private network. The guest operating systems also do not have access to the Internet, which prevents them from downloading operating system updates. Again, you must deploy appropriate substitutes on the private network.

To provide your systems with updates, install two network adapters on each of your virtual machines, by connecting one to a private switch and one to an external switch. This procedure enables the virtual machines to access the Internet and the private network.

Another method for creating an isolated network is to use virtual LANs (VLANs). This is particularly helpful if you have virtual machines on different Hyper-V servers that you want to add to the isolated network. By connecting the network adapters to an external switch and configuring them with the same VLAN identifier, you can create a network within a network, which isolates the VLAN from other computers. You can, for example, deploy a DHCP server on your VLAN without it interfering with the other DHCP servers in your production environment.

■ Business Case Scenarios

Scenario 9-1: Creating a New Virtual Switch

You need all VMs networked to each other, plus to the host operating system. Only the host operating system will be connected to the external network. What do you do?

Scenario 9-2: Configuring Advanced Network Adapter Features

You are concerned about your VMs receiving rogue DHCP servers. How can you prevent this from happening?

Configuring IPv4 and IPv6 Addressing

LESSON 10

■ Understanding IPv4 Addressing

THE BOTTOM LINE

Many enterprise administrators are so comfortable working with IPv4 addresses that they are hesitant to change. Network Address Translation (NAT) and Classless Inter-Domain Routing (CIDR) have been excellent stopgaps to the depletion of the 32-bit IP address space for years, and many would like to see them continue as such. However, the IPv6 transition, long a specter on the distant horizon, is now suddenly approaching at frightening speed, and it is time for administrators not familiar with the new technologies to catch up—or be left behind.

The IPv4 address space consists of 32-bit addresses, notated as four 8-bit decimal values from 0 to 255, separated by periods, as in the example 192.168.43.100. This is known as dotted decimal notation, and the individual 8-bit decimal values are called *octets* or *bytes*.

Each address consists of *network bits*, which identify a network, and *host bits*, which identify a particular device on that network. To differentiate the network bits from the host bits, each address must have a subnet mask.

A ***subnet mask*** is another 32-bit value consisting of binary 1 bits and 0 bits. When compared to an IP address, the bits corresponding to the 1's in the mask are the network bits, whereas the bits corresponding to the 0's are the host bits. Thus, if the 192.168.43.100 address mentioned earlier has a subnet mask of 255.255.255.0 (which in binary form is 11111111.11111111.11111111.00000000), the first three octets (192.168.43) identify the network and the last octet (100) identifies the host.

IPv4 Classful Addressing

Because the subnet mask associated with IP addresses can vary, so can the number of bits used to identify the network and the host.

The original Internet Protocol (IP) standard defines three classes of IP addresses, which provide support for networks of different sizes, as shown in Figure 10-1.

Figure 10-1
The three IPv4 address classes

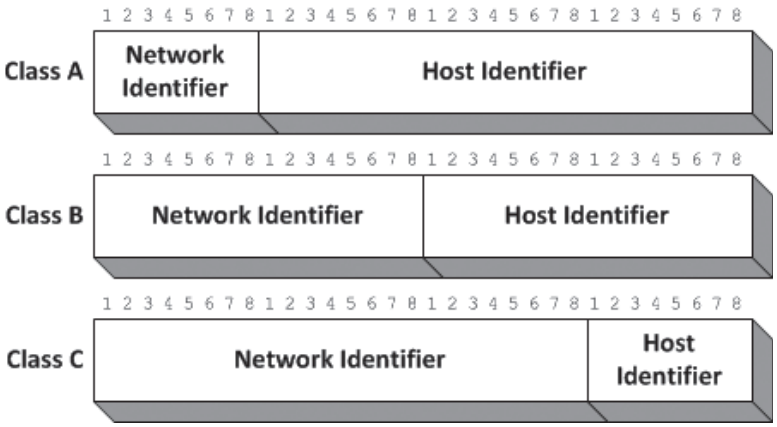


Table 10-1 lists the number of networks and hosts supported by each address class.

Table 10-1
IPv4 Address Classes

IP Address Class	Class A	Class B	Class C
First bit values (binary)	0	10	110
First byte value (decimal)	0–127	128–191	192–223
Number of network identifier bits	8	16	24
Number of host identifier bits	24	16	8
Number of possible networks	126	16,384	2,097,152
Number of possible hosts	16,777,214	65,534	254

The *First bit values (binary)* row in Table 10-1 specifies the values that the first 1, 2, or 3 bits of an address in each class must have. Early TCP/IP implementations used these bit values instead of a subnet mask to determine the class of an address. The binary values of the first bits of each address class limit the possible decimal values for the first byte of the address. For example, because the first bit of Class A addresses must be 0, the possible binary values of the first byte in a Class A address range from 00000000 to 01111111, which in decimal form are values ranging from 1 to 127. Thus, when you see an IP address in which the first byte is a number from 1 to 127, you know that this is a Class A address.

Classless Inter-Domain Routing

When IP was developed, no one imagined that the 32-bit address space would ever be exhausted. In the early 1980s, no networks had 65,536 computers, never mind 16 million, and no one worried about the wastefulness of assigning IP addresses based on these classes.

Because of that wastefulness, classful addressing was gradually obsolesced by a series of subnetting methods, including variable-length subnet masking (VLSM) and eventually ***Classless Inter-Domain Routing (CIDR)***. CIDR is a subnetting method that enables you

to place the division between the network bits and the host bits anywhere in the address, not just between octets. This makes it possible to create networks of almost any size.

CIDR also introduces a new notation for network addresses. A standard dotted-decimal address representing the network is followed by a forward slash and a numeral specifying the size of the network-identifying prefix. For example, 192.168.43.0/24 represents a single Class C address that uses a 24-bit network identifier, leaving the other 8 bits for up to 254 host identifiers. Each of those hosts would receive an address from 192.168.43.1 to 192.168.43.254, using the subnet mask 255.255.255.0.

However, by using CIDR, you can subnet this address further by allocating some of the host bits to create subnets. To create subnets for four offices, for example, you can take two of the host identifier bits, changing the network address in CIDR notation to 192.168.43.0/26. Because the network identifier is now 26 bits, the subnet mask for all four networks is now 11111111.11111111.11111111.11000000 in binary form, or 255.255.255.192 in standard decimal form.

IPv4 Subnetting

In most cases, enterprise administrators use addresses in one of the private IP address ranges to create the subnets they need. If you are building a new enterprise network from scratch, you can choose any private address block and make things easy on yourself by subnetting along the octet boundaries.

For example, you can take the 10.0.0.0/8 private IP address range and use the entire second octet as a subnet ID. This enables you to create up to 256 subnets with as many as 65,536 hosts on each one. The subnet masks for all the addresses on the subnets will be 255.255.0.0, and the network addresses will proceed as follows:

- 10.0.0.0/16
- 10.1.0.0/16
- 10.2.0.0/16
- 10.3.0.0/16
- . . .
- 10.255.0.0/16

Of course, when you are working on an existing network, the subnetting process will likely be more difficult. You might, for example, be given a relatively small range of addresses and be asked to create a certain number of subnets out of them. To do this, use the following procedure.

CALCULATE IPV4 SUBNETS

1. Determine how many subnet identifier bits you need to create the required number of subnets.
2. Subtract the subnet bits you need from the host bits and add them to the network bits.
3. Calculate the subnet mask by adding the network and subnet bits in binary form and converting the binary value to decimal.
4. Take the least significant subnet bit and the host bits, in binary form, and convert them to a decimal value.
5. Increment the network identifier (including the subnet bits) by the decimal value you calculated to determine the network addresses of your new subnets.

By using the same example from earlier in this lesson, if you take the 192.168.43.0/24 address and allocate two extra bits for the subnet ID, you end up with a binary subnet mask value of 11111111.11111111.11111111.11000000 (255.255.255.192 in decimal form, as noted earlier).

The least significant subnet bit plus the host bits gives you a binary value of 1000000, which converts to a decimal value of 64. Therefore, if we know that the network address of your first subnet is 192.168.43.0, the second subnet must be 192.168.43.64, the third 192.168.43.128, and the fourth 192.168.43.192.

Supernetting

In addition to simplifying network notation, CIDR also makes possible a technique called *IP address aggregation* or *supernetting*, which can help reduce the size of Internet routing tables. A ***supernet*** combines contiguous networks that all contain a common CIDR prefix. When an organization possesses multiple contiguous networks that can be expressed as a supernet, listing those networks in a routing table using only one entry, instead of many, becomes possible.

For example, if an organization has the following five subnets, standard practice would be to create a separate routing table entry for each one:

- 172.16.43.0/24
- 172.16.44.0/24
- 172.16.45.0/24
- 172.16.46.0/24
- 172.16.47.0/24

To create a supernet encompassing all five of these networks, you must isolate the bits they have in common. When you convert the network addresses from decimal to binary, you get the following values:

- 172.16.43.0 **10101100.00010000.00101**011.00000000
- 172.16.44.0 **10101100.00010000.00101**100.00000000
- 172.16.45.0 **10101100.00010000.00101**101.00000000
- 172.16.46.0 **10101100.00010000.00101**110.00000000
- 172.16.47.0 **10101100.00010000.00101**111.00000000

In binary form, you can see that all five addresses have the same first 21 bits. Those 21 bits become the network identifier of the supernet address, as follows:

```
10101100.00010000.00101
```

After zeroing out the host bits to form the network address and converting the binary number back to decimal form, as follows, the resulting supernet address is 172.16.40.0/21.

```
10101100.00010000.00101000.00000000  172.16.40.0/21
```

This one network address can replace the original five in routing tables duplicated throughout the Internet. Obviously, this is just an example of a technique that you can use to combine dozens or even hundreds of subnets into single routing table entries.

Assigning IPv4 Addresses

In addition to understanding how IP addressing works, you must be familiar with the methods for deploying IP addresses to the computers on a network.

You have three alternatives for assigning IPv4 addresses:

- Manual configuration
- Dynamic Host Configuration Protocol (DHCP)
- Automatic Private IP Addressing (APIPA)

The following sections cover the advantages and disadvantages of these methods.

MANUAL IPV4 ADDRESS CONFIGURATION

Configuring a TCP/IP client manually is not terribly difficult, nor is it very time-consuming. Most operating systems provide a graphical interface that enables you to enter an IPv4 address, a subnet mask, and various other TCP/IP configuration parameters. To configure IP address settings in Windows Server 2012 R2, use the following procedure.

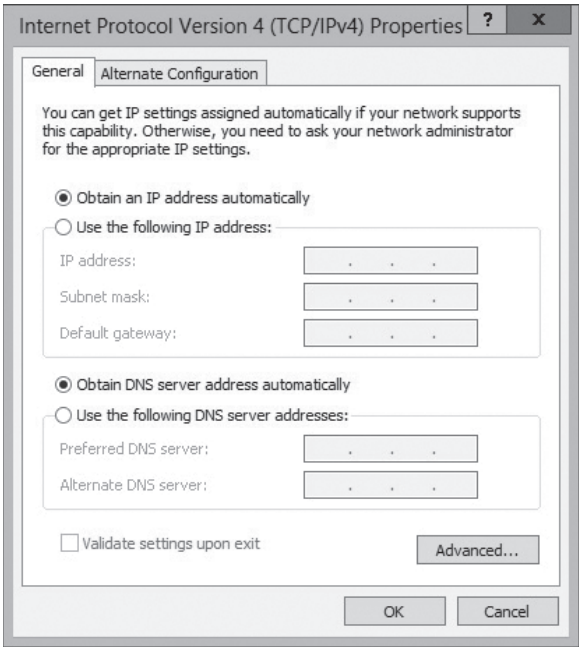
CONFIGURE IP ADDRESS SETTINGS

GET READY. Log on to Windows Server 2012 R2 using an account with Administrative privileges.

1. In the left pane of the *Server Manager* window, click the Local Server icon. The Properties tile for the server appears.
2. In the *Properties* tile, click the Ethernet hyperlink. The *Network Connections* window appears.
3. Right-click the Ethernet icon and, from the context menu, select Properties. The *Ethernet Properties* sheet appears.
4. Select the Internet Protocol Version 4 (TCP/IPv4) component and click Properties. The *Internet Protocol Version 4 (TCP/IPv4) Properties* sheet appears, as shown in Figure 10-2.

Figure 10-2

The *Internet Protocol Version 4 (TCP/IPv4) Properties* sheet



5. Select the Use the following IP address radio button and configure the following parameters with appropriate values:
   - IP address specifies the IP address on the local subnet that identifies the network interface in the computer.
   - Subnet mask specifies the mask associated with the local subnet.
   - Default gateway specifies the IP address of a router on the local subnet, which the system uses to access destinations on other networks.
6. Select the Use the following DNS server addresses radio button and configure the following parameter with appropriate values:
   - Preferred DNS server specifies the IP address of the DNS server that the system uses to resolve host names into IP addresses.
   - Alternate DNS server specifies the IP address of the DNS server that the system uses to resolve host names into IP addresses when the preferred DNS server is not available.
7. Click OK twice to close the *Internet Protocol Version 4 (TCP/IPv4)* and *Ethernet* Properties sheets.

CLOSE the *Network Connections* window.

You also can configure the IP address settings on a Windows system from the command line, using the Netsh.exe program. However, the big problem with manual configuration is that a task requiring two minutes for one workstation requires several hours for 100 workstations and several days for 1,000. Manually configuring all but the smallest networks is highly impractical, and not just for reasons of time. You also need to track the IPv4 addresses you assign and make sure each system has a unique address. This can end up being a logistical nightmare, which is why few network administrators choose this option.

DYNAMIC HOST CONFIGURATION PROTOCOL

The ***Dynamic Host Configuration Protocol (DHCP)*** is a client/server application as well as an Application-layer protocol that enables you to allocate IP addresses dynamically from a pool. Computers equipped with DHCP clients automatically contact a DHCP server when they start, and the server assigns them unique addresses and all the other configuration parameters the TCP/IP client requires.

The DHCP server leases addresses to clients, and after a predetermined interval, each client either renews its address or releases it back to the server for reallocation. DHCP not only automates the address assignment process, but it also keeps track of the addresses it assigns, preventing any address duplication on the network.

AUTOMATIC PRIVATE IP ADDRESSING (APIPA)

Automatic Private IP Addressing (APIPA) is the name assigned by Microsoft to a DHCP failover mechanism used by all current Microsoft Windows operating systems. On Windows computers, the DHCP client is enabled by default. If, after several attempts, a system fails to locate a DHCP server on the network, APIPA takes over and automatically assigns an address on the 169.254.0.0/16 network to the computer. The system then uses the Address Resolution Protocol (ARP) to ensure that no other computer on the local network is using the same address.

For a small network that consists of only a single LAN, APIPA is a simple and effective alternative to installing a DHCP server. However, for installations consisting of multiple LANs, with routers connecting them, you must take more positive control over the IP address assignment process. This usually means deploying one or more DHCP servers in some form.

■ Understanding IPv6 Addressing

THE BOTTOM LINE

As most administrators know, IPv6 is designed to increase the size of the IP address space, thus providing addresses for many more devices than IPv4. The 128-bit address size of IPv6 allows for 2^{128} possible addresses, an enormous number that works out to over 54 million addresses for each square meter of the Earth's surface.

In addition to providing more addresses, IPv6 also reduces the size of the routing tables in the routers scattered around the Internet. This is because the size of the addresses provides for more than the two levels of subnetting currently possible with IPv4.

Introducing IPv6

IPv6 addresses are different from IPv4 addresses in many ways other than length.

Instead of the four 8-bit decimal numbers separated by periods that IPv4 uses, IPv6 addresses use a notation called *colon-hexadecimal format*, which consists of eight 16-bit hexadecimal numbers, separated by colons, as follows:

```
XX:XX:XX:XX:XX:XX:XX:XX
```

Each X represents 8 bits (or 1 byte), which in hexadecimal notation is represented by two characters, as in the following example:

```
21cd:0053:0000:0000:e8bb:04f2:003c:c394
```

IPv6 Address Types

IPv6 has no broadcast transmissions, and therefore no broadcast addresses, unlike IPv4.

IPv6 supports three address types:

- **Unicast:** provides one-to-one transmission service to individual interfaces, including server farms sharing a single address. IPv6 supports several types of unicast addresses, including global, link-local, and unique local, terms that identify the scope of the address. Each type of unicast has a different ***format prefix (FP)***, a sequence of bits that identifies the type, just as an IPv4 address uses a sequence of bits to identify its class.
- **Multicast:** provides one-to-many transmission service to groups of interfaces identified by a single multicast address.
- **Anycast:** provides one-to-one-of-many transmission service to groups of interfaces, only the nearest of which (measured by the number of intermediate routers) receives the transmission.

Subnet IDs

The organization then has the 16-bit subnet ID with which to create an internal subnet hierarchy, if desired. Some possible subnetting options are as follows:

- **One-level subnet:** By setting all subnet ID bits to 0, all the computers in the organization are part of a single subnet. This option is suitable only for smaller organizations.

- **Two-level subnet:** By creating a series of 16-bit values, you can split the network into as many as 65,536 subnets. This is the functional equivalent of IPv4 subnetting, but with a much larger subnet address space.
- **Multi-level subnet:** By allocating specific numbers of subnet ID bits, you can create multiple levels of subnets, sub-subnets, and sub-sub-subnets, suitable for an enterprise of almost any size.

In one example, designed to support a large international enterprise, you could split the subnet ID as follows:

- **Country (4 bits):** Creates up to 16 subnets representing countries in which the organization has offices
- **State (6 bits):** Creates up to 64 sub-subnets within each country, representing states, provinces, or other geographical divisions
- **Office (2 bits):** Creates up to four sub-sub-subnets within each state or province, representing offices located in various cities
- **Department (4 bits):** Creates up to 16 sub-sub-sub-subnets within each office, representing the various departments or divisions.

To create a subnet ID for a particular office, you need to assign values for each field. To use the value 1 for the United States, the Country bits would be as follows:

```
0001------------
```

To create a subnet for an office in Alaska, you can use a value of 49 in the State field, which in binary form would appear as follows:

```
----110001------
```

For the second office in Alaska, use the value 2 for Office bits, as follows:

```
----------10----
```

For the Sales department in the office, use the value 9 for the Department bits, as follows:

```
------------1001
```

The resulting value for the subnet ID, in binary form, would therefore be as follows:

```
0001110001101001
```

In hexadecimal form, that would be 1c69.

Because the organization that owns the prefix wholly controls the subnet ID, enterprise administrators can adjust the number of levels in the hierarchy and the number of bits dedicated to each level as needed.

Assigning IPv6 Addresses

The processes by which you assign IPv6 addresses to network computers are similar to those in IPv4.

As with IPv4, a Windows computer can obtain an IPv6 address by three possible methods:

- **Manual allocation:** A user or administrator manually supplies an address and other information for each network interface.
- **Self-allocation:** The computer creates its own address using a process called stateless address autoconfiguration.
- **Dynamic allocation:** The computer solicits and receives an address from a Dynamic Host Configuration Protocol (DHCPv6) server on the network.

MANUAL IPV6 ADDRESS ALLOCATION

For the enterprise administrator, manual allocation of IPv6 addresses is even more impractical than in IPv4, because of the length of the addresses involved. However, it is possible, and the procedure for doing so in Windows Server 2012 R2 is the same as that for IPv4, except that you open the *Internet Protocol Version 6 (TCP/IPv6) Properties* sheet.

Because of the difficulties working with IPv6 addresses manually, the following two options are far more prevalent.

STATELESS IPV6 ADDRESS AUTOCONFIGURATION

When a Windows computer starts, it initiates the ***stateless address autoconfiguration*** process, during which it assigns each interface a link-local unicast address. This assignment always occurs, even when the interface is to receive a global unicast address later. The link-local address enables the system to communicate with the router on the link, which provides additional instructions.

DYNAMIC HOST CONFIGURATION PROTOCOL V6

If you deal with a multi-segment network, you need to use unique local or global addresses for internetwork communication, so you need either routers that advertise the appropriate network prefixes or DHCPv6 servers that can supply addresses with the correct prefixes.

The Remote Access role in Windows Server 2012 R2 supports IPv6 routing and advertising, and the DHCP Server role supports IPv6 address allocation.

■ Planning an IP Transition

THE BOTTOM LINE

Many enterprise administrators are so comfortable working with IPv4 addresses that they are hesitant to change. Network Address Translation (NAT) and Classless Inter-Domain Routing (CIDR) have been excellent stopgaps to the depletion of the 32-bit IP address space for years, and many would like to see them continue as such. However, the IPv6 transition, long a specter on the distant horizon, is now suddenly approaching at frightening speed, and it is time for administrators not familiar with the new technologies to catch up—or be left behind.

The networking industry, and particularly the Internet, has made huge investments in IPv4 technologies, and replacing them with IPv6 has had to be a gradual process. In fact, this gradual process was supposed to have begun in earnest in 1998. However, many people treat their IPv4 equipment like household appliances; unless they stop working, replacing them is not necessary. Unfortunately, the day in which that equipment stops working is approaching rapidly. So, while it might not yet be time to embrace IPv6 exclusively, you should have the transition in mind as you design your networks and make your purchasing decisions.

Enterprise administrators can do as they want within the enterprise itself. If all network devices in the organization support IPv6, they can begin to use IPv6 addresses at any time. However, the Internet is still firmly based on IPv4 and will continue to be so for several years. There, an IPv4-to-IPv6 transition must be a gradual project that includes some period of support for both IP versions.

Now and for the immediate future, you must work under the assumption that the rest of the world is using IPv4, and that you must implement a mechanism for transmitting your IPv6 traffic over an IPv4 connection. Eventually, the situation will be reversed. Most of the world will be running IPv6, and the remaining IPv4 technologies will have to transmit their older traffic over new links.

Tunneling

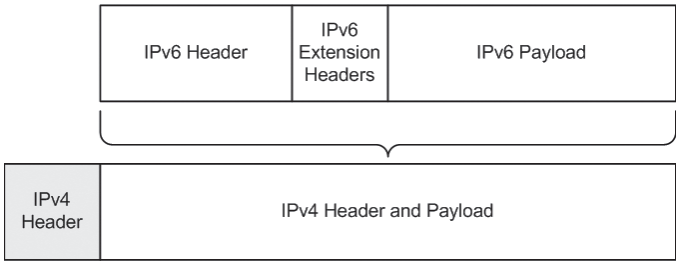
Right now, many network services are IPv4 only, and comparatively few require IPv6. Those IPv6 services are coming, however.

The DirectAccess remote networking feature in Windows Server 2012 R2 and Windows 8.1 is an example of an IPv6-only technology, and much of its complexity is due to the need to establish IPv6 connections over the IPv4 Internet.

Tunneling is the primary method for transmitting IPv6 traffic over an IPv4 network. ***Tunneling***, in this case, is the process by which a system encapsulates an IPv6 datagram within an IPv4 packet, as shown in Figure 10-3. The system then transmits the IPv4 packet to its destination, with none of the intermediate systems aware of the packet's contents.

Figure 10-3

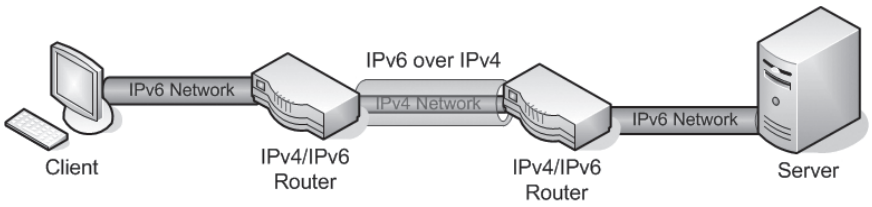
IPv6 traffic encapsulated inside an IPv4 datagram



Tunneling can work in various configurations, depending on the network infrastructure, including router-to-router, host-to-host, router-to-host, and host-to-router. However, the most common configuration is router-to-router, as in the case of an IPv4-only connection between an IPv6 branch office and an IPv6 home office, as shown in Figure 10-4.

Figure 10-4

Two IPv6 networks connected by an IPv4 tunnel



The two routers support both IPv4 and IPv6, and the local networks at each site use IPv6. However, the link connecting the two sites is IPv4 only. By creating a tunnel between the routers in the two offices, using their IPv4 interfaces, they can exchange IPv6 traffic as needed. Computers at either site can send IPv6 traffic to the other site, and the routers are responsible for encapsulating the IPv6 data in IPv4 packets for the trip through the tunnel.

Windows supports several different tunneling methods, both manual and automatic, as described in the following sections.

CONFIGURING TUNNELS MANUALLY

You can manually create semi-permanent tunnels that carry IPv6 traffic through an IPv4-only network. When a computer running Windows Server 2012 R2 or Windows 8.1 is functioning as one end of the tunnel, you can use the following command:

```
netsh interface ipv6 add v6v4tunnel "interface" localaddress remoteaddress
```

In this command, *interface* is a friendly name you want to assign to the tunnel you are creating; *localaddress* and *remoteaddress* are the IPv4 addresses forming the two ends of the tunnel. An example of an actual command would be as follows:

```
netsh interface ipv6 add v6v4tunnel "tunnel"
206.73.118.19 157.54.206.43
```

CONFIGURING TUNNELS AUTOMATICALLY

A number of mechanisms automatically create tunnels over IPv4 connections. These are technologies designed to be temporary solutions during the transition from IPv4 to IPv6. All of them include a mechanism for expressing an IPv4 address in the IPv6 format. The following sections describe IPv4-to-IPv6 transition technologies that Windows supports.

6to4

The ***6to4*** mechanism essentially incorporates the IPv4 connections in a network into the IPv6 infrastructure by defining a method for expressing IPv4 addresses in IPv6 format and encapsulating IPv6 traffic into IPv4 packets.

To enable IPv4 links to function as part of the IPv6 infrastructure, 6to4 translates public IPv4 addresses into IPv6 using the following format:

- **FP:** The 3-bit format prefix is 001 in binary, the standard global unicast value.
- **TLA:** A 13-bit TLA value for a 6to4 address is always 0002 in hexadecimal.
- **V4ADDR:** A 32-bit V4ADDR value contains the IPv4 dotted decimal address, split into four separate octets and converted into hexadecimal form.
- **SLA ID:** Organizations can use a 16-bit SLA ID (or subnet ID) field to create an internal hierarchy of sites or subnets.
- **Interface ID:** This 64-bit field identifies a specific interface on the network.

For example, to convert the IPv4 address 157.54.176.7 into a 6to4 IPv6 address, you begin with 2002 for the FP and TLA fields, and then convert the four decimal values from the IPv4 address into hexadecimal, as follows:

- 157 = 9d
- 54 = 36
- 176 = b0
- 7 = 07

Therefore, you end up with the following IPv6 address:

`2002:9d36:b007:`*subnetID*`:`*interfaceID*

The subnet and interface identifiers use the same values as any other IPv6 link on the network. The encapsulation method is the same as that for a manually created tunnel, with a standard IPv4 header and containing the IPv6 data as the payload. A 6to4 router examines incoming packets and, if it detects the 2002 value in the first block, knows to transmit the packet over the IPv4 interface, using the 32 bits following the 2002 block as the IPv4 address.

ISATAP

Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) is an automatic tunneling protocol used by the Windows workstation operating systems that emulates an IPv6 link using an IPv4 network.

ISATAP also converts IPv4 addresses into an IPv6 Link-layer address format, but it uses a different method than 6to4. An ISATAP address uses the following format:

- The first 64 bits consist of the standard link-local network identifier, the value fe80 following by 48 bits of 0s.

- The first 16-bit block of the interface identifier consists of all 0s, except for the seventh bit, which is set to 1 when the IPv4 address is globally unique, and the eighth bit, which is set to 1 when the IPv4 address identifies a multicast group. In most cases, this block consists of all 0s.
- The second 16-bit block of the interface ID consists of the value 5efe, which represents the concatenated OUI for ISATAP (5e) and the standardized value fe.
- The final 32 bits of the interface identifier consist of the IPv4 address in hexadecimal form.

Therefore, the IPv4 address 157.54.176.7 would have the following as its ISATAP address:

```
fe80:0000:0000:0000:0000:5efe:9d36:b007
```

In compressed form, the address appears as follows:

```
fe80::5efe:9d36:b007
```

ISATAP does not support multicasting, so it cannot locate routers in the usual manner, using the Neighbor Discovery protocol. Instead, the system compiles a potential routers list (PRL) using DNS queries and sends Router Discovery messages to them regularly, using Internet Control Message Protocol version 6 (ICMPv6).

Teredo

To use 6to4 tunneling, both endpoints of the tunnel must have registered IPv4 addresses. However, on many networks, the system that would function as the endpoint is located behind a NAT router, and therefore has an unregistered address. In such a case, the only registered address available is assigned to the NAT router itself, and unless the router supports 6to4 (which many do not), establishing the tunnel is impossible.

Teredo is a mechanism that addresses this shortcoming by enabling devices behind non-IPv6 NAT routers to function as tunnel endpoints. To do this, Teredo encapsulates IPv6 packets within Transport-layer User Datagram Protocol (UDP) datagrams, rather than Network-layer IPv4 datagrams, as 6to4 does.

For a Teredo client to function as a tunnel endpoint, it must have access to a Teredo server, with which it exchanges Router Solicitation and Router Advertisement messages to determine whether the client is located behind a NAT router.

Teredo clients have the most complicated form of IPv6 address yet, which uses the following format:

- **Prefix:** A 32-bit field that identifies the system as a Teredo client. Windows clients use the prefix value 2001:0000, or 2001::/32.
- **Server IPv4:** A 32-bit field containing the IPv4 address of the Teredo server the client uses.
- **Flags:** A 16-bit field, the first bit of which is the Cone flag, set to 1 when the NAT device providing access to the Internet is a cone NAT, which stores the mappings between internal and external addresses and port numbers. The second bit is reserved for future use. The seventh and eighth bits are the Universal/Local and Individual/Group flags, which are both set to 0. The Teredo standard calls for the remaining 12 bits to be set to 0, but Windows assigns a random number to these bits to prevent attackers from attempting to discover the Teredo address.
- **Port:** A 16-bit field that specifies the external UDP port that the client uses for all Teredo traffic, in obscured form. The obscuration of the port number (and the following IPv4 address) helps prevent the NAT router from translating the port as it normally would as part of its packet processing. To obscure the port, the system runs an exclusive OR (XOR) with the value ffff.
- **Client IPv4:** A 32-bit field that specifies the external IPv4 address that the client uses for all Teredo traffic, in obscured form. As with the Port field, the obscuration is the result of converting the IPv4 address to hexadecimal and running an XOR with the value ffffffff.

If, for example, the IPv4 address and port of the Teredo client are both 192.168.31.243:32000, the Teredo server uses the address 157.54.176.7, and the client is behind a cone NAT router, the Teredo address, in standard format, would consist of the elements listed in Table 10-2.

Table 10-2
Standard Teredo Address Format

Element	Description
2001:0000	Standard Teredo prefix
9d36:b007	Server IPv4 address (157.54.176.7) converted to hexadecimal
8000	Flags field with first bit set to 1 and all others 0
82ff	Client UDP port number (32000), converted to hexadecimal (7d00) and XORed with ffff
3f57:e00c	Client IPv4 address (192.168.31.243), converted to hexadecimal (C0a8:1ff3) and XORed with ffffffff

Thus, the final Teredo address is as follows:

```
2001:0000:9d36:b007:8000:82ff:3f57:e00c
```

To initiate communications, a Teredo client exchanges null packets called *bubbles* with the desired destination, using the Teredo servers at each end as intermediaries. The function of the bubble messages is to create mappings for both computers in each other's NAT routers.

■ Business Case Scenarios

Scenario 10-1: Calculating IPv4 Subnets

The enterprise administrator has assigned Arthur the network address 172.16.85.0/25 for the branch office network that he is constructing. Arthur calculates that this gives him 126 (2^7) IP addresses, which is enough for his network, but he has determined that he needs six subnets with at least 10 hosts on each one. How can Arthur subnet the address he has been given to satisfy his needs? What IP addresses and subnet masks will the computers on his branch office network use?

Scenario 10-2: Calculating IPv6 Interface IDs

Ed has three servers running Windows Server 2012 R2, for which he has to configure IPv6 global unicast addresses manually. The MAC addresses for the network interfaces in the three machines are as follows:

- 60-EB-69- 93-5E-E5
- 00-15-5D-02-12-05
- D4-AE-52-BF-C3-2D

What are the EUI-64 interface IDs that Ed will use for these three MAC addresses?

11 LESSON

Deploying and Configuring the Dynamic Host Configuration Protocol (DHCP) Service

■ Understanding DHCP

THE BOTTOM LINE

The ***Dynamic Host Configuration Protocol (DHCP)*** service automatically configures the Internet Protocol (IP) address and other TCP/IP settings on network computers by assigning addresses from a pool (called a *scope*) and reclaiming them when they are no longer in use.

Aside from being a time-consuming chore, manually configuring TCP/IP clients can result in typographical errors that cause addressing conflicts, which interrupt network communications. DHCP prevents these errors and provides many other advantages, including automatic assignment of new addresses when computers are moved from one subnet to another, and automatic reclamation of addresses that are no longer in use.

DHCP consists of three components:

- A *DHCP server application* responds to client requests for TCP/IP configuration settings.
- A *DHCP client* issues requests to servers and applies the TCP/IP configuration settings it receives to the local computer
- A *DHCP communications protocol* defines the formats and sequences of the messages exchanged by DHCP clients and servers

All Microsoft Windows operating systems include DHCP client capabilities, and all the server operating systems (including Windows Server 2012 R2) include Microsoft DHCP Server. Microsoft's DHCP implementation is based on public domain standards published by the Internet Engineering Task Force (IETF) as RFC 2131, "Dynamic Host Configuration Protocol," and RFC 2132, "DHCP Options and BOOTP Vendor Extensions," and is interoperable with other DHCP implementations.

The DHCP standards define three different IP address allocation methods:

- ***Dynamic allocation:*** The DHCP server assigns an IP address to a client computer from a scope for a specified length of time. DHCP servers using dynamic allocation only lease addresses to clients. Each client must periodically renew the lease to continue using the address. If the client allows the lease to expire, the address is returned to the scope for reassignment to another client.

- ***Automatic allocation:*** The DHCP server permanently assigns an IP address to a client computer from a scope. After the DHCP server assigns the address to the client, the only way to change it is to reconfigure the computer manually. Automatic allocation is suitable for networks where you do not often move computers to different subnets. It reduces network traffic by eliminating the periodic lease renewal messages needed for dynamic allocation. In the Windows Server 2012 R2 DHCP server, automatic allocation is essentially dynamic allocation with an indefinite lease.
- ***Manual allocation:*** The DHCP server permanently assigns a specific IP address to a specific computer on the network. In the Windows Server 2012 R2 DHCP server, manually allocated addresses are called *reservations*. You use manually allocated addresses for computers that must have the same IP address at all times, such as Internet web servers that have specific IP addresses associated with their host names in the DNS namespace. Although you can just as easily configure such computers manually, DHCP reservations prevent the accidental duplication of permanently assigned IP addresses.

In addition to IP addresses, DHCP also can provide clients with values for the other parameters needed to configure a TCP/IP client, including a subnet mask, default gateway, and DNS server addresses. The object is to eliminate the need for any manual TCP/IP configuration on a client system. For example, the Microsoft DHCP server includes more than 50 configuration parameters, which it can deliver along with the IP address, even though Windows clients can use only a subset of those parameters.

The RFC 2132 document, "DHCP Options and BOOTP Vendor Extensions," defines an extensive list of parameters that compliant servers should support, and most major DHCP server packages adhere closely to this list. Many of these parameters are designed for use by specific system configurations and are submitted by vendors for inclusion in the standard document.

■ Deploying a DHCP Server

THE BOTTOM LINE DHCP servers operate independently, so you must install the service and configure scopes on every computer that will function as a DHCP server.

The DHCP Server service is packaged as a role in Windows Server 2012 R2, which you can install by using the *Add Roles and Features Wizard*, accessible from the Server Manager console. To install the DHCP Server service on a Windows Server 2012 R2 computer with Server Manager, use the following procedure.

DEPLOY A DHCP SERVER

GET READY. Log on to Windows Server 2012 R2 using an account with Administrative privileges.

1. In the *Server Manager* window, click Manage and then click Add Roles and Features. The *Add Roles and Features Wizard* appears, displaying the *Before you begin* page.
2. Click Next. The *Select Installation Type* page appears.
3. Leave the *Role-based or feature-based installation* radio button selected and click Next. The *Select Destination Server* page appears.
4. Select the server on which you want to install the roles and/or features and click Next. The *Select Server Roles* page appears.

5. Select the DHCP Server check box. An *Add features that are required for DHCP Server* dialog box appears.
6. Click Add Features, and then click Next. The *Select features* page appears.
7. Click Next. The *DHCP Server* page appears.
8. Click Next. The *Confirm installation selections* page appears.
9. Click Install. The *Installation progress* page appears as the wizard installs the role.
10. Click Close. The wizard closes.

CLOSE Server Manager.

When you install the DHCP Server role on a computer that is a member of an AD DS domain, the DHCP Server is automatically authorized to allocate IP addresses to clients that are also members of the same domain. If the server is not a domain member when you install the role, and you join it to a domain later, you must manually authorize the DHCP server in the domain. To do this, right-click the server node in the DHCP console and, from the context menu, select *Authorize*.

After installing the DHCP Server role, you must configure the service by creating a scope before it can serve clients.

Creating a Scope

A *scope* is a range of IP addresses on a particular subnet that a DHCP server has selected for allocation.

In Windows Server versions prior to 2012, you can create a scope as you install the DHCP Server role. However, in Windows Server 2012 and Windows Server 2012 R2, the procedures are separate. To create a scope using the DHCP snap-in for Microsoft Management Console (MMC), use the following procedure.

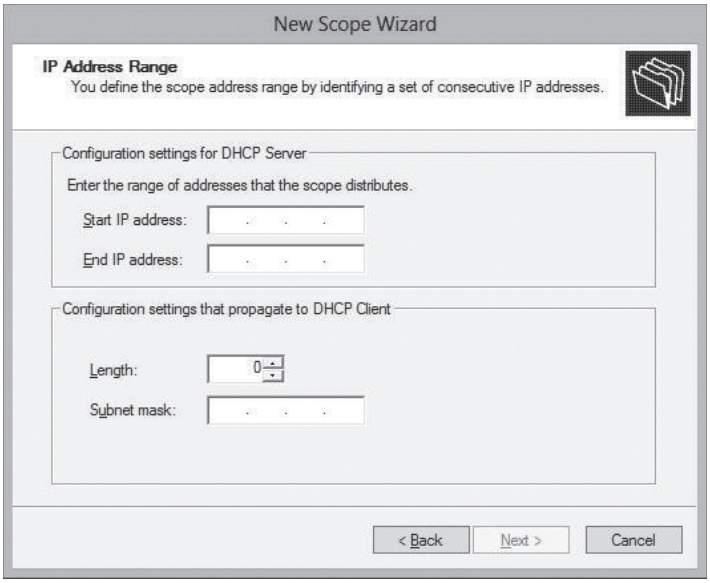
CREATE A DHCP SCOPE

GET READY. Log on to Windows Server 2012 R2, using an account with Administrative privileges.

1. In the *Server Manager* window, click Tools and then click DHCP. The *DHCP* console appears.
2. Expand the server and IPv4 nodes.
3. Right-click the IPv4 node and, from the context menu, select New Scope. The *New Scope Wizard* appears, displaying the *Welcome* page.
4. Click Next. The *Scope Name* page appears.
5. Type a name for the scope into the Name text box and click Next. The *IP Address Range* page appears, as shown in Figure 11-1.

Figure 11-1

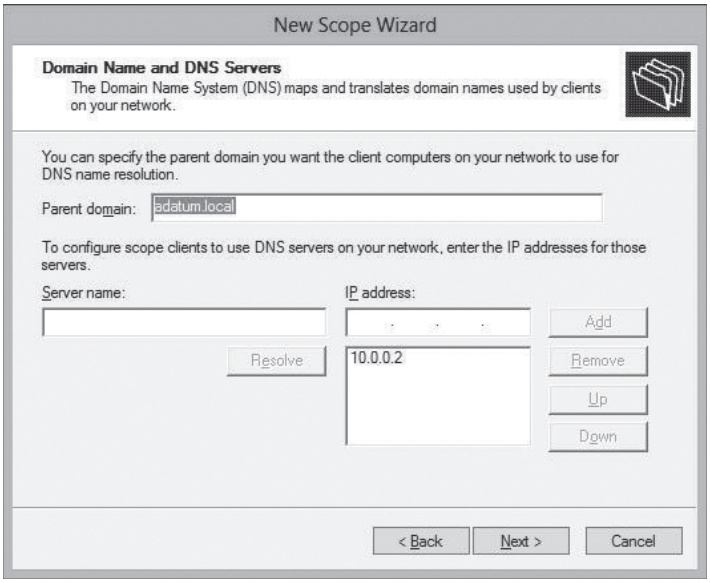
The *IP Address Range* page in the New Scope Wizard



6. In the Start IP address text box, type the first in the range of addresses you want to assign. In the End IP address text box, type the last address in the range.
7. In the Subnet Mask text box, type the mask value for the subnet on which the scope will operate and click Next. The *Add Exclusions and Delay* page appears.
8. In the Start IP address and End IP address text boxes, specify a range of addresses you want to exclude from the scope. You can also specify a delay interval between the server's receipt of DHCPDISCOVER messages and its transmission of DHCPOFFER messages. Then click Next. The *Lease Duration* page appears.
9. Specify the length of the leases for the addresses in the scope and click Next. The *Configure DHCP Options* page appears.
10. Select Yes, I want to configure these options now and click Next. The *Router (Default Gateway)* page appears.
11. In the IP address text box, specify the address of a router on the subnet served by the scope and click Add. Then click Next. The *Domain Name and DNS Servers* page appears, as shown in Figure 11-2.

Figure 11-2

The *Domain Name and DNS Servers* page in the New Scope Wizard



12. In the Server name text box, type the name of a DNS server on the network and click Resolve, or type the address of a DNS server in the IP address text box and click Add. Then click Next. The *WINS Servers* page appears.
13. Click Next. The *Activate Scope* page appears.
14. Select Yes, I want to activate this scope now and click Next. The *Completing the New Scope Wizard* page appears.
15. Click Finish. The wizard closes.

CLOSE the DHCP console.

After the role installation and configuration is complete, all DHCP clients on the subnet identified in the scope you created can obtain their IP addresses and other TCP/IP configuration settings via DHCP. You can also use the DHCP console to create additional scopes for other subnets.

Configuring DHCP Options

The *New Scope Wizard* enables you to configure a few of the most commonly used DHCP options as you create a new scope, but you can always configure the many other options at a later time.

The Windows DHCP server supports two kinds of options:

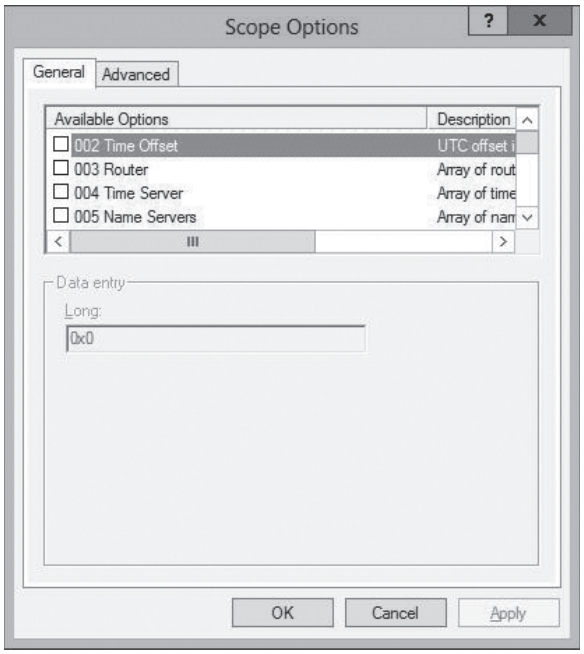
- **Scope options:** Supplied only to DHCP clients receiving addresses from a particular scope.
- **Server options:** Supplied to all DHCP clients receiving addresses from the server.

The *Router* option is a typical example of a scope option, because a DHCP client's default gateway address must be on the same subnet as its IP address. The *DNS Servers* option is typically a server option, because DNS servers do not have to be on the same subnet, and networks often use the same DNS servers for all their clients.

The process of configuring all the options supported by the Windows DHCP server is the same. To configure a scope option, you right-click the *Scope Options* node and, from the context menu, select *Configure Options*. The *Scope Options* dialog box (see Figure 11-3) provides appropriate controls for each available option.

Figure 11-3

The Scope Options dialog box



Right-clicking the *Server Options* node enables you to display the *Server Options* dialog box, which behaves in exactly the same way as the *Scope Options* dialog box.

Creating a Reservation

While DHCP is an excellent TCP/IP configuration solution for most computers on a network, it is not such a good solution for a few components. Domain controllers and DHCP servers themselves, for example, need static IP addresses.

Because the DHCP dynamic allocation method allows for the possibility that a computer's IP address could change, it is not appropriate for these particular roles. However, it is still possible to assign addresses to these computers with DHCP by using manual, instead of dynamic, allocation.

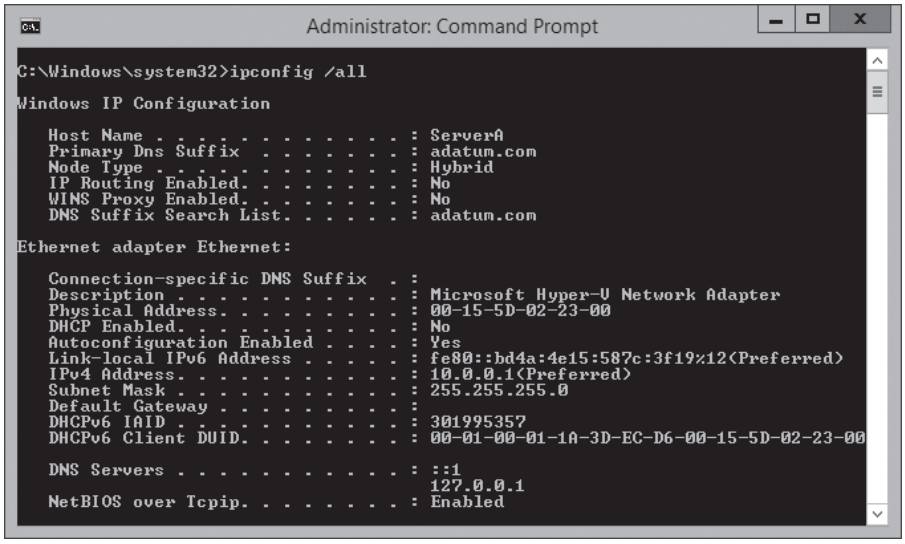
In a Windows DHCP server, a manually allocated address is called a *reservation*. You create a reservation by expanding the scope node, right-clicking the *Reservations* node, and, from the context menu, selecting *New Reservation*. The *New Reservation* dialog box appears.

In this dialog box, you specify the IP address you want to assign and associate it with the client computer's media access control (MAC) address, which is hard-coded into its network interface adapter.

To discover the MAC address of a network interface adapter, run the Ipconfig.exe program with the /all parameter, as shown in Figure 11-4, where the MAC address appears as *Physical Address*.

Figure 11-4

A MAC address in the Ipconfig display



Of course, you also can manually configure the computer's TCP/IP client, but creating a DHCP reservation ensures that your DHCP servers manage all your IP addresses. In a large enterprise, where various administrators might be dealing with DHCP and TCP/IP configuration issues, the IP address that one technician manually assigns to a computer might be included in a DHCP scope by another technician, resulting in potential addressing conflicts. Reservations create a permanent record of the IP address assignment on the DHCP server.

Using PXE

The Windows operating systems include a DHCP client that can configure the IP address and other TCP/IP settings of computers with an operating system already installed. However, it is also possible for a bare metal computer—that is, a computer with no operating system—to use DHCP.

The ***Pre-boot Execution Environment (PXE)*** is a feature built into many network interface adapters that enables them to connect to a DHCP server over the network and obtain TCP/IP client settings, even when the computer has no operating system. Administrators typically use this capability to automate the operating system deployment process on large fleets of computers.

In addition to configuring the IP address and other TCP/IP client settings on the computer, the DHCP server can also supply the workstation with an option specifying the location of a boot file that the system can download and use to start the computer and initiate a Windows operating system installation. A PXE-equipped system downloads boot files using the Trivial File Transfer Protocol (TFTP), a simplified version of the FTP protocol that requires no authentication.

USING PXE WITH WDS

Windows Server 2012 R2 includes a role called Windows Deployment Services (WDS), which enables you to manage image files that remote computers can use to start up and install Windows. For a PXE adapter to access WDS images, the DHCP server on the network must have a custom PXEClient option (option 60) configured with the string "PXEClient." This indicates that the server can also provide option 66, Boot Server Host Name, and option 67, Bootfile Name.

The PXE client on the workstation typically needs no configuration, except possibly to change the boot device order, so that the computer attempts a network boot before using the local devices.

In a properly configured WDS installation of Windows 8, the client operating system deployment process proceeds as follows:

1. The client computer starts and, finding no local boot device, attempts to perform a network boot.
2. The client computer connects to a DHCP server on the network from which it obtains a DHCPOFFER message containing an IP address and other TCP/IP configuration parameters, plus the 60, 66, and 67 PXEClient options identifying the WDS server.
3. The client connects to the WDS server and is supplied with a boot image file, which it downloads using the Trivial File Transfer Protocol (TFTP).
4. The client loads Windows PE and the Windows Deployment Services client from the boot image file onto a RAM disk (a virtual disk created out of system memory) and displays a boot menu containing a list of the install images available from the WDS server.
5. The user on the client computer selects an install image from the boot menu, and the operating system installation process begins.
6. From this point, the setup process proceeds just like a manual installation.

■ Deploying a DHCP Relay Agent

THE BOTTOM LINE

Because they rely on broadcast transmissions, DHCPv4 clients can only access DHCP servers on the local network, under normal circumstances. However, it is possible to create a DHCP infrastructure in which one server provides addresses for multiple subnets. To do this, you must install a DHCP relay agent on every subnet that does not have a DHCP server on it.

Many routers can function as DHCP relay agents, but in situations where they cannot, you can configure a Windows Server 2012 R2 computer to function as a relay agent by using the following procedure.

DEPLOY A DHCP RELAY AGENT

GET READY. Log on to Windows Server 2012 R2, using an account with Administrative privileges.

1. In the *Server Manager* window, click Manage and then click Add Roles and Features. The *Add Roles and Features Wizard* appears, displaying the *Before you begin* page.
2. Click Next. The *Select Installation Type* page appears.
3. Leave the *Role-based or feature-based installation* radio button selected and click Next. The *Select Destination Server* page appears.
4. Select the server on which you want to install the roles and/or features and click Next. The *Select Server Roles* page appears.
5. Select the Remote access check box. An *Add features that are required for Remote Access* dialog box appears.
6. Click Add Features. Then click Next. The *Select features* page appears.
7. Click Next. The *Remote Access* page appears.
8. Click Next. The *Role Services* page appears.
9. Select the Routing check box. The *Add features that are required for Routing* page appears.
10. Click Add Features. Then click Next. The *Web Server Role (IIS)* page appears.
11. Click Next. The *Confirm installation selections* page appears.
12. Click Install. The *Installation progress* page appears as the wizard installs the role.
13. Click the Open the Getting Started Wizard link. The *Configure Remote Access – Getting Started Wizard* appears.
14. Click Deploy VPN only. The *Routing and Remote Access* console appears.
15. Right-click the server node and, on the context menu, select Configure and Enable Routing and Remote Access. The *Routing and Remote Access Server Setup Wizard* launches.
16. Click Next to bypass the *Welcome* page. The *Configuration* page appears, as shown in Figure 11-5.

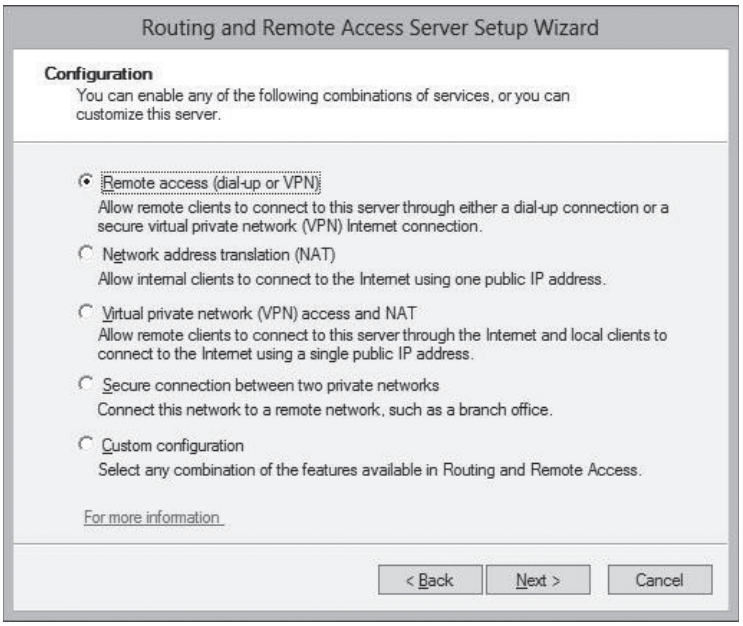


Figure 11-5

The *Configuration* page of the Routing and Remote Access Server Setup Wizard

17. Select Custom configuration and click Next. The *Custom Configuration* page appears.
18. Select the LAN routing check box and click Next. The *Completing the Routing and Remote Access Server Setup Wizard* page appears.
19. Click Finish. A *Routing and Remote Access* message box appears, prompting you to start the service.
20. Click Start Service.
21. Expand the IPv4 node. Then, right-click the General node and, in the context menu, select New Routing Protocol. The *New Routing Protocol* dialog box appears.
22. Select DHCP Relay Agent and click OK. A *DHCP Relay Agent* node appears, subordinate to the IPv4 node.
23. Right-click the DHCP Relay Agent node and, on the context menu, select New Interface. The *New Interface for DHCP Relay Agent* dialog box appears.
24. Select the interface to the subnet on which you want to install the relay agent and click OK. The *DHCP Relay Properties* sheet for the interface appears, as shown in Figure 11-6.

TAKE NOTE*

You can also create a relay agent for DHCPv6 by adding a routing protocol to the IPv6 node

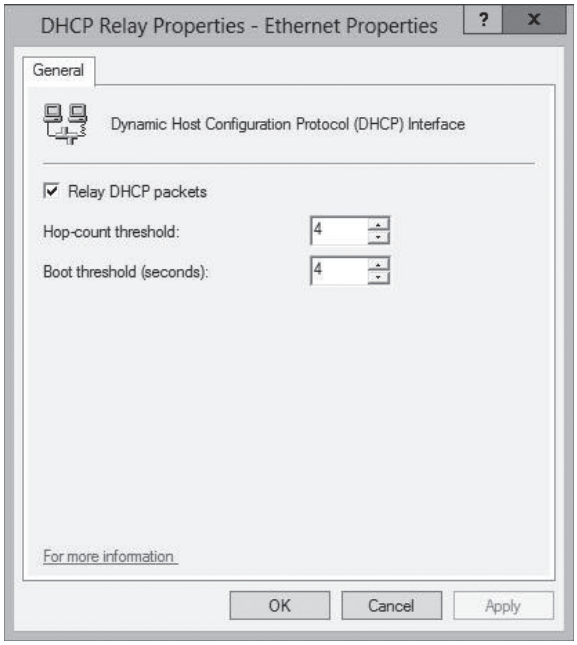


Figure 11-6

The *DHCP Relay Properties* sheet for a selected interface

25. Leave the *Relay DHCP packets* check box selected and configure the following settings, if needed:
    - Hop-count threshold specifies the maximum number of relay agents that DHCP messages can pass through before being discarded. The default value is 4 and the maximum value is 16. This setting prevents DHCP messages from being relayed endlessly around the network.
    - Boot threshold specifies the time interval (in seconds) that the relay agent should wait before forwarding each DHCP message it receives. The default value is 4 seconds. This setting enables you to control which DHCP server processes the clients for a particular subnet.
26. Click OK.
27. Right-click the DHCP Relay Agent node and, on the context menu, select Properties. The *DHCP Relay Agent Properties* sheet appears.

28. Type the IP address of the DHCP server to which you want the agent to relay messages and click Add. Repeat this step to add servers, if necessary.
29. Click OK.

CLOSE the *Routing and Remote Access* console.

At this point, the server is configured to relay DHCP messages to the server addresses you specified.

■ Business Case Scenarios

Scenario 11-1: Configuring DHCP Servers

After deploying a large number of wireless laptop computers on the network, Taylor, the IT director at Contoso, Ltd. decides to use DHCP to enable the laptop users to move from one subnet to another without having to manually reconfigure their IP addresses. Soon after the DHCP deployment, however, Taylor notices that some of the IP address scopes are being depleted, resulting in some computers being unable to connect to a new subnet. What can Taylor do to resolve this problem without altering the network's subnetting?

Scenario 11-2: Maximizing Lease Availability

You are configuring DHCP scope options for Contoso, Ltd. The company has a limited number of IP addresses available for clients, and it wants to configure DHCP to maximize IP address availability. Choose all of the following actions that will accomplish this objective:

a. Set long lease durations for IP addresses.
b. Set short lease durations for IP addresses.
c. Configure a DHCP option to automatically release an IP address when the computer shuts down.
d. Create DHCP reservations for all portable computers.

12 LESSON

Deploying and Configuring the DNS Service

Understanding the DNS Architecture

THE BOTTOM LINE

The Domain Name System (DNS) is a crucial element of both Internet and Active Directory communications.

All TCP/IP communication is based on IP addresses. Each computer on a network has at least one network interface, which is called a ***host***, in TCP/IP parlance, and each host has an IP address that is unique on that network. Every datagram transmitted by a TCP/IP system contains the sending computer's IP address and the intended recipient's IP address. However, when users access a shared folder on the network or a website on the Internet, they do so by specifying or selecting a host name, not an IP address. This is because names are easier to remember and use than IP addresses.

For TCP/IP systems to use these friendly host names, they must have some way to discover the IP address associated with a specific name. In the early days of TCP/IP networking, each computer had a list of names and their equivalent IP addresses, called a ***host table***. At that time, there were few enough computers on the fledgling Internet for the maintenance and distribution of a single host table to be practical.

Today, millions of computers are on the Internet, and the idea of maintaining and distributing a single file containing names for all of them is absurd. Rather than a host table stored on every computer, TCP/IP networks today use DNS servers to convert host names into IP addresses. This conversion process is referred to as ***name resolution***. The resulting solution was the Domain Name System (DNS).

Creating a DNS Standard

When the developers of what became the Internet recognized the increasing impracticality of the host table, they set about devising a solution that would not only solve their immediate maintenance and distribution problems, but would also remain a viable solution for decades to come.

At its core, the DNS is still a list of names and their equivalent IP addresses, but the method for creating, storing, and retrieving the names is different from those in a host table. The DNS consists of three elements, as follows:

- **The DNS name space:** The DNS standards define a tree-structured name space in which each branch of the tree identifies a ***domain***. Each domain contains a collection of ***resource records*** that contain host names, IP addresses, and other information. Query operations are attempts to retrieve specific resource records from a particular domain.
- **Name servers:** A DNS server is an application running on a server computer that maintains information about the domain tree structure and (usually) contains authoritative

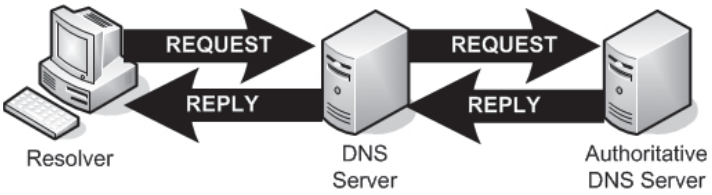
information about one or more specific domains in that structure. The application is capable of responding to queries for information about the domains for which it is the authority, and also of forwarding queries about other domains to other name servers. This enables any DNS server to access information about any domain in the tree.

- **Resolvers:** A ***resolver*** is a client program that generates DNS queries and sends them to a DNS server for fulfillment. A resolver has direct access to at least one DNS server and can also process referrals to direct its queries to other servers when necessary.

In its most basic form, the DNS name resolution process consists of a resolver submitting a name resolution request to its designated DNS server. When the server does not possess information about the requested name, it forwards the request to another DNS server on the network. The second server generates a response containing the IP address of the requested name and returns it to the first server, which relays the information in turn to the resolver, as shown in Figure 12-1. In practice, however, the DNS name resolution process can be considerably more complex, as you learn in the following sections.

Figure 12-1

DNS servers relay requests and replies to other DNS servers



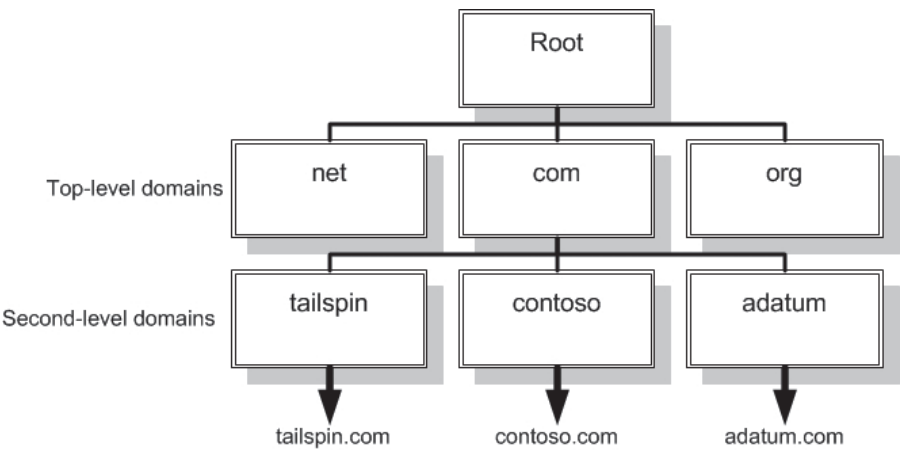
DNS Naming

To facilitate the continued growth of the namespace, the developers of the DNS created a two-tiered system, consisting of domain names and host names. The basic principle is that the administrators of individual networks obtain domain names from a centralized authority, and then assign the host names within that domain themselves. This process enables administrators to assign host names without worrying about duplication, as long as each host name is unique within its domain.

The domain name part of a DNS name is hierarchical and consists of two or more words, separated by periods. The DNS namespace takes the form of a tree that, much like a file system, has its root at the top. Just beneath the root is a series of top-level domains, and beneath each top-level domain is a series of second-level domains, as shown in Figure 12-2. At minimum, the complete DNS name for a computer on the Internet consists of a host name, a second-level domain name, and a top-level domain name, written in this order and separated by periods. The complete DNS name for a particular computer is called its ***fully qualified domain name (FQDN)***.

Figure 12-2

The DNS domain hierarchy



Unlike an IP address, which places the network identifier first and follows it with the host, the notation for an FQDN places the host name first, followed by the domain name, with the top-level domain name last. In the example cited previously, the FQDN *www.contoso.com* consists of a host (or computer) called *www* in the *contoso.com* domain. In the contoso.com domain name, *com* is the top-level domain and *contoso* is the second-level domain. Technically, every FQDN should end with a period, representing the root of the DNS tree, as follows:

```
www.contoso.com.
```

However, the period is rarely included in FQDNs today.

Understanding the DNS Domain Hierarchy

The hierarchical nature of the DNS namespace is designed to make it possible for any DNS server on the Internet to locate the authoritative source for any domain name, by using a minimum number of queries. This efficiency results from the domains at each level of the hierarchy are responsible for maintaining information about the domains at the next lower level.

The authoritative source for any domain is the DNS server (or servers) responsible for maintaining that domain's resource records. Each level of the DNS domain hierarchy has name servers responsible for the individual domains at that level.

At the top of the DNS hierarchy are the root name servers. The ***root name servers*** are the highest-level DNS servers in the entire namespace, and they maintain information about the top-level domains. All DNS server implementations are preconfigured with the IP addresses of the root name servers, because these servers are the ultimate source for all DNS information. When a computer attempts to resolve a DNS name, it begins at the top of the namespace hierarchy with the root name servers and works down the levels until it reaches the authoritative server for the domain in which the name is located.

TOP-LEVEL DOMAINS

Just beneath the root name servers are the top-level domains (TLDs). The original DNS name space called for six ***generic top-level domains (gTLDs),*** dedicated to specific purposes, as follows:

- *com:* Commercial organizations
- *edu:* Four-year, degree-granting educational institutions in North America
- *gov:* United States government institutions
- *mil:* United States military applications
- *net:* Networking organizations
- *org:* Noncommercial organizations

The *edu*, *gov*, and *mil* domains are reserved for use by certified organizations, but the *com*, *org*, and *net* domains are called ***global domains,*** because organizations anywhere in the world can register second-level domains within them.

SECOND-LEVEL DOMAINS

Each top-level domain has its own collection of second-level domains. Individuals and organizations can purchase these domains for their own use. For the payment of an annual fee, you can purchase the rights to a second-level domain.

To use the domain name, you must supply the registrar with the IP addresses of two DNS servers that you want to be the authoritative sources for information about the domain. A DNS server is a software program that runs on a computer. DNS server products are available

for all the major network operating systems. The DNS servers do not need to be located on the registrant's network; many companies outsource their Internet server hosting chores and use their service provider's DNS servers.

SUBDOMAINS

After you purchase the rights to a second-level domain, you can create as many hosts as you want in that domain, by creating new resource records on the authoritative servers. You can also create as many additional domain levels as you want. The only limitations to the subdomains and hosts you can create in your second-level domain are as follows:

- Each individual domain name can be no more than 63 characters long.
- The total FQDN (including the trailing period) can be no more than 255 characters long.

For the convenience of users and administrators, most domain names do not approach these limitations.

Understanding DNS Communications

Although all Internet applications use DNS to resolve host names into IP addresses, this name resolution process is easiest to see when you use a web browser to access an Internet site.

When you type a URL containing a DNS name (such as *www.microsoft.com*) into the browser's *Address* box and press the *Enter* key, you might see a message that says something like, "Finding Site: www.microsoft.com." Then, a few seconds later, you might see a message that says, "Connecting to," followed by an IP address. It is during this interval that the DNS name resolution process occurs.

From the client's perspective, the procedure that occurs during these few seconds consists of the application sending a query message to its designated DNS server that contains the name to be resolved. The server replies with a message containing the IP address corresponding to that name. Using the supplied address, the application can transmit a message to the intended destination. When you examine the DNS server's role in the process, you see the procedure's complexity.

Comprehending DNS Server Caching

The DNS name resolution process might seem long and complex, but in many cases, it isn't necessary for the client's DNS server to send queries to the servers for each domain specified in the requested DNS name. That is, DNS servers are capable of retaining the information they learn about the DNS name space in the course of their name resolution procedures and storing it in a cache on the local drive.

A DNS server that receives requests from clients, for example, caches the addresses of the requested systems, as well as the addresses for authoritative servers of particular domains. The next time that a client requests the resolution of a previously resolved name, the server can respond immediately with the cached information, as shown in Figure 12-3. In addition, if a client requests another name in one of the same domains, the server can send a query directly to an authoritative server for that domain, and not to a root name server. Thus, the names in commonly accessed domains generally resolve more quickly because one of the servers along the line has information about the domain in its cache, whereas names in obscure domains take longer because the entire request/referral process is needed.

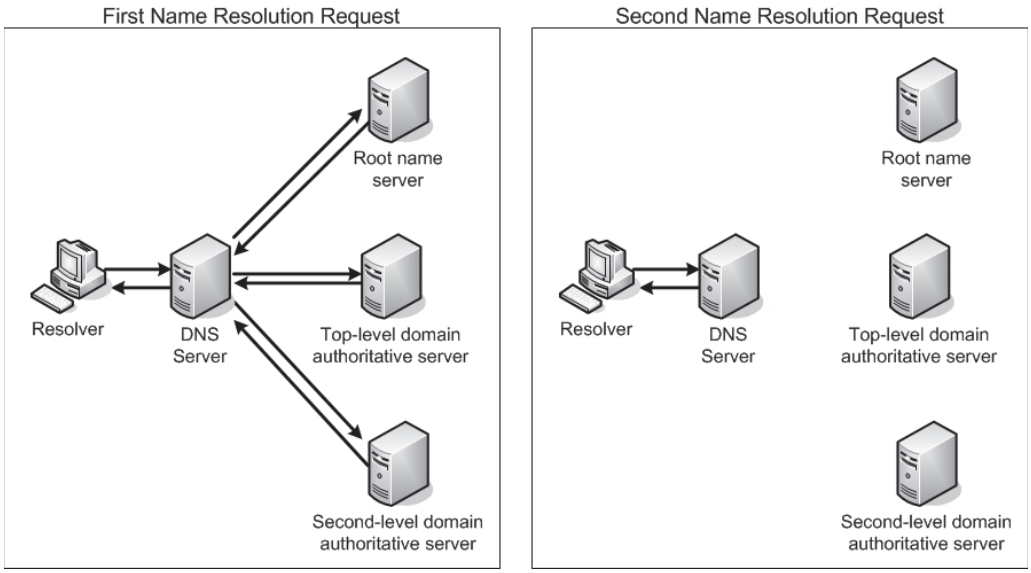


Figure 12-3

Name caching enables the second name resolution request for the same name to bypass the referral process

NEGATIVE CACHING

In addition to storing information that aids in the name resolution process, most modern DNS server implementations are also capable of negative caching. ***Negative caching*** occurs when a DNS server retains information about names that do not exist in a domain. If, for example, a client sends a query to its DNS server containing a name in which the second-level domain does not exist, the top-level domain server will return a reply containing an error message to that effect. The client's DNS server will then retain the error message information in its cache. The next time a client requests a name in that domain, the DNS server will be able to respond immediately with its own error message, without consulting the top-level domain.

CACHE DATA PERSISTENCE

Caching is a vital element of the DNS architecture, because it reduces the number of requests sent to the root name and top-level domain servers, which, being at the top of the DNS tree, are the most likely to act as a bottleneck for the whole system. However, caches must be purged eventually, and there is a fine line between effective and ineffective caching.

Because DNS servers retain resource records in their caches, it can take hours or even days for changes made in an authoritative server to be propagated around the Internet. During this period, users might receive incorrect information in response to a query. If information remains in server caches too long, then the changes that administrators make to the data in their DNS servers take too long to propagate around the Internet. If caches are purged too quickly, then the number of requests sent to the root name and top-level domain servers increases precipitously.

The amount of time that DNS data remains cached on a server is called its *Time To Live (TTL).* Unlike most data caches, the TTL is not specified by the administrator of the server where the cache is stored. Instead, the administrators of each authoritative DNS server specify how long the data for the resource records in their domains or zones should be retained in the servers where it is cached. This enables administrators to specify a TTL value based on the volatility of their server data. On a network where changes in IP addresses or the addition of new resource records is frequent, a lower TTL value increases the likelihood that clients will receive current data. On a network that rarely changes, you can use a longer TTL value, and minimize the number of requests sent to the parent servers of your domain or zone.

To modify the TTL value for a zone on a Windows Server 2012 DNS server, right-click the zone, open the Properties sheet, and click the *Start Of Authority (SOA)* tab. On this tab, you can modify the TTL for this record setting from its default value of one hour.

Using DNS Forwarders

DNS servers send recursive queries to other servers when you configure a server to function as a forwarder.

On a network running several DNS servers, you might not want all the servers sending queries to other DNS servers on the Internet. If the network has a slow connection to the Internet, for example, several servers transmitting repeated queries might use too much of the available bandwidth.

To prevent this, most DNS implementations enable you to configure one server to function as the forwarder for all Internet queries generated by the other servers on the network. Any time that a server has to resolve the DNS name of an Internet system and fails to find the needed information in its cache, it transmits a recursive query to the forwarder, which is then responsible for sending its own iterative queries over the Internet connection. After the forwarder resolves the name, it sends a reply back to the original DNS server, which relays it to the client.

To configure forwarders on a Windows Server 2012 R2 DNS server, use the following procedure.

CONFIGURE FORWARDERS

GET READY. Log on to Windows Server 2012 R2 domain controller by using an account with administrative privileges.

1. In the *Server Manager* window, click Tools > DNS. The *DNS Manager* console appears.
2. Right-click the server node and, from the context menu, select Properties. The server's Properties sheet appears.
3. Click the Forwarders tab.
4. Click Edit. The Edit Forwarders dialog box appears.
5. Type the name or address of the DNS server you want to function as a forwarder and press Enter. The system validates the name or address by connecting to the DNS server.
6. Click OK to close the Edit Forwarders dialog box and add the servers to the Forwarders tab.
7. Click OK to close the server's Properties sheet.

CLOSE the *DNS Manager* console.

Designing a DNS Deployment

THE BOTTOM LINE

All the DNS information in the preceding sections might at first seem gratuitous, but understanding the structure of the DNS and how the clients and servers communicate is crucial to creating an effective DNS deployment plan.

Every computer on a TCP/IP network needs access to a DNS server, but this does not mean that you must deploy your own DNS servers on your network. Internet service providers (ISPs) nearly always include the use of their DNS servers into their rates, and in some cases, it might be better to use other DNS servers, rather than run your own.

When designing a DNS deployment, you should first determine what DNS services your network requires. Consider the information in the following sections as you create your design.

Resolving Internet Names

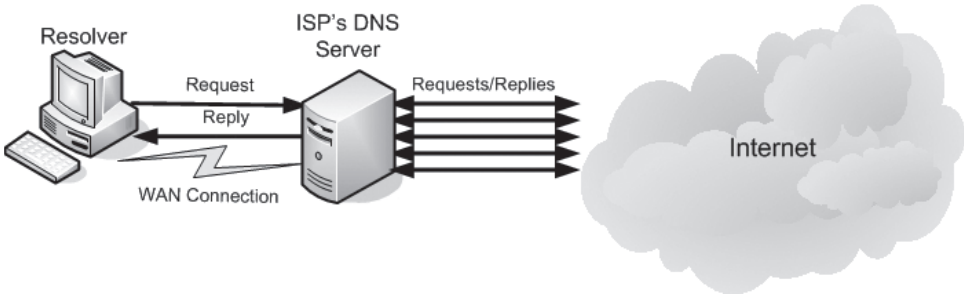
If you provide your network users with access to the Internet, as most organizations do, then every user must have at least one DNS server address specified in its TCP/IP configuration settings. If you use DHCP servers to assign IP addresses to the network computers, you can configure the servers to configure the clients' DNS server addresses as well.

For Internet name resolution purposes, the only functions required of the DNS server are the capability to process incoming queries from resolvers and send its own queries to other DNS servers on the Internet. A DNS server that performs only these functions is known as a ***caching-only server***, because it is not the authoritative source for any domain and hosts no resource records of its own.

Installing your own caching-only DNS server is a simple matter, or you can use the DNS servers supplied by your ISP. The important factor to consider in this decision is the amount of traffic generated by the server's query process. In the DNS name resolution process, the client resolver and its DNS server exchange one query message and one reply. If the clients on your local network use the DNS servers on your ISP's network, then your Internet connection has to handle only these two messages, as shown in Figure 12-4.

Figure 12-4

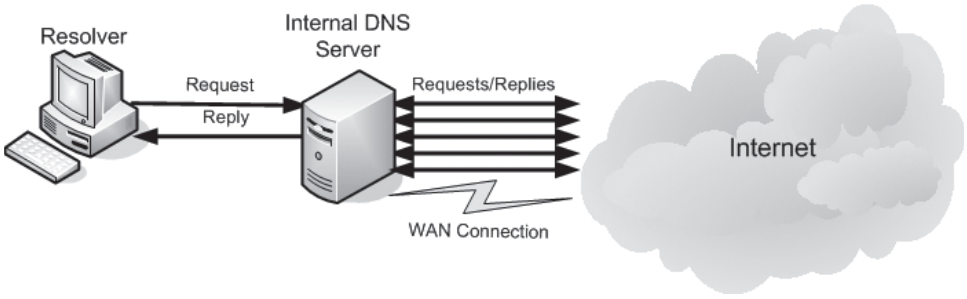
Using an ISP's caching-only DNS server



If, however, you install a DNS server on your local network, the recursive queries the server receives from clients cause it to send numerous iterative queries to various other DNS servers on the Internet. These multiple message exchanges must all pass over the Internet connection, as shown in Figure 12-5. When you have hundreds or thousands of clients using the DNS server, the amount of iterative query traffic the server generates can overburden your Internet connection or greatly increase its cost.

Figure 12-5

Using your own caching-only DNS server



As a general rule, if your network requires no DNS services other than name resolution, you should consider using off-site DNS servers.

Hosting Internet Domains

If you host a domain on the Internet, you must pay an annual fee to register a second-level domain name with a commercial registrar and supply it with the IP addresses of your DNS servers. These servers are the authoritative source for information about your domain. Therefore, they must have registered IP addresses and be accessible from the Internet at all times.

The two main reasons for registering an Internet domain name are to host web servers and to create e-mail addresses. The authoritative DNS servers for the domain must have resource records that can provide Internet users with the IP addresses of your web servers and e-mail servers. If the authoritative DNS servers are offline, Internet users might be unable to access your web servers, and e-mail messages destined for your domain could bounce back to their senders.

As with name resolution, the DNS servers you use to host your domain can be computers on your own network or servers supplied by a commercial entity. The DNS servers that host your domain do not need to be located in that domain, nor do they need to be supplied by the registrar from whom you obtained your domain name. You can usually pay an additional fee to your domain registrar and have them host the domain on their servers, or you can use you ISP's servers, also for an additional fee.

■ Creating Internet Domains

THE BOTTOM LINE

Designing a DNS namespace for your organization's Internet presence is usually the easiest part of deploying DNS. Most organizations register a single second-level domain and use it to host all their Internet servers.

In most cases, the selection of a second-level domain name depends on what is available. A large portion of the most popular top-level domain, *com*, is already depleted, and you might find that the name you want to use is already taken. In this case, you have three alternatives:

- Choose a different domain name.
- Register the name in a different top-level domain.
- Attempt to purchase the domain name from its current owner.

Some organizations maintain multiple sites on the Internet, for various reasons. Your organization might be involved in several separate businesses that warrant individual treatment, or your company might have independent divisions with different sites. You might also create different sites for retail customers, wholesale customers, and providers. Whatever the reason, the following two basic ways can help you implement multiple sites on the Internet:

- **Register a single second-level domain name and then create multiple subdomains beneath it:** For the price of a single domain registration, you can create as many third-level domains as you need, and you can also maintain a single brand across all your sites. For example, a company called *Contoso Pharmaceuticals* might register the contoso.com domain, and then create separate websites for doctors and patients, in domains called *doctors.contoso.com* and *patients.contoso.com*.
- **Register multiple second-level domains:** If your organization consists of multiple, completely unrelated brands or operations, this is often the best solution. You must pay a separate registration fee for each domain name, however, and you must maintain a separate DNS namespace for each domain. A problem might also arise when you try to integrate your Internet domains with your internal network. You can select one of your second-level domains to integrate with your internal namespace, or you can leave your internal and external namespaces completely separate, as discussed later in this lesson.

■ Deploying a DNS Server

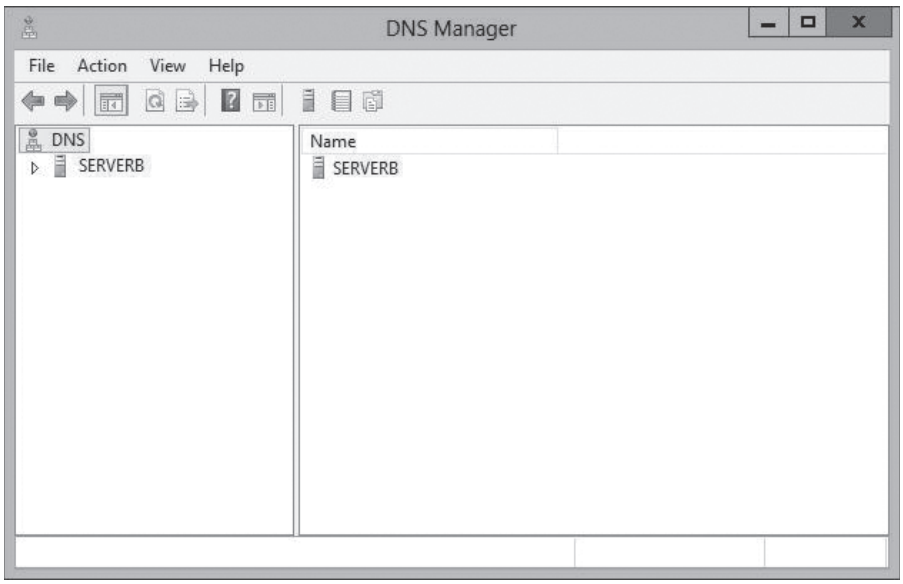
THE BOTTOM LINE

The process of deploying a DNS server on a Windows Server 2012 R2 computer is a matter of installing the DNS Server role, by using the Add Roles and Features Wizard in Server Manager. The actual installation requires no additional input; there are no additional pages in the wizard and no role services to select.

After you install the DNS Server role, the computer is ready to perform caching-only name resolution services for any clients that have access to it. The role also installs the DNS Manager console, shown in Figure 12-6, which you use to configure the DNS server's other capabilities. To configure the server to perform other services, consult the following sections.

Figure 12-6

The DNS Manager console



Creating Zones

A ***zone*** is an administrative entity you create on a DNS server to represent a discrete portion of the DNS namespace.

Administrators typically divide the DNS namespace into zones to store them on different servers and to delegate their administration to different people. Zones always consist of entire domains or subdomains. You can create a zone that contains multiple domains, as long as those domains are contiguous in the DNS namespace. For example, you can create a zone containing a parent domain and its child, because they are directly connected, but you cannot create a zone containing two child domains without their common parent, because the two children are not directly connected.

You can divide the DNS namespace into multiple zones and host them on a single DNS server, although there is usually no persuasive reason to do so. The DNS server in Windows Server 2012 R2 can support as many as 200,000 zones on a single server, although it is hard to imagine a scenario that would require that many. In most cases, an administrator creates multiple zones on a server and then delegates most of them to other servers, which then become responsible for hosting them.

Every zone consists of a zone database, which contains the resource records for the domains in that zone. The DNS server in Windows Server 2012 R2 supports three zone types, which specify where the server stores the zone database and what kind of information it contains. These zone types are as follows:

- **Primary zone:** creates a primary zone that contains the master copy of the zone database, where administrators make all changes to the zone's resource records. If the *Store The Zone In Active Directory (Available Only If DNS Server Is A Domain Controller)* check box is cleared, the server creates a primary master zone database file on the local drive. This is a simple text file that is compliant with most non-Windows DNS server implementations.

- **Secondary zone:** creates a duplicate of a primary zone on another server. The secondary zone contains a backup copy of the primary master zone database file, stored as an identical text file on the server's local drive. You cannot modify the resource records in a secondary zone manually; you can update them only by replicating the primary master zone database file, using a process called a ***zone transfer***. You should always create at least one secondary zone for each file-based primary zone in your namespace, both to provide fault tolerance and to balance the DNS traffic load.
- **Stub zone:** creates a copy of a primary zone that contains the key resource records that identify the authoritative servers for the zone. The stub zone forwards or refers requests. When you create a stub zone, you configure it with the IP address of the server that hosts the zone from which you created the stub. When the server hosting the stub zone receives a query for a name in that zone, it either forwards the request to the host of the zone or replies with a referral to that host, depending on whether the query is recursive or iterative.

The DNS was designed long before Active Directory, so most of the Internet relies on primary and secondary zones using text-based database files. The most common DNS server implementation on the Internet is a UNIX program called *bind* that uses these databases.

However, for DNS servers supporting internal domains, and especially AD DS domains, using the Windows DNS server to create a primary zone and store it in Active Directory is the recommended procedure. When you store the zone in the AD DS database, you do not need to create secondary zones or perform zone transfers, because AD DS takes the responsibility for replicating the data, and your backup solution used to protect Active Directory protects the DNS data.

USING ACTIVE DIRECTORY-INTEGRATED ZONES

When you are running the DNS server service on a computer that is an AD DS domain controller and you select the *Store The Zone In Active Directory (Available Only If DNS Server Is A Domain Controller)* check box while creating a zone in the *New Zone Wizard*, the server does not create a zone database file. Instead, the server stores the DNS resource records for the zone in the AD DS database. Storing the DNS database in Active Directory provides several advantages, including ease of administration, conservation of network bandwidth, and increased security.

In Active Directory-integrated zones, the zone database is replicated automatically to other domain controllers, along with all other Active Directory data. Active Directory uses a multiple master replication system so that copies of the database are updated on all domain controllers in the domain. You can modify the DNS resource records on any domain controller hosting a copy of the zone database, and Active Directory will update all of the other domain controllers automatically. You don't need to create secondary zones or manually configure zone transfers, because Active Directory performs all database replication activities.

By default, Windows Server 2012 R2 replicates the database for a primary zone stored in Active Directory to all the other domain controllers running the DNS server in the AD DS domain where the primary is located. You can also modify the scope of zone database replication to keep copies on all domain controllers throughout the enterprise, or on all domain controllers in the AD DS domain, whether or not they are running the DNS server. You can also create a custom replication scope that copies the zone database to the domain controllers you specify.

Active Directory conserves network bandwidth by replicating only the DNS data that has changed since the last replication, and by compressing the data before transmitting it over the network. The zone replications also use the full security capabilities of Active Directory, which are considerably more robust than those of file-based zone transfers.

CREATING AN ACTIVE DIRECTORY ZONE

To create a new primary zone and store it in Active Directory, use the following procedure.

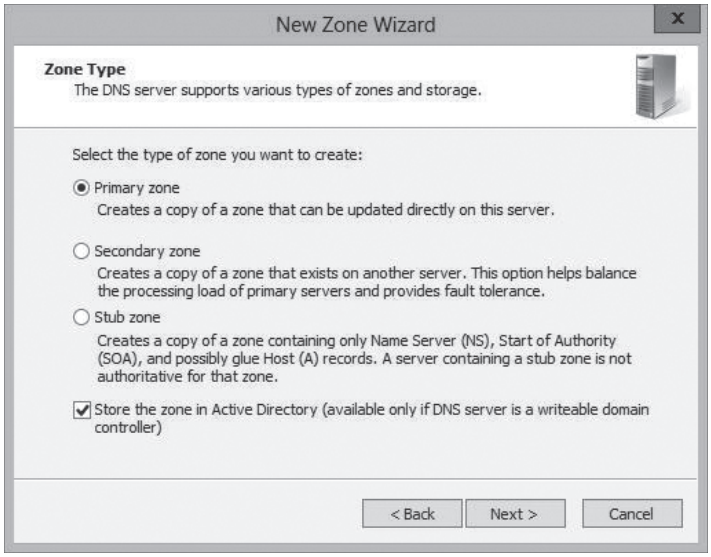
CREATE AN ACTIVE DIRECTORY ZONE

GET READY. Log on to Windows Server 2012 R2 domain controller using an account with administrative privileges.

1. In the *Server Manager* window, click Tools > DNS. The *DNS Manager* console appears.
2. Expand the server node and select the Forward Lookup Zones folder.
3. Right-click the Forward Lookup Zones folder and, from the context menu, select New Zone. The *New Zone Wizard* appears.
4. Click Next to bypass the *Welcome* page. The *Zone Type* page appears, as shown in Figure 12-7.

Figure 12-7

The *Zone Type* page of the New Zone Wizard



5. Leave the *Primary Zone* option and the *Store The Zone In Active Directory (Available Only If DNS Server Is A Domain Controller)* check box selected and click Next. The *Active Directory Zone Replication Scope* page appears.
6. Click Next. The *Zone Name* page appears.
7. Specify the name you want to assign to the zone in the Zone Name text box and click Next. The *Dynamic Update* page appears.
8. Select one of the following options:
   - Allow only secure dynamic updates
   - Allow both nonsecure and secure dynamic updates
   - Do not allow dynamic updates
9. Click Next. The *Completing the New Zone Wizard* page appears.
10. Click Finish. The wizard creates the zone.

CLOSE the *DNS Manager* console.

After you create a primary zone, you can create resource records that specify the names of the hosts on the network and their equivalent IP addresses.

Creating Resource Records

When you run your own DNS server, you create a resource record for each host name that you want to be accessible by the rest of the network.

There are several different types of resource records used by DNS servers, the most important of which are as follows:

- **SOA (Start of Authority):** indicates that the server is the best authoritative source for data concerning the zone. Each zone must have an SOA record, and only one SOA record can be in a zone.
- **NS (Name Server):** identifies a DNS server functioning as an authority for the zone. Each DNS server in the zone (whether primary master or secondary) must be represented by an NS record.
- **A (Address):** provides a name-to-address mapping that supplies an IPv4 address for a specific DNS name. This record type performs the primary function of the DNS, converting names to addresses.
- **AAAA (Address):** provides a name-to-address mapping that supplies an IPv6 address for a specific DNS name. This record type performs the primary function of the DNS, converting names to addresses.
- **PTR (Pointer):** provides an address-to-name mapping that supplies a DNS name for a specific address in the *in-addr.arpa* domain. This is the functional opposite of an A record, used for reverse lookups only.
- **CNAME (Canonical Name):** creates an alias that points to the *canonical* name (that is, the "real" name) of a host identified by an A record. Administrators use CNAME records to provide alternative names by which systems can be identified.
- **MX (Mail Exchanger):** identifies a system that directs e-mail traffic sent to an address in the domain to the individual recipient, a mail gateway, or another mail server.

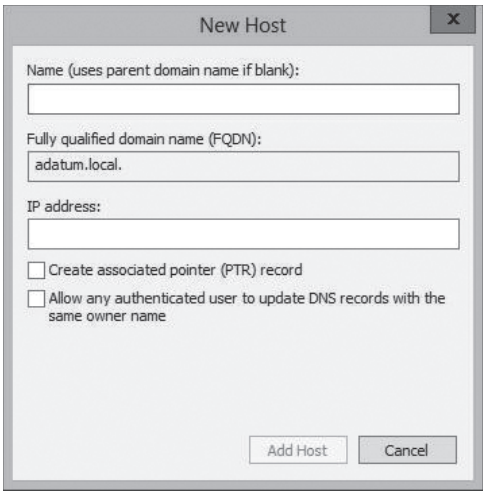
To create a new Address resource record, use the following procedure.

CREATE AN ADDRESS RESOURCE RECORD

GET READY. Log on to Windows Server 2012 R2 using an account with Administrative privileges.

1. In the *Server Manager* window, click Tools > DNS. The *DNS Manager* console appears.
2. Expand the server node and select the Forward Lookup Zones folder.
3. Right-click the zone in which you want to create the record and, from the context menu, select New Host (A or AAAA). The *New Host* dialog box appears, as shown in Figure 12-8.

Figure 12-8
The New Host dialog box



4. In the Name text box, type the host name for the new record. The FQDN for the record appears.

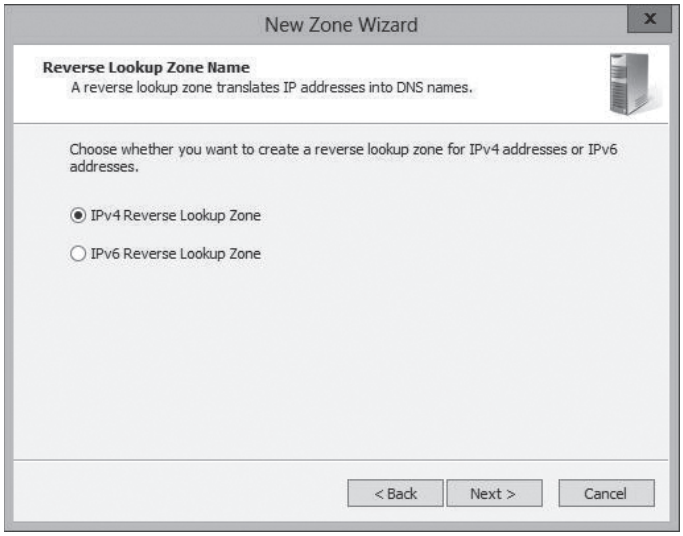
5. In the IP address text box, type the IPv4 or IPv6 address associated with the host name.
6. Select the following check boxes, if necessary:
   - Create associated pointer (PTR) record creates a reverse name lookup record for the host in the in-addr.arpa domain
   - Allow any authenticated use to update DNS records with the same owner name enables users to modify their own resource records
7. Click Add Host. The new resource record is created in the zone you selected.

CLOSE the *DNS Manager* console.

To create a PTR record for a new host, you can select the *Create associated pointer (PTR) record* check box in the *New Host* dialog box, but that will only be effective if a reverse lookup zone already exists on the server. To create the zone, you follow the same procedure described previously, but select the *Reverse Lookup Zones* folder. This causes the *New Zone Wizard* to add a *Reverse Lookup Zone Name* page like the one shown in Figure 12-9.

Figure 12-9

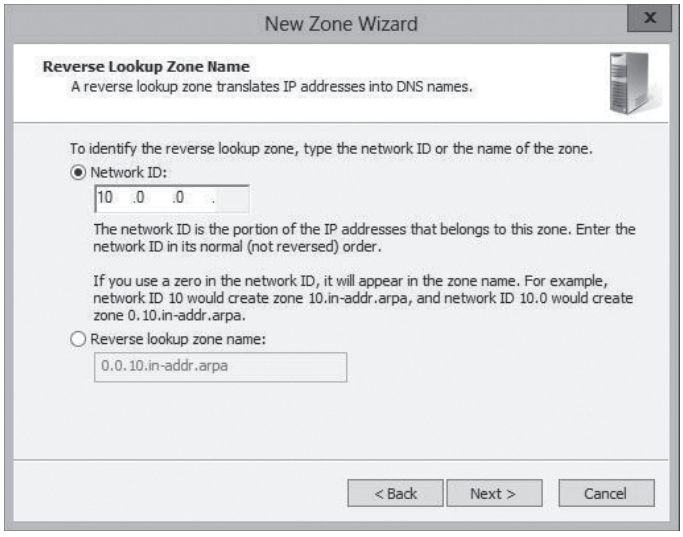
The *Reverse Lookup Zone Name* page in the New Zone Wizard



A second page then appears, as shown in Figure 12-10, in which you supply the Network ID that the wizard uses to create the zone.

Figure 12-10

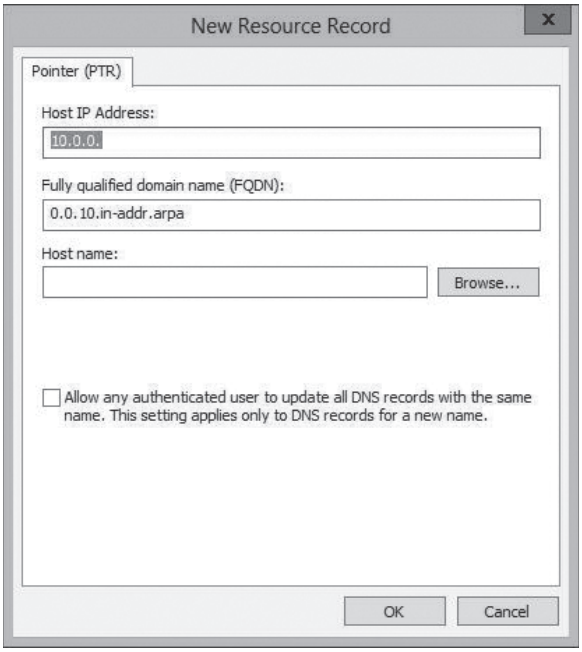
The second *Reverse Lookup Zone Name* page in the New Zone Wizard



After the zone is created, you can either create PTR records along with A or AAAA records, or create a new PTR record by using the interface shown in Figure 12-11.

Figure 12-11

The New Resource Record dialog box



Configuring DNS Server Settings

After you install a DNS server and create zones and resource records on it, you can alter settings to modify its behavior.

The following sections describe some of these settings.

CONFIGURING ACTIVE DIRECTORY DNS REPLICATION

To modify the replication scope for an Active Directory-integrated zone, open the zone's Properties sheet in the *DNS Manager* console, and in the *General* tab, click the *Change button for Replication: All DNS Servers In the Active Directory Domain* to display the *Change Zone Replication Scope* dialog box. The options are the same as those in the New Zone Wizard.

CONFIGURING ROOT HINTS

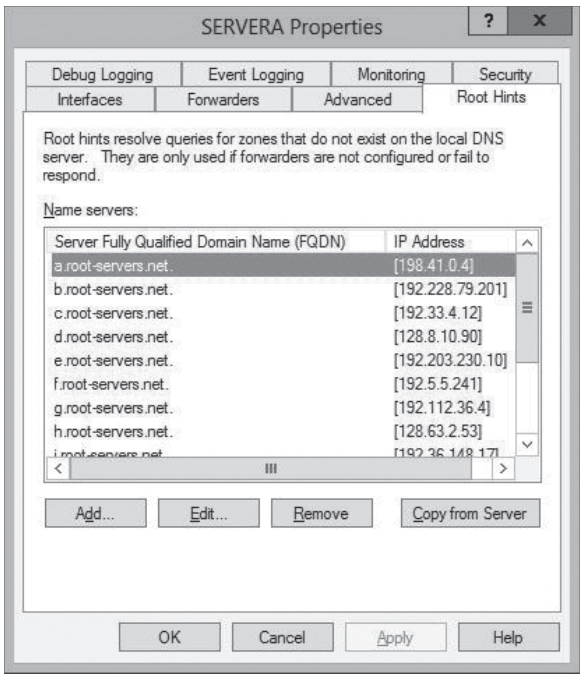
Every DNS server must be able to contact the root name servers to initiate name resolution processes. Most server implementations, including Microsoft DNS Server, are preconfigured with the names and addresses of multiple root name servers. These are called *root hints*.

The 13 root name server names are located in a domain called *root-servers.net*, and are named using letters of the alphabet. The servers are scattered around the world on different subnets, to provide fault tolerance.

To modify the root hints on a Windows Server 2012 DNS server, right-click the server node, open the Properties sheet, and click the *Root Hints* tab, as shown in Figure 12-12. On this tab, you can add, edit, or remove root hints from the list provided.

Figure 12-12

The Root Hints tab on a DNS server's Properties sheet



■ Business Case Scenarios

Scenario 12-1: Deploying DNS Servers

Harold is a freelance networking consultant who has designed a network for a small company with a single location. The owner of the company wants to use an Active Directory domain, so Harold installs a Windows Server 2012 R2 domain controller with the Active Directory Domain Services and DNS Server roles. Harold also uses DHCP to configure all of the workstations on the network to use the DNS services provided by the domain controller.

Soon after the installation, however, the owner of the company reports extremely slow Internet performance. After examining the traffic passing over the Internet connection, you determine that it is being flooded with DNS traffic. What can you do to reduce the amount of DNS traffic passing over the internet connection?

Scenario 12-2: Regulating DNS Traffic

Ralph is an enterprise administrator for Wingtip Toys, which has recently expanded its customer service division by adding 100 workstations. All of the workstations on the company network are configured to use a server on the perimeter network as their primary DNS server and a server on their ISP's network as a secondary. As a result of the expansion, Internet performance has slowed down perceptibly, and a Network Monitor trace indicates that there is a disproportionate amount of DNS traffic on the link between the perimeter network and the ISP's network. What are two ways that Ralph can reduce the amount of DNS traffic passing over the Internet connection?

Installing Domain Controllers

LESSON 13

■ Introducing Active Directory

THE BOTTOM LINE

A directory service is a repository of information about the resources—hardware, software, and human—connected to a network. Users, computers, and applications throughout the network can access the repository for various purposes, including user authentication, storage of configuration data, and even simple white pages–style information lookups. ***Active Directory Domain Services (AD DS)*** is the directory service that Microsoft first introduced in Windows 2000 Server and has upgraded in each successive server operating system release, including Windows Server 2012 R2.

AD DS is a directory service that enables you to create organizational divisions called domains. A ***domain*** is a logical container of network components, hosted by at least one server designated as a ***domain controller***. The domain controllers for each domain replicate their data among themselves for fault tolerance and load-balancing purposes.

Understanding Active Directory Architecture

Active Directory is a hierarchical directory service, based on the domain, that is scalable in both directions.

In AD DS, you can subdivide a domain into organizational units and populate it with objects. You can also create multiple domains and group them into sites, trees, and forests. As a result, AD DS provides a highly flexible architecture that can accommodate the smallest and the largest organizations, as well as provide various design options.

The following sections examine the components you can use to design and build an Active Directory structure.

UNDERSTANDING OBJECTS AND ATTRIBUTES

An AD DS domain is a hierarchical structure that takes the form of a tree, much like a file system. The domain consists of objects, each of which represents a logical or physical resource. Objects come in two basic classes: container objects and leaf objects. A ***container object*** can have other objects subordinate to it, whereas a ***leaf object*** cannot have subordinate objects.

The container objects essentially form the branches of the tree, with the leaf objects growing on the branches.

The domain itself is a container object, as are the organizational unit objects within the domain. Leaf objects can represent users, computers, groups, applications, and other resources on the network.

Every object consists of ***attributes***, which store information about the object. A container object has, as one of its attributes, a list of all the other objects it contains. Leaf objects have attributes that contain information about the specific resource the object represents. Some attributes are created automatically, such as the globally unique identifier (GUID) that the domain controller assigns to each object when it creates it, whereas you must supply information for other attributes manually.

UNDERSTANDING DOMAINS

The domain is the fundamental component of the Active Directory architecture. You can zoom into a domain and create a hierarchy within it, and you can zoom out and create a hierarchy out of multiple domains. In AD DS, domains function by default as the boundaries for virtually all directory functions, including administration, access control, database management, and replication. You begin the process of designing an Active Directory infrastructure by deciding what domains to create, and you begin deploying AD DS by creating your first domain.

ZOOMING IN: ORGANIZATIONAL UNITS

While the domain is the fundamental division in the Active Directory service, the extreme scalability that AD DS provides can result in domains containing many thousands of objects. When domains grow this large, dividing the security and administrative responsibility for the domain among several divisions or departments can become necessary. To make this possible, you can create objects within a domain called organizational units.

An ***organizational unit (OU)*** is a container object that functions in a subordinate capacity to a domain, something like a subdomain, but without the complete separation of security policies. As container objects, OUs can contain other OUs, as well as leaf objects. You can apply separate Group Policy settings to an OU and delegate the administration of an OU as needed. However, an OU is still part of the domain and still inherits policies and permissions from its parent objects.

ZOOMING IN: GROUPS

Group objects are not containers, as OUs are, but they perform a similar function, with important differences. Groups are not full-fledged security divisions, as OUs are; you cannot apply Group Policy settings to a group object directly. However, group members—which can be leaf objects, such as users or computers, as well as other groups—inherit permissions assigned to that group.

One of the most important differences between groups and OUs is that group memberships are independent of the domain's tree structure. A group can have members located anywhere in the domain and, in some cases, can have members from other domains.

ZOOMING OUT: DOMAIN TREES

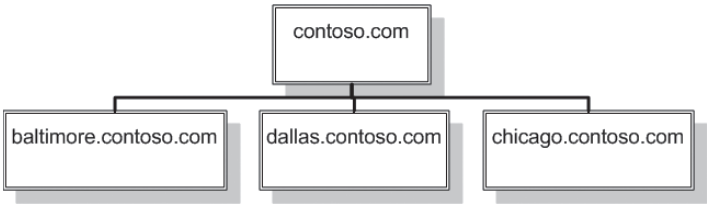
When designing an AD DS infrastructure, you might, in some cases, want to create multiple domains. Active Directory scales upward from the domain, just as easily as it scales downward.

Active Directory uses the Domain Name System (DNS) naming conventions for its domains. You can create an Active Directory domain using the registered domain name you use on the Internet, or you can create an internal domain name, without registering it.

When you create your first domain on an Active Directory network, you are, in essence, creating the root of a ***domain tree***. You can populate the tree with additional domains as long as they are part of the same contiguous namespace (see Figure 13-1). These subdomains are said to be part of the same tree as *contoso.com* because they use the same top- and second-level domain names.

Figure 13-1

An internal Active Directory domain tree



You can add as many domains to the tree as you need, using any number of levels, as long as you conform to the DNS naming limitations, which call for a maximum of 63 characters per domain name and 255 characters for the fully qualified domain name (FQDN).

ZOOMING OUT: FORESTS

An organization might want to use multiple domains that cannot be part of the same tree, because they are not contiguous. For example, a single corporation might run two operations out of the same facilities, each with its own Internet domain name, such as *contoso.com* and *adatum.com*. You can create two separate Active Directory domains using these two names, but they cannot be parts of the same tree, because one is not subordinate to the other. You can, however, create two separate domain trees and join them together in a parent structure called a ***forest***.

An Active Directory forest consists of one or more separate domain trees, which have the same two-way trust relationships between them as two domains in the same tree. When you create the first domain on an Active Directory network, you are in fact creating a new forest, and that first domain becomes the ***forest root domain***. Therefore, if you create the *contoso.com* domain first, that domain becomes the root of the *contoso.com* forest. When you create the *adatum.com* domain in the same forest, it retains its status as a separate domain in a separate tree, but it is still considered part of the *contoso.com* forest.

It is important to understand that separate trees in the same forest still have trust relationships between them, even though they do not share a domain name. If you want to create two domains completely separate from one another, you must create each one in a separate forest.

INTRODUCING THE GLOBAL CATALOG

Domains also function as the hierarchical boundaries for the AD DS database. A domain controller maintains only the part of the database that defines that domain and its objects. However, Active Directory clients still need a way to locate and access the resources of other domains in the same forest. To make this possible, each forest has a ***global catalog***, which lists all objects in the forest, along with a subset of each object's attributes.

To locate an object in another domain, Active Directory clients perform a search of the global catalog first. This search provides the client with the information it needs to find the object in the specific domain that contains it. Thus, the global catalog prevents the client from having to search all domains in the forest.

■ Deploying Active Directory Domain Services

THE BOTTOM LINE

To create a new domain, or to add a domain controller to an existing domain, you must install the Active Directory Domain Services role on a Windows Server 2012 R2 computer, and then run the *Active Directory Domain Services Configuration Wizard*.

To use a Windows Server 2012 R2 computer as a domain controller, you must configure it to use static IP addresses, not addresses supplied by a Dynamic Host Configuration Protocol (DHCP) server. If you are creating a domain in an existing forest, or adding a domain controller to an existing domain, you also must configure the computer to use the DNS server that hosts the existing forest or domain, at least during the Active Directory installation.

Installing the Active Directory Domain Services Role

Although the Active Directory Domain Services role does not actually convert the computer into a domain controller, installing it prepares the computer for the conversion process.

To install the Active Directory Domain Services role, use the following procedure.

INSTALL THE ACTIVE DIRECTORY DOMAIN SERVICES ROLE

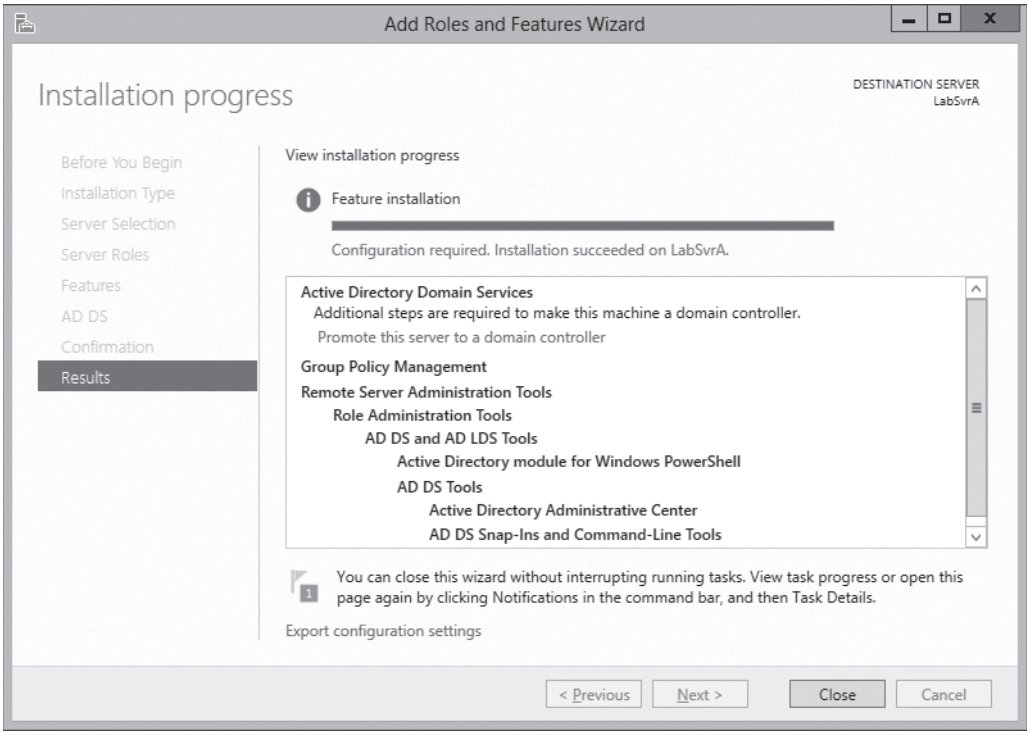
GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges.

1. From the *Server Manager*'s Manage menu, select Add Roles and Features. The *Add Roles and Features Wizard* appears, displaying the *Before you begin* page.
2. Click Next. The *Select Installation Type* page appears.
3. Leave the *Role-based or feature-based installation* radio button selected and click Next. The *Select Destination Server* page appears.
4. Select the server that you want to promote to a domain controller and click Next. The *Add Roles and Features Wizard* displays the *Select server roles* page.
5. Select the Active Directory Domain Service role. The *Add features that are required for Active Directory Domain Services?* page appears.
6. Click Add Features to accept the dependencies, and then click Next. The *Select features* page appears.
7. Click Next. The *Active Directory Domain Services* page appears, displaying information about the role.
8. Click Next. A *Confirm installation selections* page appears.
9. Select from the following optional functions, if desired:
   - Restart the destination server automatically if desired causes the server to restart automatically when the installation is complete, if the selected roles and features require it.
   - Export configuration settings creates an XML script documenting the procedures performed by the wizard, which you can use to install the same configuration on another server using Windows PowerShell.
   - Specify an alternate source path specifies the location of an image file containing the software needed to install the selected roles and features.

10. Click Install. The *Installation progress* page appears, as shown in Figure 13-2. After the role is installed, a *Promote this server to a domain controller* link appears.

Figure 13-2

The *Installation progress* page in the *Add Roles and Features Wizard*



PAUSE. Leave the wizard open.

After you install the role, you can proceed to run the *Active Directory Domain Services Installation Wizard*. The wizard procedure varies, depending on what function the new domain controller will serve. The following sections describe the procedures for the most common types of domain controller installations.

Creating a New Forest

When beginning a new AD DS installation, you first need to create a new forest, which you do by creating the first domain in the forest, the forest root domain.

To create a new forest, use the following procedure.

CREATE A NEW FOREST

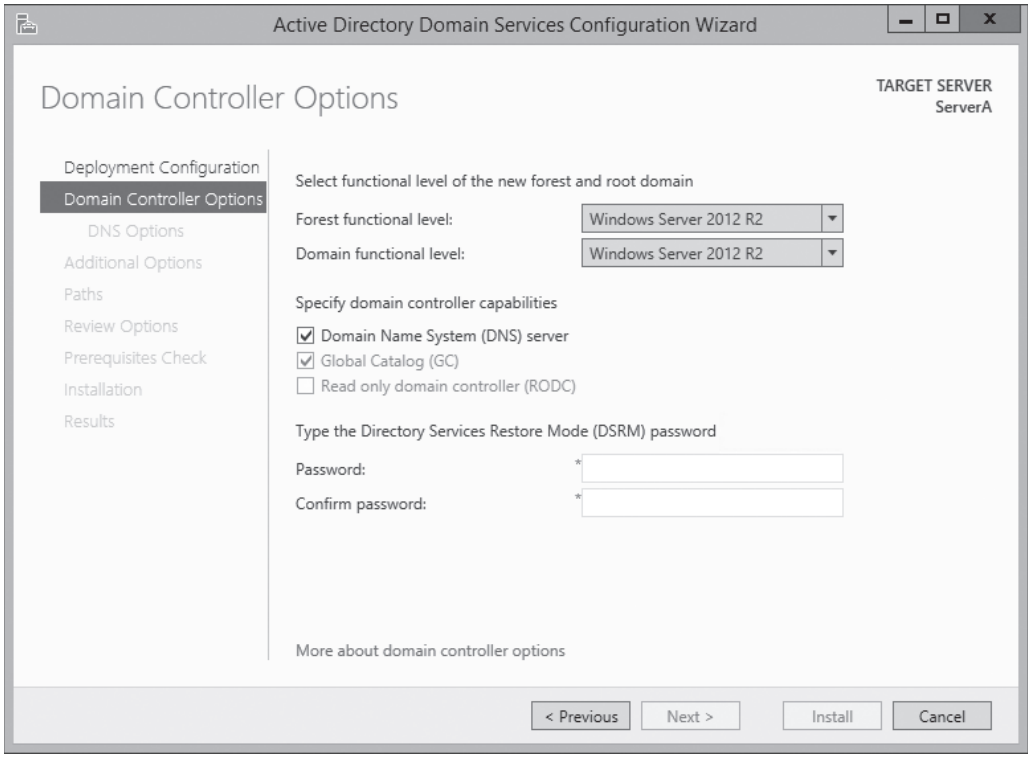
GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges and install the Active Directory Domain Services role, as described earlier in this lesson.

1. On the *Installation progress* page that appears at the end of the Active Directory Domain Services role installation procedure, click the Promote this server to a domain controller hyperlink. The *Active Directory Domain Services Configuration Wizard* appears, displaying the *Deployment Configuration* page.

2. Select the Add a new forest option and, in the Root domain name text box, type the name of the domain you want to create.
3. Click Next. The *Domain Controller Options* page appears, as shown in Figure 13-3.

Figure 13-3

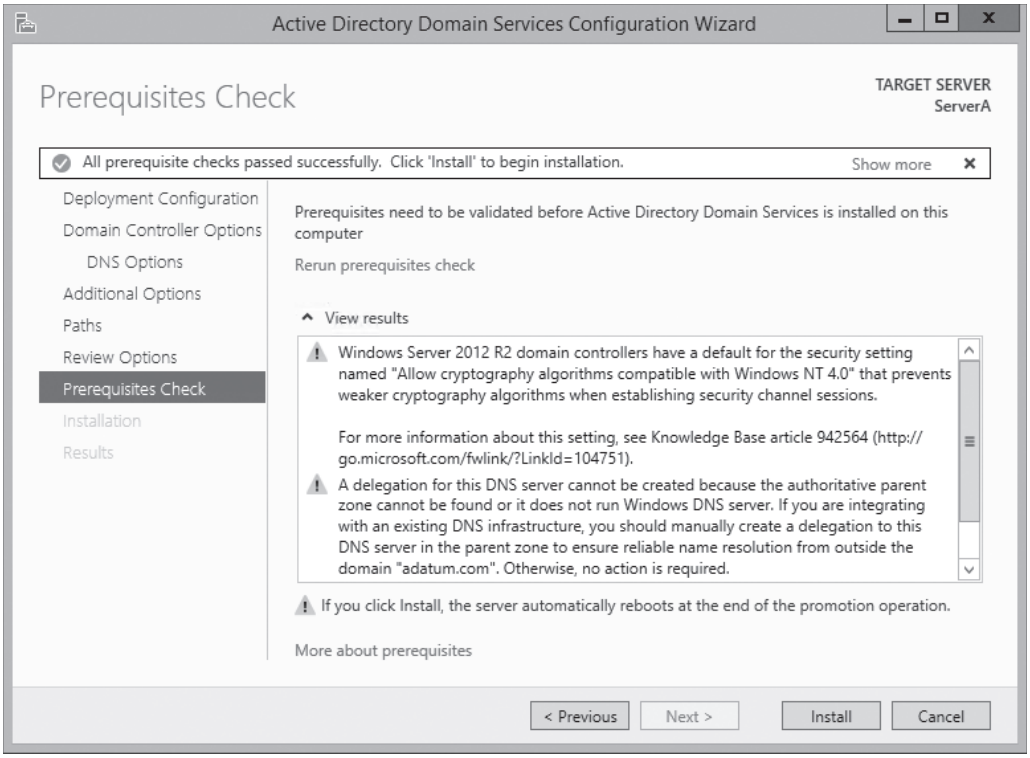
The *Domain Controller Options* page of the *Active Directory Domain Services Configuration Wizard*



4. Make changes on the *Domain Controller Options* page, as needed:
   - If you plan to add domain controllers running earlier versions of Windows Server to this forest, select the earliest Windows version you plan to install from the Forest functional level drop-down list.
   - If you plan to add domain controllers running earlier versions of Windows Server to this domain, select the earliest Windows version you plan to install from the Domain functional level drop-down list.
   - If you do not already have a DNS server on your network, leave the Domain Name System (DNS) server check box selected. If you have a DNS server on the network and the domain controller is configured to use that server for DNS services, clear the check box.
5. In the Password and Confirm password text boxes, type the password you want to use for Directory Services Restore Mode (DSRM) and click Next. The *DNS options* page appears, with a warning that a delegation for the DNS server cannot be created, because the DNS Server service is not installed yet.
6. Click Next. The *Additional Options* page appears displaying the NetBIOS equivalent of the domain name you specified.
7. Modify the name, if desired, and click Next. The *Paths* page appears.
8. Modify the default locations for the AD DS files, if desired, and click Next. The *Review Options* page appears.
9. Click Next. The *Prerequisites Check* page appears, as shown in Figure 13-4.

Figure 13-4

The *Prerequisites Check* page of the *Active Directory Domain Services Configuration Wizard*



The wizard performs a number of environment tests to determine whether the system can function as a domain controller. The results can appear as cautions, which enable the procedure to continue, or warnings, which require you to perform certain actions before the server can be promoted.

10. After the system passes all the prerequisite checks, click Install. The wizard creates the new forest and configures the server to function as a domain controller.

RESTART the computer.

With the forest root domain in place, you can now proceed to create additional domain controllers in that domain, or add new domains to the forest.

Adding a Domain Controller to an Existing Domain

Every Active Directory domain should have a minimum of two domain controllers.

To add a domain controller to an existing Windows Server 2012 R2 domain, use the following procedure.

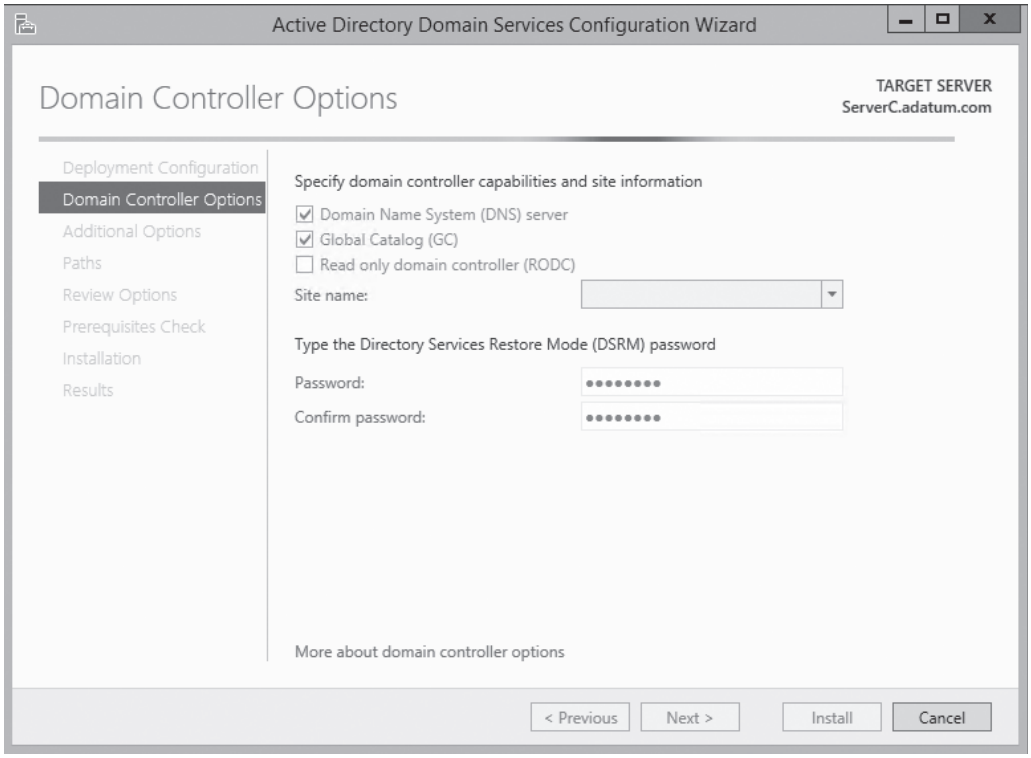
ADD A DOMAIN CONTROLLER TO AN EXISTING DOMAIN

GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges and install the Active Directory Domain Services role, as described earlier in this lesson.

1. On the *Installation progress* page that appears at the end of the Active Directory Domain Services role installation procedure, click the Promote this server to a domain controller hyperlink. The *Active Directory Domain Services Configuration Wizard* appears, displaying the *Deployment configuration* page.
2. Select the Add a domain controller to an existing domain option and then click Select.
3. If you are not logged on to an existing domain in the forest, a *Credentials for deployment operation* dialog box appears, in which you must supply administrative credentials for the domain to proceed. After you are authenticated, a *Select a domain from the forest* dialog box appears.
4. Select the domain to which you want to add a domain controller and click OK. The selected domain name appears in the *Domain* field.
5. Click Next. The *Domain Controller Options* page appears, as shown in Figure 13-5.

Figure 13-5

The *Domain Controller Options* page of the *Active Directory Domain Services Configuration Wizard*

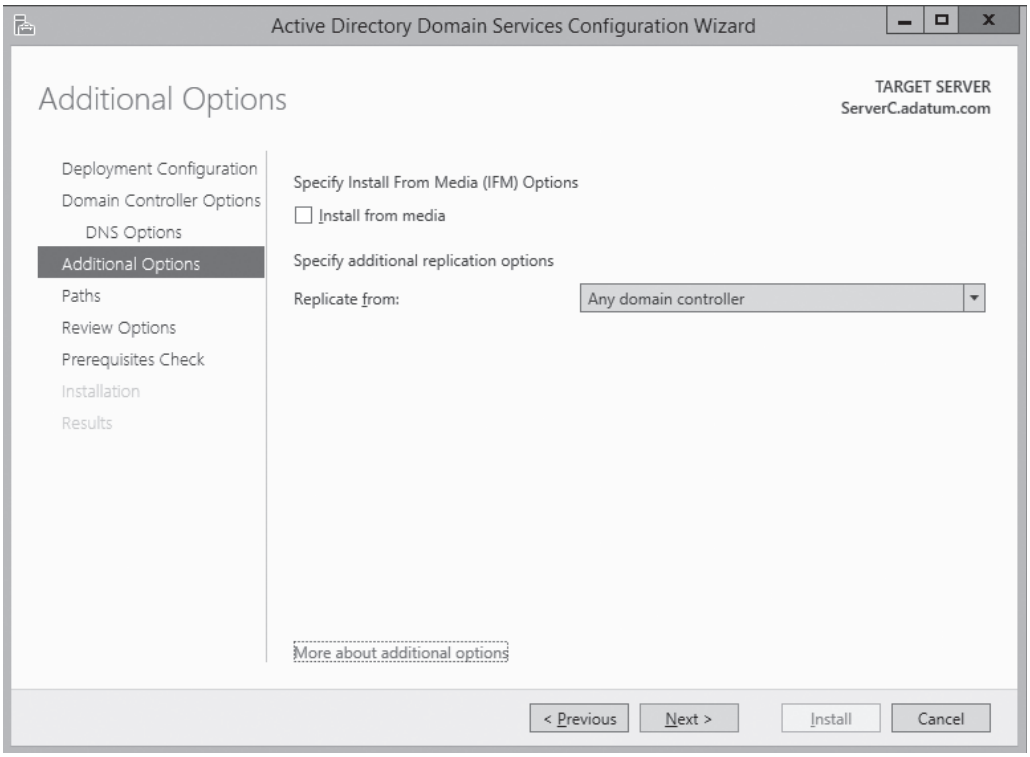


6. Select or clear the check boxes on the *Domain Controller Options* page as needed:
   - If you want to install the DNS Server service on the computer, leave the Domain Name System (DNS) server check box selected. Otherwise, the DNS server the computer is configured to use will host the domain.
   - Leave the Global Catalog (GC) check box selected if you want the computer to function as a global catalog server. This is essential if you are deploying the new domain controller at a site that does not already have a GC server.
   - Select the Read only domain controller (RODC) check box to create a domain controller that administrators cannot use to modify AD DS objects.
7. In the Site Name drop-down list, select the site where the domain controller will be located.

8. In the Password and Confirm Password text boxes, type the password you want to use for Directory Services Restore Mode (DSRM) and click Next. The *Additional Options* page appears, as shown in Figure 13-6.

Figure 13-6

The *Additional Options* page of the *Active Directory Domain Services Configuration Wizard*



9. To use the Install From Media (IFM) option, select the Install from media check box.
10. In the Replicate from drop-down list, select the existing domain controller that the server should use as a data source. Then click Next. The *Paths* page appears.
11. Modify the default locations for the AD DS files, if desired, and click Next. The *Review Options* page appears.
12. Click Next. The *Prerequisites Check* page appears.
13. After the system passes all the prerequisite checks, click Install. The wizard configures the server to function as a domain controller.

RESTART the computer.

The domain controller is now configured to service the existing domain. If the new domain controller is located in the same site as another, AD DS replication between the two begins automatically.

Installing AD DS on Server Core

In Windows Server 2012 R2, you can now install Active Directory Domain Services on a computer running the Server Core installation option and promote the system to a domain controller, all by using Windows PowerShell.

In Windows Server 2008 and Windows Server 2008 R2, the accepted method for installing AD DS on a computer using the Server Core installation option is to create an answer file and load it from the command prompt using the Dcpromo.exe program with the /unattend parameter.

In Windows Server 2012 and Windows Server 2012 R2, running Dcpromo.exe with no parameters no longer launches the *Active Directory Domain Services Configuration Wizard*. However, administrators who have already invested considerable time into developing answer files for unattended domain controller installations can continue to execute them from the command prompt, although doing so also produces a warning that *The dcpromo unattended operation is replaced by the ADDSDeployment module for Windows PowerShell*.

For AD DS installations on Server Core, Windows PowerShell is now the preferred method. As with the wizard-based installation, the Windows PowerShell procedure occurs in two phases: first, you must install the Active Directory Domain Services role; then, you must promote the server to a domain controller.

Installing the Active Directory Domain Services role via Windows PowerShell is no different from installing any other role. In an elevated Windows PowerShell session, use the following command:

```
Install-WindowsFeature –name AD-Domain-Services
-IncludeManagementTools
```

After you install the role, promoting the server to a domain controller is somewhat more complicated. The ADDSDeployment Windows PowerShell module includes separate cmdlets for the three deployment configurations covered in the previous sections:

- Install-AddsForest
- Install-AddsDomainController
- Install-AddsDomain

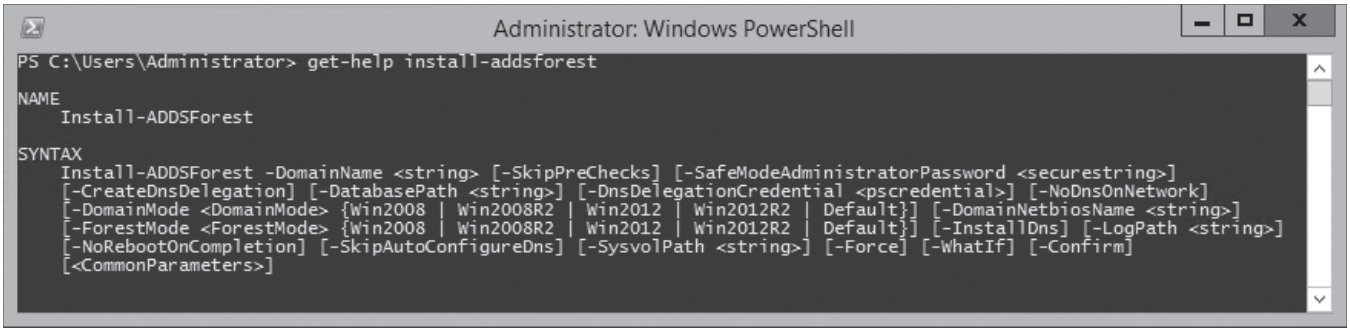
Each of these cmdlets has a great many possible parameters to support the many configuration options you find in the Active Directory Domain Services Configuration Wizard. In its simplest form, the following command would install a domain controller for a new forest called *adatum.com*:

```
Install-AddsForest -DomainName “adatum.com”
```

The defaults for all the cmdlet’s other parameters are the same as those in the *Active Directory Domain Services Configuration Wizard*. Running the cmdlet with no parameters steps through the options, prompting you for values. You can also display basic syntax information using the Get-Help command, as shown in Figure 13-7.

Figure 13-7

Syntax for the Install-AddsForest cmdlet in Windows PowerShell



Using Install from Media (IFM)

Earlier in this lesson, in the procedure for installing a replica domain controller, the *Additional Options* page of the *Active Directory Domain Services Configuration Wizard* included an *Install from media* check box. This option enables you to streamline the process of deploying replica domain controllers to remote sites.

Normally, installing a domain controller on an existing domain creates the AD DS database structure, but it has no data until the server can receive replication traffic from the other domain controllers. When the domain controllers for a particular domain are well connected, such as by a local area network (LAN), replication occurs automatically and almost immediately after the new domain controller is installed.

When you install a domain controller at a remote location, however, the connection to the other domain controllers is most likely a wide area network (WAN) link, which is typically slower and more expensive than a LAN connection. In this case, the initial replication with the other domain controllers can be much more of a problem. The slow speed of the WAN link might cause the replication to take a long time and might flood the connection, delaying regular traffic. If the domain controllers are located in different AD DS sites, no replication occurs until an administrator creates and configures the required site links.

By using a command-line tool called Ndtsutil.exe, you can avoid these problems by creating domain-controller installation media that include a copy of the AD DS database. By using this media when installing a remote domain controller, the data is installed along with the database structure, and no initial replication is necessary.

To create Install From Media (IFM) media, you must run the Ntdsutil.exe program on a domain controller running the same version of Windows that you intend to deploy. The program is interactive, requiring you to enter a sequence of commands such as the following:

- `Ntdsutil`-launches the program.
- `Activate instance ntds`-focuses the program on the installed AD DS instance.
- `Ifm`-switches the program into IFM mode.
- `Create Full|RODC <path name>`-creates media for either a full read/write domain controller or a read-only domain controller and saves it to the folder specified by the path name variable.

TAKE NOTE*

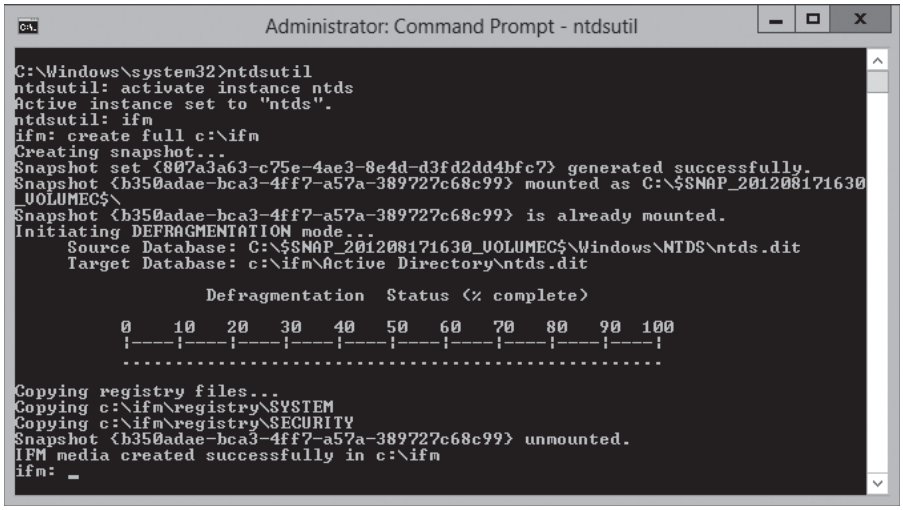
The Ntdsutil.exe create command also supports parameters that include the contents of the SYSVOL volume with the AD DS data. The Windows Server 2012 R2 version of the program adds a nodefrag parameter, which speeds up the media creation process by skipping the defragmentation.

When you execute these commands, the Ntdsutil.exe program creates a snapshot of the AD DS database, mounts it as a volume to defragment it, and then saves it to the specified folder, along with a copy of the Windows Registry (see Figure 13-8).

After you create the IFM media, you can transport it to the servers you intend to deploy as domain controllers by any convenient means. To use the media, you run the *Active Directory Domain Services Configuration Wizard* in the usual way, select the *Install from media* check box, and specify the path to the location of the folder.

Figure 13-8

An Ntdsutil.exe command sequence



Upgrading Active Directory Domain Services

Introducing Windows Server 2012 R2 onto an existing AD DS installation is easier than it has ever been in previous versions of the operating system.

You can upgrade an AD DS infrastructure in two ways. You can upgrade the existing down-level domain controllers to Windows Server 2012 R2, or you can add a new Windows Server 2012 R2 domain controller to your existing installation.

As noted in Lesson 1, "Installing Servers," the upgrade paths to Windows Server 2012 R2 are few. You can upgrade a Windows Server 2008 or Windows Server 2008 R2 domain controller to Windows Server 2012 R2, but no earlier versions are upgradable.

In the past, if you wanted to add a new domain controller to an existing AD DS installation based on previous Windows versions, you had to run the Adprep.exe program to upgrade the domains and forest. Depending on the installation's complexity, this could involve logging on to various domain controllers with different credentials, locating different versions of Adprep.exe, and running the program several times using /domainprep parameter for each domain and the /forestprep parameter for the forest.

In Windows Server 2012 R2, the Adprep.exe functionality has been fully incorporated into Server Manager in the *Active Directory Domain Services Configuration Wizard*. When you install a new Windows Server 2012 R2 domain controller, you have to supply only appropriate credentials, and the wizard takes care of the rest.

Adprep.exe, still included with the operating system, supports the old preparation method, if you prefer it, but you have no compelling reason to use it.

Deploying Active Directory IaaS on Windows Azure

In addition to running Windows Server 2012 R2 on physical computers and locally-hosted virtual machines, Microsoft's Windows Azure service enables administrators to run Windows Server 2012 R2 on cloud-based virtual machines provided by Microsoft. This capability, called Infrastructure as a Service (IaaS), enables administrators to run applications in the cloud while maintaining full control over the virtual machines themselves.

Windows Azure resources can be self-contained in the cloud, and administrators can create a virtualized AD DS forest to organize and manage them. It is also possible to configure Windows Azure resources as an extension to the existing physical and virtual resources hosted on a private network. For example, after creating a virtual network in the Windows Azure cloud and connecting it to your private network with a site-to-site link using a virtual private networking (VPN) device, you can create a Windows Server 2012 R2 virtual machine in the cloud and configure it as a domain controller for an existing domain.

The process of installing AD DS on a Windows Azure virtual machine and promoting it to a domain controller is no different from that of a private network server. You use the Add Roles and Features Wizard to install the AD DS role and then use the Active Directory Domain Services Configuration Wizard to configure the domain controller. The complicated part of the process is the configuration of the virtual network infrastructure to allow communication between the cloud network and your physical network.

Windows Azure is an idea platform for AD DS domain controllers because it provides IP address consistency in a new way. Windows Azure virtual machines must obtain IP addresses from DHCP servers – you cannot assign static IP addresses to them – but unlike standard DHCP address leases that can expire, causing the address to change, a cloud VM retains its IP address lease for its lifetime.

Removing a Domain Controller

With the loss of Dcpromo.exe, the process of demoting a domain controller has changed, and it is not immediately intuitive.

To remove a domain controller from an AD DS installation, you must begin by running *the Remove Roles and Features Wizard*, as shown in the following procedure.

REMOVE A REPLICA DOMAIN CONTROLLER

GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges.

1. From the Server Manager's Manage menu, select Remove Roles and Features. The *Remove Roles and Features Wizard* appears, displaying the *Before you begin* page.

2. Click Next. The *Select Destination Server* page appears.
3. Select the server that you want to demote from a domain controller and click Next. The *Remove Server Roles* page appears.
4. Clear the Active Directory Domain Services check box. A *Remove features that require Active Directory Domain Services* dialog box appears.
5. Click Remove Features. A *Validation Results* dialog box appears.
6. Click the Demote this domain controller hyperlink. The *Active Directory Domain Services Configuration Wizard* appears, displaying the *Credentials* page.
7. Click Change to supply alternative credentials for demoting the domain controller, if necessary.
8. Select the Force the removal of this domain controller check box and click Next. The *New Administrator Password* page appears.
9. In the Password and Confirm Password text boxes, type the password you want the server to use for the local Administrator account after the demotion. Then click Next. The *Review Options* page appears.
10. Click Demote. The wizard demotes the domain controller and restarts the system.
11. Log on using the local Administrator password you specified earlier.
12. From the Manage menu, select Remove Roles and Features. The *Remove Roles and Features Wizard* appears as before, displaying the *Before you begin* page.
13. Click Next. The *Select Destination Server* page appears.
14. Select the server that you want to demote from a domain controller and click Next. The *Remove Server Roles* page appears.
15. Clear the Active Directory Domain Services check box. A *Remove features that require Active Directory Domain Services* dialog box appears.
16. Click Remove Features, and then click Next. The *Remove Features* page appears.
17. Click Next. The *Confirm Removal Selections* page appears.
18. Click Remove. The wizard removes the role.

CLOSE the wizard and restart the server.

Configuring the Global Catalog

As noted earlier, the global catalog is an index of all AD DS objects in a forest that prevents systems from having to perform searches among multiple domain controllers.

The importance of the global catalog varies depending on the size of your network and its site configuration. For example, if your network consists of a single domain, with domain controllers all located at the same site and well connected, the global catalog serves little purpose other than universal group searches. You can make all your domain controllers global catalog servers, if you want. The searches will be load balanced, and the replication traffic likely will not overwhelm the network.

However, if your network consists of multiple domains with domain controllers located at multiple sites connected by WAN links, the global catalog configuration is critical. If at all possible, you do not want users performing AD DS searches that must reach across slow, expensive WAN links to contact domain controllers at other sites. Placing a global catalog server at each site is recommended in this case. The initial replication might generate a lot of traffic, but the savings in the long run should be significant.

When you promote a server to a domain controller, you have the option of making the domain controller a global catalog server. If you decline to do so, however, you can make any domain controller a global catalog server using the following procedure.

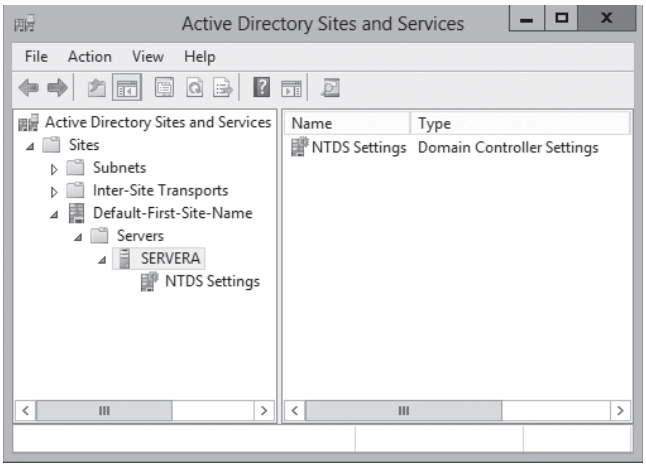
CREATE A GLOBAL CATALOG SERVER

GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges.

1. From the Server Manager's Tools menu, select Active Directory Sites and Services. The *Active Directory Sites and Services* console appears.
2. Expand the site where the domain controller you want to function as a global catalog server is located. Then expand the Servers folder and select the server you want to configure, as shown in Figure 13-9.

Figure 13-9

The Active Directory Sites and Services console



3. Right-click the NTDS Settings node for the server and, from the context menu, select Properties. The *NTDS Settings Properties* sheet appears.
4. Select the Global Catalog check box and click OK.

CLOSE the *Active Directory Sites and Services* console.

Troubleshooting DNS SRV Registration Failure

The Domain Name System (DNS) is essential to the operating of Active Directory Domain Services. To accommodate directory services such as AD DS, a special DNS resource record was created that enables clients to locate domain controllers and other vital AD DS services.

When you create a new domain controller, one of the most important parts of the process is the registration of the server in the DNS. This automatic registration is the reason an AD DS network must have access to a DNS server that supports the Dynamic Updates standard defined in RFC 2136.

If the DNS registration process fails, computers on the network cannot locate that domain controller, the consequences of which can be serious. Computers will be unable to use that domain controller to join the domain; existing domain members will be unable to log on; and other domain controllers will be unable to replicate with it.

DNS problems are, in most cases, due to general networking faults or DNS client configuration error. When troubleshooting these problems, you should first try pinging the DNS server, and then make sure that the TCP/IP client configuration has the correct addresses for the DNS servers it should be using.

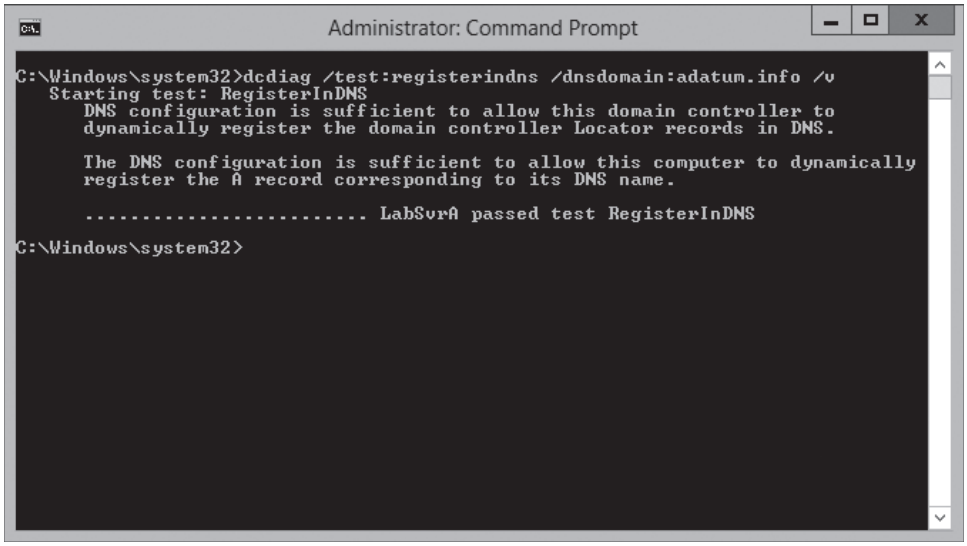
To confirm that a domain controller has been registered in the DNS, open a command-prompt window with administrative privileges and enter the following command:

```
dcdiag /test:registerindns /dnsdomain:<domain name> /v
```

A successful result appears, as shown in Figure 13-10.

Figure 13-10

A successful dcdiag test



■ Business Case Scenarios

Scenario 13-1: Creating AD DS Domains

Robert is designing a new Active Directory Domain Services infrastructure for a company called Litware, Inc., which has its headquarters in New York and two additional offices in London and Tokyo. The London office consists only of sales and marketing staff; it does not have its own IT department. The Tokyo office is larger, with representatives from all of the company departments, including a full IT staff. The Tokyo office is connected to the headquarters using a 64 Kbps demand-dial link, and the London office has a 512-Kbps frame relay connection. The company has registered the litware.com domain name, and Robert has created a subdomain called inside.litware.com for use by Active Directory.

Based on this information, design an Active Directory infrastructure for Litware, Inc. that is as economical as possible, specifying how many domains to create, what to name them, how many domain controllers to install, and where. Explain each of your decisions.

Scenario 13-2: Using Install from Media (IFM)

As you prepare some remote sites for domain controllers, you need the most efficient way to deploy replica DCs. You learn about Install from Media checkbox on the Additional Options page of the Active Directory Domain Services Configuration Wizard. Outline the steps taken in the interactive IFM wizard.

LESSON 14

Creating and Managing Active Directory Users and Computers

Creating User Objects

THE BOTTOM LINE

The user account is the primary means by which people using an Active Directory Domain Services (AD DS) network access resources.

Resource access for individuals takes place through their individual user accounts. To gain access to the network, prospective network users must authenticate to a network with a specific user account. Authentication is the process of confirming a user's identity by using a known value such as a password, a smart card, or a fingerprint. When a user supplies a name and password, the authentication process validates the credentials supplied in the logon against information that is stored within the AD DS database. Do not confuse authentication with authorization, which is the process of confirming that an authenticated user has the correct permissions to access one or more network resources.

The following two types of user accounts run on Windows Server 2012 R2 systems:

- ***Local users*** can access only resources on the local computer and are stored in the local ***Security Account Manager (SAM)*** database on the computer where they reside. Local accounts are never replicated to other computers, nor do these accounts provide domain access. A local account configured on one server cannot be used to access resources on a second server; you need to configure a second local account in that case.
- ***Domain users*** can access AD DS or network-based resources, such as shared folders and printers. Account information for these users is stored in the AD DS database and replicated to all domain controllers within the same domain. A subset of the domain user account information is replicated to the global catalog, which is then replicated to other global catalog servers throughout the forest.

By default, two built-in user accounts are created on a computer running Windows Server 2012 R2: the Administrator account and the Guest account. Built-in user accounts can be local accounts or domain accounts, depending on whether the server is a standalone server or a domain controller. In the case of a standalone server, the built-in accounts are local accounts on the server itself. On a domain controller, the built-in accounts are domain accounts that are replicated to each domain controller.

Creating Single Users

For some administrators, creating individual user accounts is a daily task, and there are many ways to go about it.

Windows Server 2012 R2 redesigned the Active Directory Administrative Center (ADAC) application, first introduced in Windows Server 2008 R2, to fully incorporate new features such as the Active Directory Recycle Bin and fine-grained password policies. You can also use the tool to create and manage AD DS user accounts.

To create a single user account with the Active Directory Administrative Center, use the following procedure.

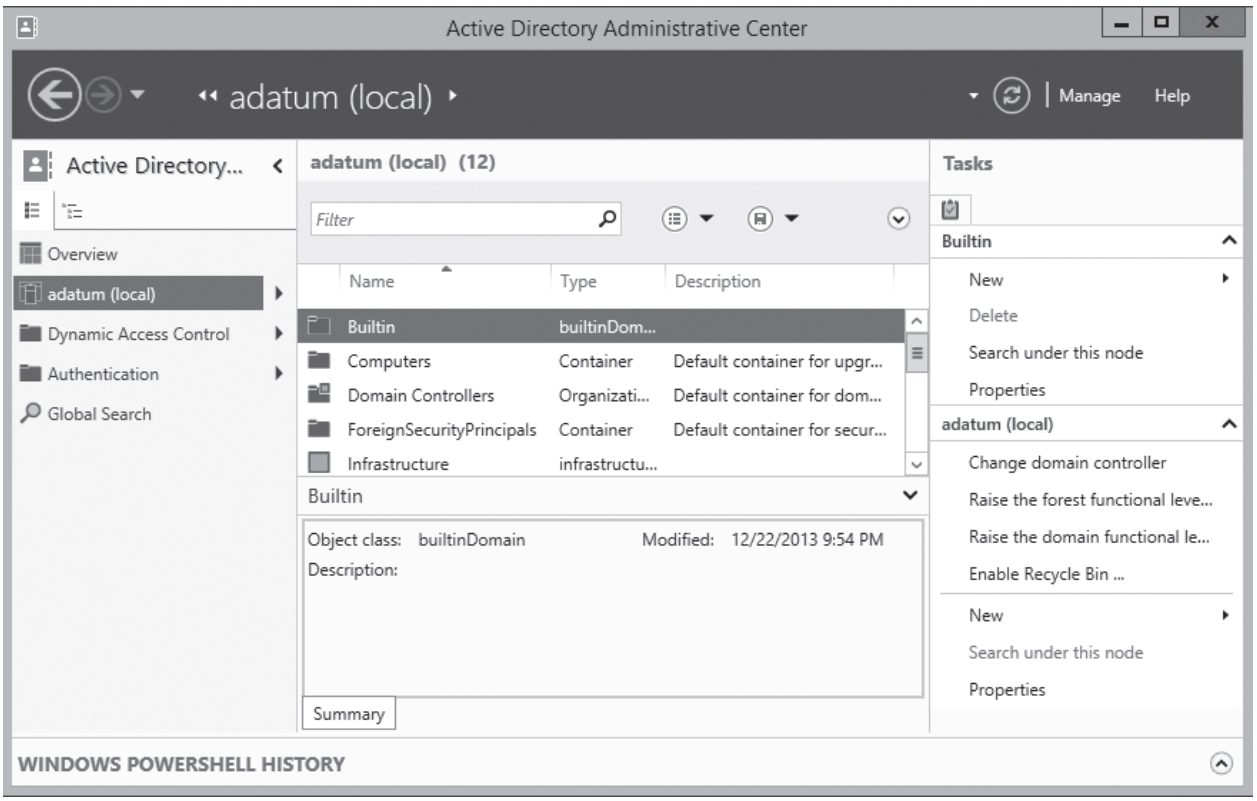
CREATE A USER WITH ACTIVE DIRECTORY ADMINISTRATIVE CENTER

GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges.

1. From the Tools menu in the *Server Manager* window, select Active Directory Administrative Center. The *Active Directory Administrative Center* console appears, as shown in Figure 14-1.

Figure 14-1

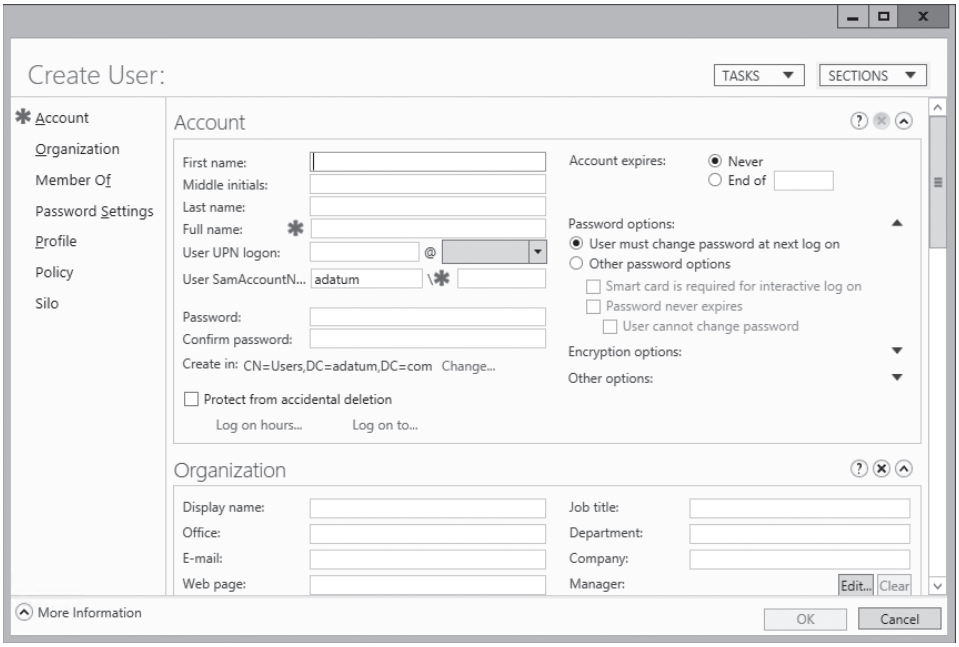
The Active Directory Administrative Center console



2. In the left pane, find the domain in which you want to create the user object and select a container in that domain.
3. In the Tasks pane, under the container name, click New > User. The *Create User* window appears, as shown in Figure 14-2.

Figure 14-2

The *Create User window* in the Active Directory Administrative Center console



4. Type the user's name in the Full Name field and an account name in the User SamAccountName Logon field.
5. Type an initial password for the user in the Password and Confirm password fields.
6. Supply information for any of the optional fields on the page you want.
7. Click OK. The user object appears in the container.

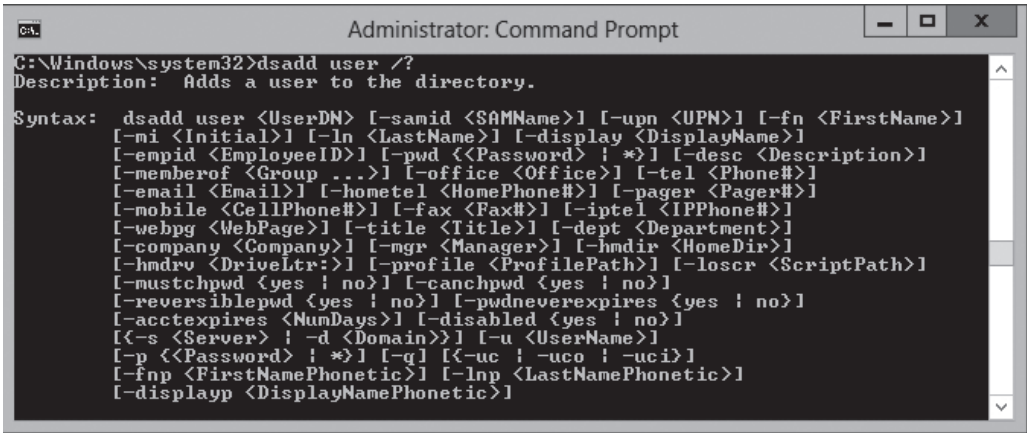
CLOSE the *Active Directory Administrative Center* console.

USING DSADD.EXE

For administrators more comfortable with the traditional command prompt, the Dsadd.exe program can create new user objects by using the syntax shown in Figure 14-3.

Figure 14-3

Syntax of the Dsadd.exe program



To create a user by using the Dsadd.exe utility, you must know the distinguished name (DN) for the user and the user's login ID, also known as the ***SAM account name*** attribute within AD DS. The distinguished name of an object signifies its location within the Active Directory structure. For example, in the distinguished name **cn=Elizabeth Andersen,ou=Research,dc=adatum,dc=com**, the "cn" refers to the common name for Elizabeth Andersen's user account, which resides in the Research OU, which resides in the adatum.com domain.

Creating User Templates

In some cases, you need to create single users on a regular basis, but the user accounts contain so many attributes that creating them individually becomes time-consuming.

One way to speed up the process of creating complex user objects is to use the New-ADUser cmdlet or the Dsadd.exe program and retain your commands in a script or batch file. However, if you prefer a graphical interface, you can do roughly the same thing by creating a user template.

A user template is a standard user object containing boilerplate attribute settings. To create a new user with these settings, you copy the template to a new user object and change the name and any other attributes that are unique to the user.

To create a user template with the Active Directory Users and Computers console, use the following procedure.

CREATE A USER TEMPLATE

GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges.

1. From the Tools menu in the *Server Manager* window, select Active Directory Users and Computers. The *Active Directory Users and Computers* console appears.
2. In the left pane, find the domain in which you want to create the user object and select a container in that domain.
3. From the Action menu, select New > User. The *New Object – User Wizard* appears.
4. Type Default Template, or a similarly descriptive name, in the Full Name field and an account name in the User logon name field.
5. Click Next. The second page of the *New Object – User Wizard* appears.
6. Type an initial password for the user in the Password and Confirm password fields.
7. Clear the *User must change password at next logon* check box.
8. Select the Account is disabled check box and click Next. A confirmation page listing the settings you configured appears.
9. Click Finish. The wizard creates the user object and closes.
10. Locate the user you just created in the console and double-click it to open its Properties sheet.
11. Modify the attributes on the various tabs with values common to all the users you create.
12. Click OK.

CLOSE the *Active Directory Users and Computers* console.

To use the template, right-click the *Default Template* user object and, from the context menu, select *Copy*. The *Copy Object – User Wizard* appears.

Enter the required unique information for the user and clear the *Account is disabled* check box before clicking *OK*. The wizard creates a new user object with all the attributes you configured in the template.

Creating Multiple Users

Administrators sometimes have to create hundreds or thousands of user objects, which makes the single object creation procedures impractical.

The previous sections in this lesson describe the procedures for creating single users and group objects by using the graphical user interface (GUI) and some of the available command-line tools in Windows Server 2012 R2. The following sections examine some of the mechanisms for automating the creation of large numbers of Active Directory objects.

USING CSVDE.EXE

Some applications such as Microsoft Excel can generate a number of users, along with their accompanying information, to add to the AD DS database. In these cases, you can export information from the applications by saving it to a file in ***Comma-Separated Values (CSV)*** format. CSV format also can be used to import information into and export it from other third-party applications. For example, you might need to export user account information to a file for use in another third-party application, such as a UNIX database. CSVDE.exe provides this capability as well.

A CSV file is a plain text file that consists of records, each on a separate line that are divided into fields and separated by commas. The format saves database information in a universally understandable way.

The CSVDE.exe command-line utility enables you to import or export Active Directory objects. It uses a CSV file that is based on a header record, which identifies the attribute contained in each comma-delimited field. The ***header record*** is the first line of the text file that uses proper attribute names. To import into AD DS, the attribute names in the CSV file must match the attributes allowed by the Active Directory schema. For example, to import a list of people and telephone numbers as users into the Active Directory database, you need to create a header record that accurately reflects the object names and attributes you want to create. Review the following attributes that are commonly used for creating user accounts.

- *dn* specifies the distinguished name of the object so that the object can be properly placed in Active Directory.
- *samAccountName* populates the SAM account field
- *objectClass* specifies the type of object to be created, such as user, group, or OU.
- *telephoneNumber* populates the Telephone Number field.
- *userPrincipalName* populates the User Principal Name field for the account.

As you create your CSV file, you must order the data to reflect the sequence of the attributes in the header record. If fields and data are out of order, you will either encounter an error when running the CSVDE.exe utility or you might not get accurate results in the created objects. The following header record example uses the previously listed attributes to create a user object.

```
dn,samAccountName,userPrincipalName,telephoneNumber,
objectClass
```

A data record conforming to this header record appears as follows:

```
"cn=Elizabeth
Andersen,ou=Research,dc=adatum,dc=com",eander,eander@
adatum.com,586-555-1234,user
```

After you add a record for each account you want to create, save the file using .csv as the extension. You then use the following command syntax to run the CSVDE.exe program and import the file.

```
csvde.exe -i -f <filename.csv>
```

The -i switch tells CSVDE.exe that this operation will import data. The -f switch specifies the .csv file containing the records to be imported.

USING LDIFDE.EXE

LDIFDE.exe is a utility that has the same basic functionality as CSVDE.exe and provides the capability to modify existing records in Active Directory. For this reason, LDIFDE.exe is a more flexible option. Consider an example where you need to import 200 new users into your AD DS structure. In this case, you can use CSVDE.exe or LDIFDE.exe to import the users. However, you can use LDIFDE.exe to modify or delete the objects later, whereas CSVDE.exe does not provide this option.

You can use any text editor to create the LDIFDE.exe input file, which is formatted according to the ***LDAP Data Interchange Format (LDIF)*** standard. The format for the data file containing the object records you want to create is significantly different from CSVDE.exe. The following example shows the syntax for a data file to create the same user account discussed in the CSVDE.exe example.

```
dn: "cn=Elizabeth
Andersen,ou=Research,dc=adatum,dc=com"

changetype: add
ObjectClass: user
SAMAccountName: eander
UserPrincipalName: eander@adatum.com
telephoneNumber: 586-555-1234
```

By using LDIFDE.exe, you can specify one of three actions to perform with the LDIF file:

- *Add* creates new objects using the LDIF records.
- *Modify* modifies existing object attributes using the LDIF records.
- *Delete* deletes existing objects using the LDIF records.

After creating the data file and saving it using the .LDF file extension, use the following syntax to execute the LDIFDE.exe program.

```
ldifde –i –f <filename.ldf>
```

The next example illustrates the LDIF syntax to modify the telephone number of an existing user object. Note that the hyphen in the last line is required for the file to function correctly.

```
dn: "cn=Elizabeth
Andersen,ou=Research,dc=adatum,dc=com"

changetype: modify
replace: telephoneNumber
telephoneNumber: 586-555-1111
-
```

■ Creating Computer Objects

THE BOTTOM LINE

Because an AD DS network uses a centralized directory, you need a way to track the computers that are part of the domain. To do this, Active Directory uses computer accounts, which are realized in the form of computer objects in the Active Directory database. You might have a valid Active Directory user account and a password, but if your computer is not represented by a computer object, you cannot log on to the domain.

Computer objects are stored in the Active Directory hierarchy just as user objects are, and they possess many of the same capabilities, such as the following:

- Computer objects consist of properties that specify the computer's name, where it is located, and who is permitted to manage it.
- Computer objects inherit group policy settings from container objects such as domains, sites, and organizational units.
- Computer objects can be members of groups and inherit permissions from group objects.

When a user attempts to log on to an Active Directory domain, the client computer establishes a connection to a domain controller to authenticate the user's identity. However, before the user authentication occurs, the two computers perform a preliminary authentication using their respective computer objects, to ensure that both systems are part of the domain. The NetLogon service running on the client computer connects to the same service on the domain controller, and then each one verifies that the other system has a valid computer account. When this validation is completed, the two systems establish a secure communications channel between them, which they can then use to begin the user authentication process.

The computer account validation between the client and the domain controller is a genuine authentication process using account names and passwords, just as when a user authenticates to the domain. The difference is that the passwords used by the computer accounts generate automatically and keep hidden. You can reset a computer account, but you do not need to supply passwords for everyone.

What all this means for administrators is that, in addition to creating user accounts in the domain, they have to make sure that the network computers are part of the domain as well. Adding a computer to an AD DS domain consists of two steps:

- **Creating a computer account:** You create a computer account by creating a new computer object in Active Directory and assigning the name of an actual computer on the network.
- **Joining the computer to the domain:** When you join a computer to the domain, the system contacts a domain controller, establishes a trust relationship with the domain, locates (or creates) a computer object corresponding to the computer's name, alters its security identifier (SID) to match the computer object, and modifies its group memberships.

How these steps are performed, and who performs them, depends on the way in which you deploy computers on your network. You can create new computer objects in many ways, but how you elect to do this depends on several factors, including the number of objects you need to create, where you will be when creating the objects, and what tools you prefer to use.

Generally speaking, you create computer objects when you deploy new computers in the domain. After an object represents a computer and joins to the domain, any user in the domain can log on from that computer. For example, you do not need to create new computer objects or rejoin computers to the domain when employees leave the company and new hires start using their computers. However, if you reinstall the operating system on a computer, you must create a new computer object for it (or reset the existing one), because the newly installed computer will have a different SID.

The creation of a computer object must always occur before the corresponding computer can join the domain, although it sometimes does not appear that way. You can use the following two basic strategies for creating Active Directory computer objects:

- Create the computer objects in advance by using an Active Directory tool, so that the computers can locate the existing objects when they are in the domain.
- Begin the joining process first and let the computer create its own computer object.

In each case, the computer object exists before the joining takes place. In the second strategy, the joining process appears to begin first, but the computer creates the object before the joining process begins.

When you have a number of computers to deploy, particularly in different locations, you might prefer to create the computer objects in advance. For large numbers of computers, you can automate the computer object creation process by using command-line tools and batch files.

Computer objects have relatively few attributes, and in most cases, you will most likely just supply them with a name, which can be up to 64 characters long. This name must match the name of the computer joined with the object.

■ Managing Active Directory Objects

THE BOTTOM LINE After you create user and computer objects, you can manage them and modify them in many of the same ways that you created them.

Double-clicking any object in the ADAC or the Active Directory Users and Computers console opens the Properties sheet for that object. The windows appear different, but they contain the same information and provide the same capability to alter the object attributes.

Joining Computers to a Domain

The process of joining a computer to a domain must occur at the computer itself and be performed by a member of the computer's local Administrators group.

After logging on, you join a computer running Windows Server 2012 R2 to a domain from the *Computer Name* tab in the System Properties sheet. You can access the System Properties sheet from Server Manager, by clicking the *Computer name or domain* hyperlink on the server's *Properties tile*, from the *Control Panel*.

On a computer that is not joined to a domain, the Computer Name tab displays the name assigned to the computer during the operating system installation, and the name of the workgroup to which the system currently belongs (which is WORKGROUP, by default). To join the computer to the domain, click *Change* to display the *Computer Name Changes* dialog box.

In this dialog box, the Computer name field enables you to change the name assigned to the computer during installation. Depending on whether you have already created a computer object, observe the following precautions:

- To join a domain in which you have already created a computer object for the system in AD DS, the name in this field must match the name of the object exactly.
- If you intend to create a computer object during the joining process, the name in this field must not already exist in the domain.

When you select the *Domain* option button and enter the name of the domain the computer will join, the computer establishes contact with a domain controller for the domain and a second *Computer Name Changes* dialog box appears, which prompts you for the name and password of a domain user account with permission to join the computer to the domain.

After you authenticate with the domain controller, the computer is welcomed to the domain and you are instructed to restart the computer.

CREATING COMPUTER OBJECTS WHILE JOINING

You can join a computer to a domain whether or not you have already created a computer object for it. After the computer authenticates to the domain controller, the domain controller scans the Active Directory database for a computer object with the same name as the computer. If it does not find a matching object, the domain controller creates one in the Computers container, using the name supplied by the computer.

To create the computer object automatically in this manner, you might expect that the user account you specify when connecting to the domain controller must have object creation privileges for the Computers container, such as membership in the administrators group. However, this is not always the case.

Domain users also can create computer objects themselves through an interesting, indirect process. The Default Domain Controllers Policy GPO grants a user right called *Add Workstations To The Domain* to the Authenticated Users special identity. Any user who is successfully authenticated to Active Directory is permitted to join up to ten workstations to the domain, and create 10 associated computer objects, even if the user does not possess explicit object creation permissions.

With Add Workstations To The Domain user right, "workstations" is the operative word. Authenticated users can add up to 10 workstations to the domain, but not servers.

JOINING A DOMAIN WHILE OFFLINE

It is typical for you to join computers to domains while the computers are connected to the network and have access to a domain controller. However, there are situations in which you might want to set up computers without access to a domain controller, such as a new branch office installation. In these cases, it is possible to perform an offline domain join, by using a command-line program called *Djoin.exe*.

The offline domain join procedure requires you to run the Djoin.exe program twice, once on a computer with access to a domain controller, and then again on the computer to be joined. When connected to the domain controller, the program gathers computer account metadata for the system to be joined and saves it to a file. The syntax for this phase of the process is as follows:

```
djoin /provision /domain <domain name>
/machine <computer name> /savefile <filename.txt>
```

You then transport the metadata file to the computer to be joined and run Djoin.exe again, specifying the name of the file. The program saves the metadata from the file to the computer, so that the next time it has access to a domain controller, the system is automatically joined to the domain. The syntax for the second phase of the process is as follows:

```
djoin /requestODJ /loadfile <filename.txt>
/windowspath %SystemRoot% /localos
```

Managing Disabled Accounts

Disabling a user account prevents anyone from using it to log on to the domain until an administrator with the appropriate permissions enables it again.

You can disable user accounts manually, to prevent their use while preserving all their attributes, but it is also possible for a system to automatically disable them. For example, repeated violations of password policy settings can disable an account, to prevent intruders from making further attack attempts.

To disable or enable a user or computer account in ADAC or Active Directory Users and Computers, you simply right-click the object and select *Disable* or *Enable* from the context menu. You can also disable and enable multiple accounts by selecting multiple objects and right-clicking.

To disable or enable a user or computer account with Windows PowerShell, use the following cmdlet syntax:

```
Disable-ADAccount –Identity <account name>
Enable-ADAccount –Identity <account name>
```

■ Business Case Scenarios

Scenario 14-1: Creating User Objects

You are a network administrator who is in the process of building an Active Directory network for a company called Fabrikam, Inc., and you have to create user objects for the 75 users in the Inside Sales department. You have already created the fabrikam.com domain and an OU called Inside Sales for this purpose. The Human Resources department has provided you with a list of the users' names and has instructed you to create the account names by using the first initial and the last name. Each user object must also have the value Inside Sales in the Department property and Fabrikam, Inc. in the Company property. Using the first name in the list, Oliver Cox, as an example, which of the following command-line formats would enable you to create the 75 user objects, with the required property values?

a. dsadd "Oliver Cox" –company "Fabrikam, Inc." –dept "Inside Sales"
b. dsadd user CN=Oliver Cox,CN=Inside Sales,DC=fabrikam,DC=com –company Fabrikam, Inc. –dept Inside Sales
c. dsadd –company "Fabrikam, Inc." –dept "Inside Sales" "CN=Oliver Cox,CN=Inside Sales,DC=fabrikam,DC=com"
d. dsadd user "CN=Oliver Cox,CN=Inside Sales,DC=fabrikam,DC=com" –company "Fabrikam, Inc." –dept "Inside Sales"

Scenario 14-2: Considering Security Guidelines

You are preparing a new branch office with new computers. You would like to join the computers to the domain. Unfortunately, the branch office network is available. How would you proceed?

LESSON 15

Creating and Managing Active Directory Groups and Organizational Units

■ Working with Organizational Units

THE BOTTOM LINE

OUs can be nested to create a design that enables administrators to take advantage of the natural inheritance of the Active Directory hierarchy. You should limit the number of nested OUs, because too many levels can slow the response time to resource requests and complicate the application of Group Policy settings.

When you install Active Directory Domain Services (AD DS), there is only one OU in the domain, by default: the Domain Controllers OU. The domain administrator must create all other OUs.

You can find another type of container object in a domain, called a *container*. For example, a newly created domain has several container objects, including one called *Users*, which contains the domain's predefined users and groups, and another called *Computers*, which contains the computer objects for all the systems joined to the domain.

Unlike organizational units, you cannot assign Group Policy settings to computer objects, nor can you delegate their administration. You also cannot create new container objects by using the standard Active Directory administration tools, such as the Active Directory Users and Computers console. You can create container objects by using scripts, but there is no compelling reason to do so. Organizational units are the preferred method of subdividing a domain.

Creating OUs

The OU is the easiest object type to create in the AD DS hierarchy. You need only supply a name for the object and define its location in the Active Directory tree.

To create an organizational unit object by using the Active Directory Administrative Center, use the following procedure.

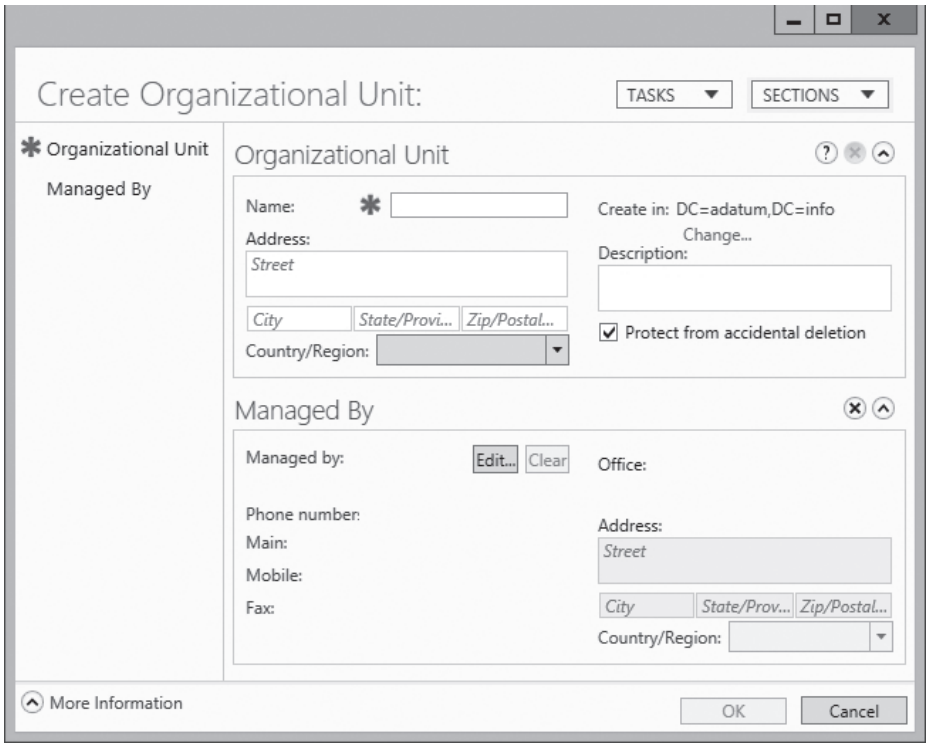
CREATE AN OU WITH ACTIVE DIRECTORY ADMINISTRATIVE CENTER

GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges.

1. From the Tools menu in the *Server Manager* window, select Active Directory Administrative Center. The *Active Directory Administrative Center* console appears.
2. In the left pane, right-click the object beneath which you want to create the new OU and, from the context menu, select New > Organizational Unit. The *Create Organizational Unit* window appears, as shown in Figure 15-1.

Figure 15-1

The *Create Organizational Unit* window in the Active Directory Administrative Center console



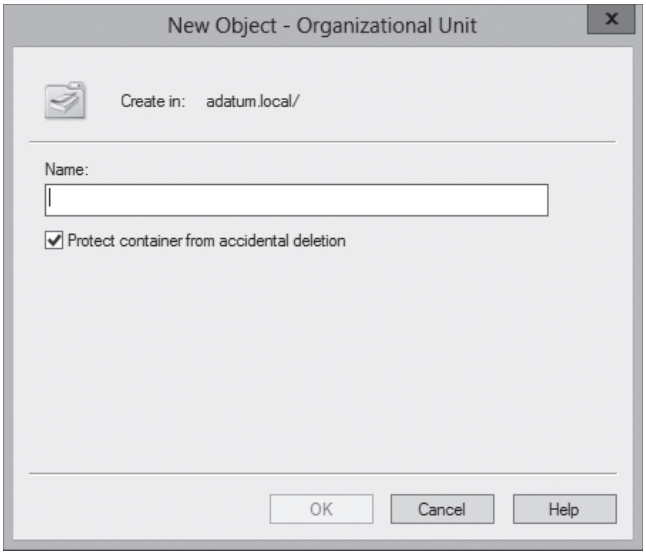
3. In the Name field, type a name for the OU and add any optional information you want.
4. Click OK. The organizational unit object appears in the container.

CLOSE the *Active Directory Administrative Center* console.

Creating an OU in the Active Directory Users and Computers console works in a similar way, although the *New Object – Organizational Unit* dialog box is different in appearance, as shown in Figure 15-2.

Figure 15-2

The *New Object – Organizational Unit* dialog box in the Active Directory Users and Computers console



After you create an OU, you can double-click it to open its Properties sheet, in which you can modify its attributes, or right-click it and select *Move*, to open the *Move* dialog box.

Using OUs to Delegate Active Directory Management Tasks

Creating OUs enables you to implement a decentralized administration model, in which others manage portions of the AD DS hierarchy, without affecting the rest of the structure.

Delegating authority at a site level affects all domains and users within the site. Delegating authority at the domain level affects the entire domain. However, delegating authority at the OU level affects only that OU and its subordinate objects. By granting administrative authority over an OU structure, as opposed to an entire domain or site, you gain the following advantages:

- **Minimal number of administrators with global privileges:** By creating a hierarchy of administrative levels, you limit the number of people who require global access.
- **Limited scope of errors:** Administrative mistakes, such as a container deletion or group object deletion, affect only the respective OU structure.

The ***Delegation of Control Wizard*** provides a simple interface you can use to delegate permissions for domains, OUs, or containers. AD DS has its own system of permissions, much like those of NTFS and printers. The Delegation of Control Wizard is essentially a front-end interface that creates complex combinations of permissions based on specific administrative tasks.

The wizard interface enables you to specify the users or groups to which you want to delegate management permissions and the specific tasks you want them to be able to perform. You can delegate predefined tasks or create custom tasks that enable you to be more specific.

DELEGATE ADMINISTRATIVE CONTROL OF AN OU

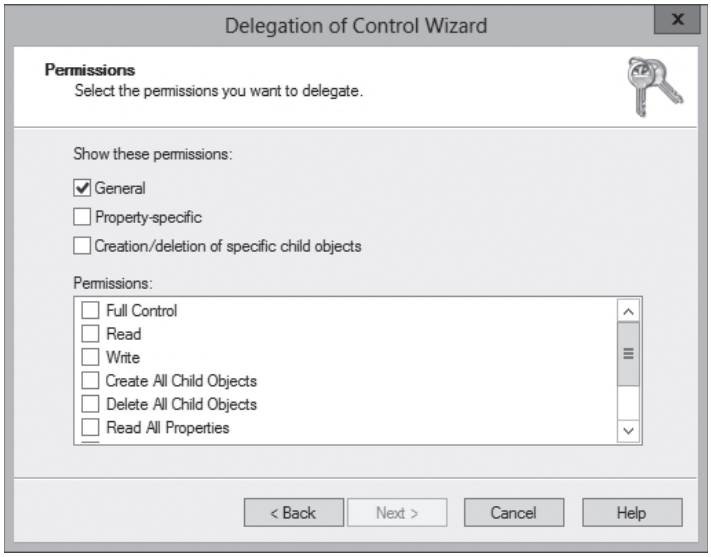
GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges.

1. From the Tools menu in the *Server Manager* window, select Active Directory Users and Computers. The *Active Directory Users and Computers* console appears.

2. Right-click the object over which you want to delegate control, and click Delegate Control. The *Delegation of Control Wizard* appears, displaying the *Welcome* page.
3. Click Next. The *Users or Groups* page appears.
4. Click Add. The *Select Users, Computers, or Groups* dialog box appears.
5. Type the name of the user or group to which you want to delegate control of the object, and click OK. The user or group appears in the *Selected users and groups* list.
6. Click Next. The *Tasks to Delegate* page appears with the following options:
   - **Delegate the following common tasks:** This option enables you to choose from a list of predefined tasks.
   - **Create a custom task to delegate:** This option enables you to be more specific about the task delegation.
7. Select Create a custom task to delegate and click Next. The *Active Directory Object Type* page appears, displaying the following options.
   - **This folder, existing objects in this folder, and creation of new objects in this folder:** This option delegates control of the container, including all its current and future objects.
   - **Only the following objects in the folder:** This option enables you to select specific objects to be controlled. You can select *Create selected objects in this folder* to allow selected object types to be created, or select *Delete selected objects in this folder* to allow selected object types to be deleted.
8. Select This folder, existing objects in this folder, and creation of new objects in this folder and click Next. The *Permissions* page appears, as shown in Figure 15-3.

Figure 15-3

The Permissions page of the Delegation of Control Wizard



9. Set the delegated permissions according to your needs for the user or group to which you delegate control. You can combine permissions from all three of the following options:
   - **General:** displays general permissions, which are equal to those displayed on the *Security* tab in an object's properties. For example, selecting Full Control for general permissions is inclusive of all property rights as well.

- **Property-specific:** displays permissions that apply to specific attributes or properties of an object. If you select the Read permission using the General option, all read-specific properties are selected.
- **Creation/deletion of specific child objects:** displays permissions that apply to creation and deletion permissions for specified object types.

10. Click Next. The *Completing the Delegation of Control Wizard* page appears.
11. Click Finish.

CLOSE the *Active Directory Users and Computers* console.

In this procedure, you grant permissions over a portion of Active Directory to a specified administrator or group of administrators. Although you can use the *Delegation of Control Wizard* to grant permissions, you cannot use it to modify or remove permissions. To perform these tasks, you must use the interface provided in the *Security* tab in the AD DS object's Properties sheet.

■ Working with Groups

THE BOTTOM LINE

Since the early days of the Microsoft server operating system, administrators have used ***groups*** to manage network permissions. Groups enable you to assign permissions to multiple users simultaneously. A group can be defined as a collection of user or computer accounts that functions as a security principal, in much the same way that a user does.

In Windows Server 2012 R2, when a user logs on to Active Directory, an ***access token*** is created that identifies the user and all the user's group memberships. Domain controllers use this access token to verify a user's permissions when the user attempts to access a local or network resource. By using groups, you can grant multiple users the same permission level for resources on the network. If, for example, you have 25 users in the graphics department who need access to a color printer, you can either assign each user the appropriate permissions for the printer, or you can create a group containing the 25 users and assign the appropriate permissions to the group. By using a group object to access a resource, you accomplish the following:

- When users need access to the printer, you can add them to the group. Once added, the user receives all permissions assigned to this group. Similarly, you can remove users from the group when you want to revoke their access to the printer.
- You need to make only one change to modify the level of access to the printer for all the users. Changing the group's permissions changes the permission level for all group members. Without the group, you need to modify all 25 user accounts individually.

Users can be members of more than one group. In addition, groups can contain other Active Directory objects, such as computers, and other groups in a technique called ***group nesting***. Group nesting describes the process of configuring one or more groups as members of another group. For example, consider a company that has two groups: marketing and graphic design. Graphic design group members have access to a high-resolution color laser printer. If the marketing group personnel also need access to the printer, you can add the marketing group as a member of the graphic design group. This gives the marketing group members the same permission to the color laser printer as the members of the graphic design group.

Working with Default Groups

Although there are many situations in which you might create groups to organize users or computers and then assign permissions, there are several built-in security groups that the system creates when you install AD DS on Windows Server 2012 R2. Many of the built-in groups have predefined user rights that enable their members to perform certain system-related tasks, such as backup and restore. You can add accounts to the default groups, to grant users the same rights, in addition to any resource access permissions the groups possess.

The default groups are located in the Built-in and Users container objects in AD DS. The list of predefined groups in these containers varies depending on the installed services. For example, installing the Dynamic Host Configuration Protocol (DHCP) server role creates two new groups in the Users container, called *DHCP Administrators* and *DHCP Users*.

All the default groups are security groups. Active Directory does not include any default distribution groups. The Built-in container, holds domain local groups, and the Users container has groups of various scopes.

You can view the groups described in the Built-in and Users containers and manage their memberships by using the *Active Directory Administrative Center* or *Active Directory Users and Computers*. The only difference between these groups and the ones you create yourself is that you cannot delete the default groups.

Nesting Groups

As discussed previously, *group nesting* is the term used when groups are added as members of other groups. For example, when you make a global group a member of a universal group, it is nested within the universal group.

Group nesting reduces the number of times you need to assign permissions to users in different domains in a multidomain forest. For example, if you have multiple child domains in your AD DS hierarchy, and the users in each domain need access to an enterprise database application located in the parent domain, the simplest way to set up access to this application is as follows:

1. Create global groups in each domain that contain all users needing access to the enterprise database.
2. Create a universal group in the parent domain. Include each location's global group as a member.
3. Add the universal group to the required domain local group to assign the necessary permission to access and use the enterprise database.

This traditional approach to group nesting in AD DS is often referred to using the mnemonic *AGUDLP*: you add **A**ccounts to **G**lobal groups, add those global groups to **U**niversal groups, add universal groups to **D**omain **L**ocal groups, and, finally, assign **P**ermissions to the domain local groups.

For example, the Backup Operators group enables members to perform backups on the computers in the domain, because the Backup Operators group receives the *Backup files and directories* user right through the Default Domain Controllers Policy and Local Security Policy GPOs.

You can use the same method to create your own domain local groups, which you delegate administrative tasks and user rights for particular OUs. Then, after creating global groups (or universal groups, for forest-wide assignments) and adding them to the domain local groups, the structure is in place.

For example, if you run a single-domain enterprise and want to grant the manager of your Boston office the ability to back up the computers at that site, you could use a procedure like the following:

1. Create an OU for the branch office called *Boston*.
2. Create a domain local group called *Boston Backup*.
3. Create a GPO called *Boston OU Backup* and grant the Boston Backup group the *Backup files and directories* user right.
4. Link the GPO to the Boston OU.
5. Create a global group called *Boston Office Managers*.
6. Add the Boston Office Managers group to the Boston Backup group as a member.

All that remains is to add the user account for the Boston branch manager to the Boston Office Managers global group. If the manager goes on vacation or leaves the company, the administrator at the central office just has to add another user from the Boston office to the Boston Office Managers group in order to grant someone else the rights to perform backups.

This procedure grants one user the rights needed to perform one task at one site. Obviously, it is not efficient to delegate every task individually this way. A more efficient method is for enterprise administrators to organize their lists of tasks into groups, which they or others will assign to IT staffers. Microsoft refers to these groups as management roles and has published lists of recommended roles for service management and data management tasks, as shown in Table 15-1.

Table 15-1

Active Directory Management Roles

Service Management Roles	Data Management Roles
Forest Configuration Operators	Business Unit Administrators
Domain Configuration Operators	Account Administrators
Security Policy Administrators	Workstation Administrators
Service Administration Managers	Server Operators
Domain Controller Administrators	Resource Administrators
Backup Operators	Security Group Administrators
Schema Administrators	Help Desk Operators
Replication Management Administrators	Application-Specific Administrators
Replication Monitoring Operators	
DNS Administrators	

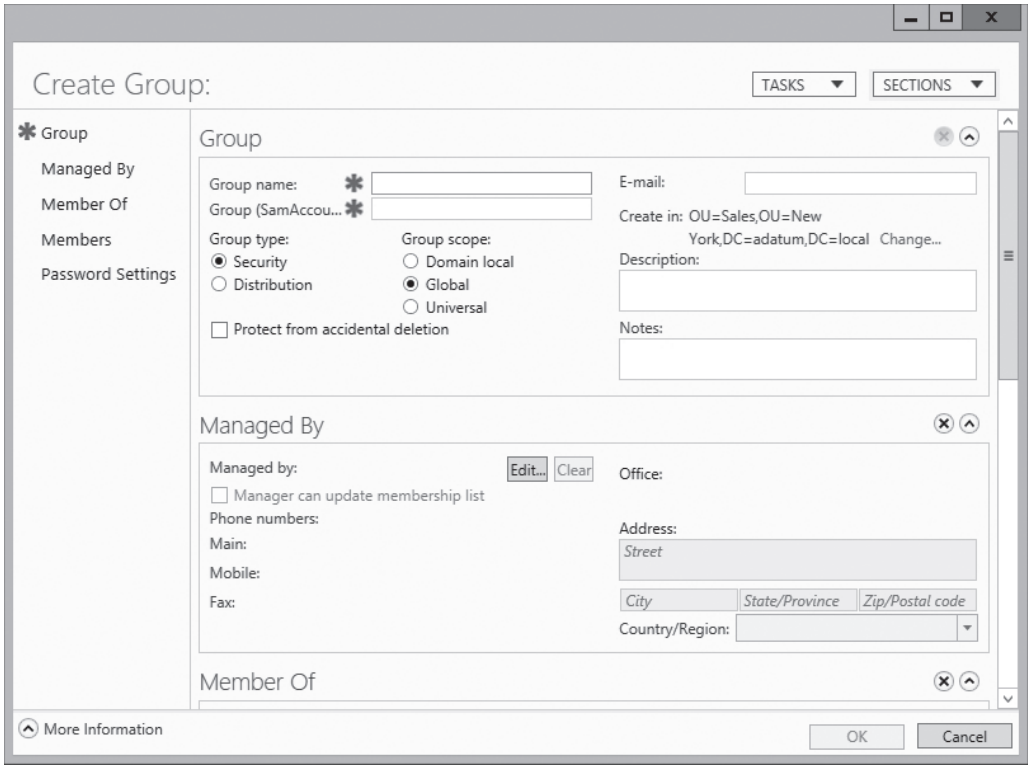
Creating Groups

The procedure for creating groups in Active Directory Administrative Center or Active Directory Users and Computers is similar to that of creating organizational units.

When you create a group, you must specify a name for the group object. The name you select can be up to 64 characters long and must be unique in the domain. You must also choose a group type and a group scope. Figure 15-4 shows the Create Group window in Active Directory Administrative Center.

Figure 15-4

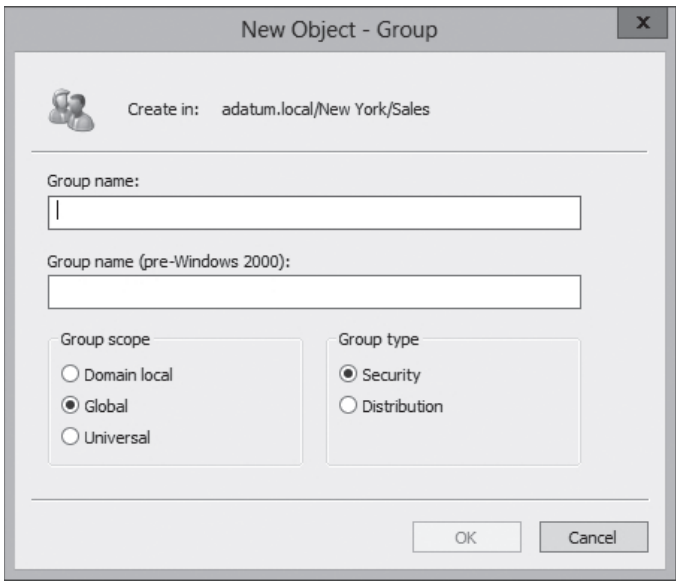
Creating a group in Active Directory Administrative Center



The New Object – Group dialog box in Active Directory Users and Computers is slightly different in appearance, but contains the same basic controls, as shown in Figure 15-5.

Figure 15-5

Creating a group in Active Directory Users and Computers



Although the graphical AD DS utilities are a convenient tool for creating and managing groups individually, they are not the most efficient method for creating large numbers of security principals. The command-line tools included with Windows Server 2012 R2 enable you to create and manage groups in large numbers by using batch files or other types of scripts, as discussed in Lesson 14, "Creating and Managing Active Directory Users and Computers." Some of these tools are discussed in the following sections.

Managing Group Memberships

Unlike the Active Directory Administrative Center, which enables you to specify a group's members as you create the group, in Active Directory Users and Computers, you must create the group object first, and then add members to it.

To add members to a group, you select it in the console and, from the *Action* menu, select *Properties* to open the group's Properties sheet, and then select the *Members* tab.

By using the *Members* tab, you can add objects to the group's membership list. On the *Member Of* tab, you can add the group to the membership list of another group. For both of these tasks, you use the standard *Select Users, Contacts, Computers, Service Accounts, or Groups* dialog box.

After you enter or find the objects you want to add, click *OK* to close the Properties sheet and add the objects to the group's membership list.

MANAGING GROUP MEMBERSHIP USING GROUP POLICY

It is also possible to control group memberships by using Group Policy. When you create Restricted Groups policies, you can specify the membership for a group and enforce it, so that no one can add or remove members.

To create Restricted Groups policies, use the following procedure.

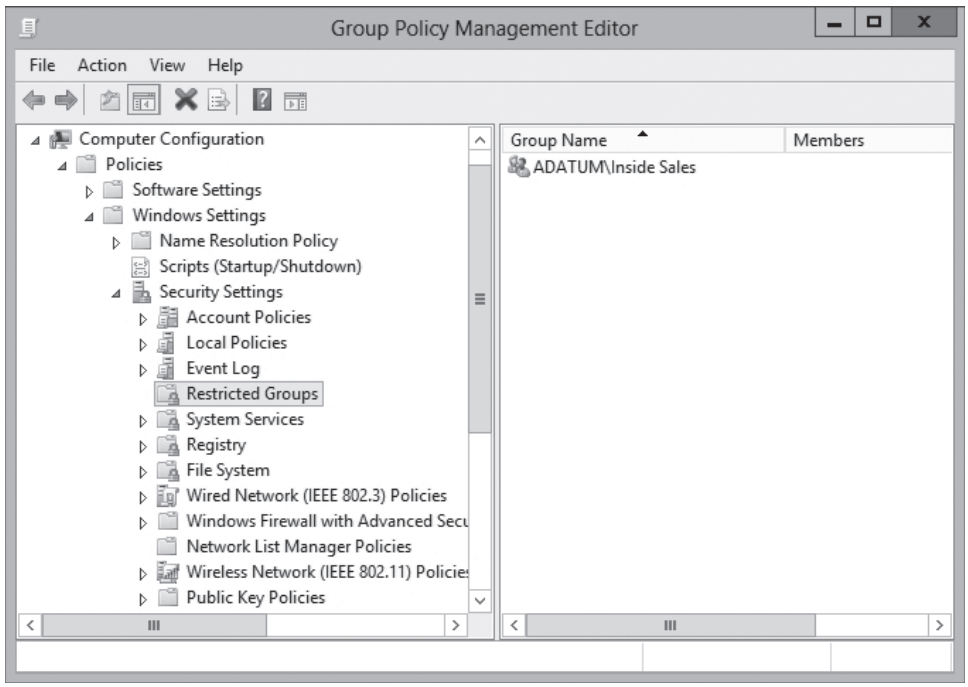
CREATE A RESTRICTED GROUPS POLICY

GET READY. Log on to the server running Windows Server 2012 R2 using an account with administrative privileges.

1. From the Tools menu in the *Server Manager* window, select Group Policy Management. The *Group Policy Management* console appears.
2. Create a new Group Policy object (GPO) and link it to your domain.
3. Open the GPO in the *Group Policy Management Editor* and browse to the Computer Configuration\Policies\Windows Settings\Security Settings\Restricted Groups folder, as shown in Figure 15-6.

Figure 15-6

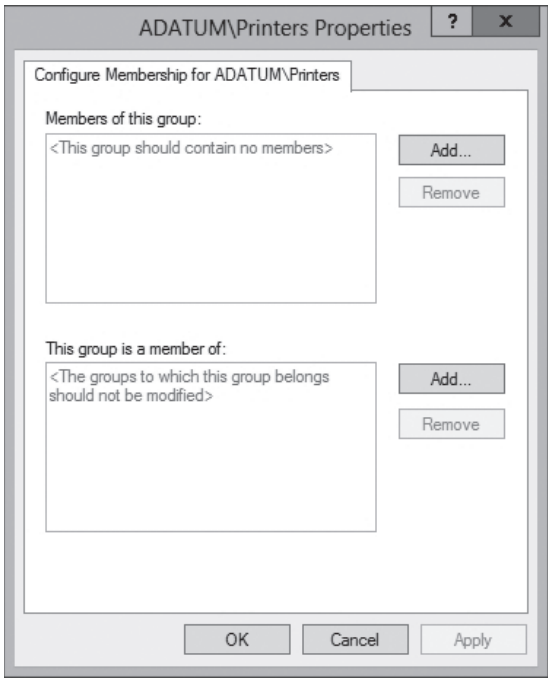
The Restricted Groups folder in the Group Policy object



4. Right-click the Restricted Groups folder and from the context menu, select Add Group. The *Add Group* dialog box appears.
5. Type or browse to add a group object and click OK. The group appears in the *Restricted Groups* folder and a Properties sheet for the policy appears, as shown in Figure 15-7.

Figure 15-7

The Properties sheet for a Restricted Groups policy



6. Click one or both of the Add buttons to add objects that should be members of the group, or other groups of which the group should be a member.
7. Click OK.

CLOSE the *Group Policy Management Editor* and *Group Policy Management* consoles.

The members you specify for a group in a Restricted Groups policy are the only members permitted to remain in that group. The policy does not prevent you from modifying the group membership by using other tools, but the next time the system refreshes its group policy settings, the group membership list will be overwritten by the policy.

Converting Groups

As group functions change, you might need to change a group object from one type to another.

For example, you might have created a distribution group that contains 100 members from multiple departments working on the same project for the purpose of sending e-mail messages. As the project progresses, members might need to access a common database. By converting the distribution group to a security group and assigning permissions to the group, you can provide the project members with access to the common database without having to create a new group and add 100 members to it again.

To change the type of a group, open the group's Properties sheet in the *Active Directory Administrative Center* or the *Active Directory Users and Computers* console. On the *General* tab, you can modify the *Group Type* option and click *OK*.

The process for changing the group's scope is the same, except that you select one of the *Group Scope* options on the *General* tab. The AD DS utilities enable you to perform only permissible scope changes. Table 15-2 lists the scope changes that are permitted.

Table 15-2

Active Directory Group Scope Conversion Restrictions

	To Domain Local	To Global	To Universal
From Domain Local	Not applicable	Not permitted	Permitted only when the domain local group does not have other domain local groups as members
From Global	Not permitted	Not applicable	Permitted only when the global group is not a member of another global group
From Universal	No restrictions	Permitted only when the universal group does not have other universal groups as members	Not applicable

■ Business Case Scenarios

Scenario 15-1: Administering Groups for Humongous Insurance

You are a network administrator for Humongous Insurance. Humongous Insurance has a multidomain forest. The forest root is humongousinsurance.com. There are also two child domains named west.humongousinsurance.com and east.humongousinsurance.com. The company has approximately 7,000 users, 7,000 client workstations, and 100 servers.

All domains are Windows Server 2013 domains. The forest root domain has 10 domain controllers. Five of those domain controllers are configured as DNS servers and two are configured as global catalog servers. The West domain has three domain controllers. Two of those domain controllers are configured as DNS servers. One of those domain controllers is configured as a global catalog server. The East domain has two Windows Server 2012 R2 domain controllers and three Windows 2008 domain controllers.

The forest root domain is located in College Station, Texas. The East domain is located in Gainesville, Florida. The West domain is located in San Diego, California. An Active Directory site is configured for each of these locations. The site for College Station is named Main_Site. The Gainesville site is named East_Site. The San Diego site is named West_Site.

You are one of several network administrators assigned to handle the forest root domain and College Station site. Your manager, Jean Trenary, has called a meeting of all network and desktop administrators. She wants to address several issues.

1. Jean says four internal auditors are in the forest root domain. Two internal auditors are in each of the child domains. Each set of internal auditors has been placed in a global group within each domain. These groups are named IA_Main, IA_East, and IA_West

after their respective locations. Jean wants all of the members of these groups to be able to access a common set of resources in the Main domain, while still segregating the auditors' ability to access other resources in domains other than their own. What is the recommended way to configure the groups to allow the desired functionality?

2. The network administrators from the West domain want to know why everyone always recommends placing global groups into universal groups, instead of placing the users directly into the universal groups. What should you tell them?
3. Jean approves a plan to hire assistants for each domain to create and manage user accounts. How can you give the assistants the immediate ability to help in this way, without making them domain administrators?
4. Two employees have been hired to back up data and manage printers for the Main_Site. Which built-in groups will give these users the permissions they require to manage the domain controllers? How should you set up their accounts and group memberships?

Scenario 15-2: Planning GPOs for Tailspin Toys

Tailspin Toys is running a single Windows Server 2012 R2 Active Directory domain with multiple OUs configured for each of its 12 locations. An administrator at each location is responsible for managing GPOs and user accounts. You are the enterprise administrator responsible for planning the infrastructure. For each of the following challenges, document your options and be prepared to share them with other students.

1. Administrators located at each location should be able to create new GPOs and edit any that they have created. They should not be able to change or delete GPOs that they have not created. What are your options for providing this functionality?
2. All users in each location are currently in one OU. Certain group policies should only apply to users in some departments and not others. What options should you consider that will allow group policies to be applied to only the necessary users?
3. Although you have created a domain-wide policy that enforces restrictions on administrative tools, you do not want those settings to apply to users for which you have delegated administrative permissions on each location's OU. What are your options to solve this?

Creating Group Policy Objects

LESSON 16

■ Introducing Group Policy

THE BOTTOM LINE

Group Policy is a mechanism for controlling and deploying operating system settings to computers all over your network. Group Policy consists of user and computer settings for the various Microsoft Windows operating systems, which the systems implement during computer startup and shutdown and user logon and logoff. You can configure one or more Group Policy objects (GPOs) and then use a process called ***linking*** to associate them with specific Active Directory Domain System (AD DS) objects. When you link a GPO to a container object, all the objects in that container receive the settings you configured in the GPO. You can link multiple GPOs to a single AD DS container or link one GPO to multiple containers throughout the AD DS hierarchy.

You can use Group Policy objects to manage any or all of the following types of settings:

- Registry-based policies, such as user desktop settings and environment variables, provide a consistent, secure, manageable working environment that addresses the users' needs and the organization's administrative goals. As the name implies, these settings modify the Windows Registry.
- Software installation policies ensure that users always have the latest versions of applications. If application files are inadvertently deleted, the policies can make repairs without user intervention.
- ***Folder redirection*** enables users to store local files on a network drive for backup, making them accessible from anywhere on the network.
- Offline file storage works with folder redirection to provide local file caching. This enables users to access their files even when the network is inaccessible.
- Scripts, including logon, logoff, startup, and shutdown commands, can assist in configuring the user environment.
- Windows Deployment Services (WDS) assists in rebuilding or deploying workstations quickly and efficiently in an enterprise environment.
- Microsoft Internet Explorer settings provide quick links and bookmarks for user accessibility, in addition to browser options, such as proxy use, acceptance of cookies, and caching options.
- Security settings protect resources on computers in the enterprise.

Depending on the organization's needs, you can choose which features and settings to implement. For example, you can create a policy for a public access computer in a library that configures the desktop environment with a proprietary library-access system.

In addition, you can disable the capability to write to the computer's hard drive. As you determine the needs of different users and address the needs within corporate security and computing policies, you can plan the best methods to implement Group Policy.

Group Policies can be linked to sites, domains, or organizational units (OUs) to apply the settings to all users and computers within the Active Directory containers. However, an advanced technique, called ***security filtering***, enables you to apply GPO settings to only one or more users or groups within a container by selectively granting the "Apply Group Policy" permission to one or more users or security groups.

Understanding Group Policy Objects

Group Policy objects (GPOs) contain the Group Policy settings to deploy to user and computer objects within a site, domain, or organizational unit. To deploy a GPO, you must associate it with the container to which it is deployed. This association links the GPO to the desired Active Directory Domain Services object. Administrative tasks for Group Policy include creating GPOs, specifying where they are stored, and managing the AD DS links.

There are three types of GPOs: ***local GPOs***, ***domain GPOs,*** and ***starter GPOs***.

LOCAL GPOS

All Windows operating systems have support for local Group Policy objects, sometimes known as LGPOs. Windows versions since Windows Server 2008 R2 and Windows Vista can support ***multiple local GPOs***. This enables you to specify a different local GPO for administrators or to create specific GPO settings for one or more local users configured on a workstation. This capability is particularly valuable for computers in public locations such as libraries and kiosks, which are not part of an Active Directory infrastructure.

Older Windows releases (prior to Windows Vista) can support only one local GPO, and the settings in that local GPO can apply only to the computer, not to individual users or groups.

Local GPO settings are stored on the local computer in the %systemroot%/System32/GroupPolicy folder.

A local GPO has the following characteristics:

- Local GPOs contain fewer options than domain GPOs. They do not support folder redirection or Group Policy software installation. Fewer security settings are available.
- When a local and a nonlocal (Active Directory-based) GPO have conflicting settings, the local GPO settings are overwritten by those of the nonlocal GPO.

DOMAIN GPOS

Nonlocal GPOs are created in Active Directory and are linked to sites, domains, or OUs. After linked to a container, the settings in the GPO are applied to all users and computers within the container by default. The content of each nonlocal GPO is stored in the following two locations:

- ***Group Policy container (GPC)*** is an Active Directory object that stores the properties of the GPO.
- ***Group Policy template (GPT)*** is located in the Policies subfolder of the SYSVOL share; the GPT is a folder that stores policy settings, such as security settings and script files.

STARTER GPOS

Starter GPOs is a feature introduced in Windows Server 2008. A starter GPO is a template for the creation of domain GPOs based on a standard collection of settings. When you create a new GPO from a starter GPO, all the policies in the starter are automatically copied to the new GPO as its default settings.

Configuring a Central Store

In Windows Server 2008 and Windows Vista, Microsoft replaced the token-based administrative template (ADM) files used with previous versions of Group Policy with an XML-based file format (***ADMX***). Administrative templates are the files defining the registry-based settings that appear in Group Policy objects.

Earlier Windows versions created a copy of the ADM files for each GPO you created and placed it in the SYSVOL volume of a domain controller. A large Active Directory installation could easily have dozens of GPOs, and each copy of the ADM files required 4 megabytes of storage. The result was a condition called ***SYSVOL bloat***, in which hundreds of megabytes of redundant information was stored on SYSVOL volumes, which had to be replicated to all the domain controllers for the domain.

To address this problem, Group Policy tools can now access the ADMX files from a ***Central Store***, a single copy of the ADMX files stored on domain controllers. To use a Central Store, however, you must create the appropriate folder in the SYSVOL volume on a domain controller.

By default, tools such as the Group Policy Management console save the ADMX files to the \%systemroot%\PolicyDefinitions folder, which on most computers is C:\Windows\PolicyDefinitions. To create a Central Store, you must copy the entire PolicyDefinitions folder to the same location as the Group Policy templates, that is, %systemroot%\SYSVOL\sysvol\<*domain name*>\Policies, or, in UNC notation, \\<*domain name*>\SYSVOL\<*domain name*>\Policies.

■ Using the Group Policy Management Console

THE BOTTOM LINE

The ***Group Policy Management console*** is the Microsoft Management Console (MMC) snap-in that you use to create Group Policy objects and manage their deployment to Active Directory Domain Services objects. The ***Group Policy Management Editor*** is a separate snap-in that opens GPOs and enables you to modify their settings.

There are several different ways of working with these two tools, depending on what you want to accomplish. You can create a GPO and then link it to a domain, site, or OU, or create and link a GPO in a single step. Windows Server 2012 R2 implements the tools as the Group Policy Management feature, and installs them automatically with the Active Directory Domain Services role. You can install the feature manually on a member server by using the Add Roles and Features Wizard in Server Manager. It is also included in the Remote Server Administration Tools package for Windows workstations.

Creating and Linking Nonlocal GPOs

If, as recommended previously, you leave the default GPOs unaltered, the first steps in deploying your own customized Group Policy settings are to create one or more new Group Policy objects and link them to appropriate AD DS objects.

To use the Group Policy Management console to create a new GPO and link it to an organizational unit object in AD DS, perform the following procedure.

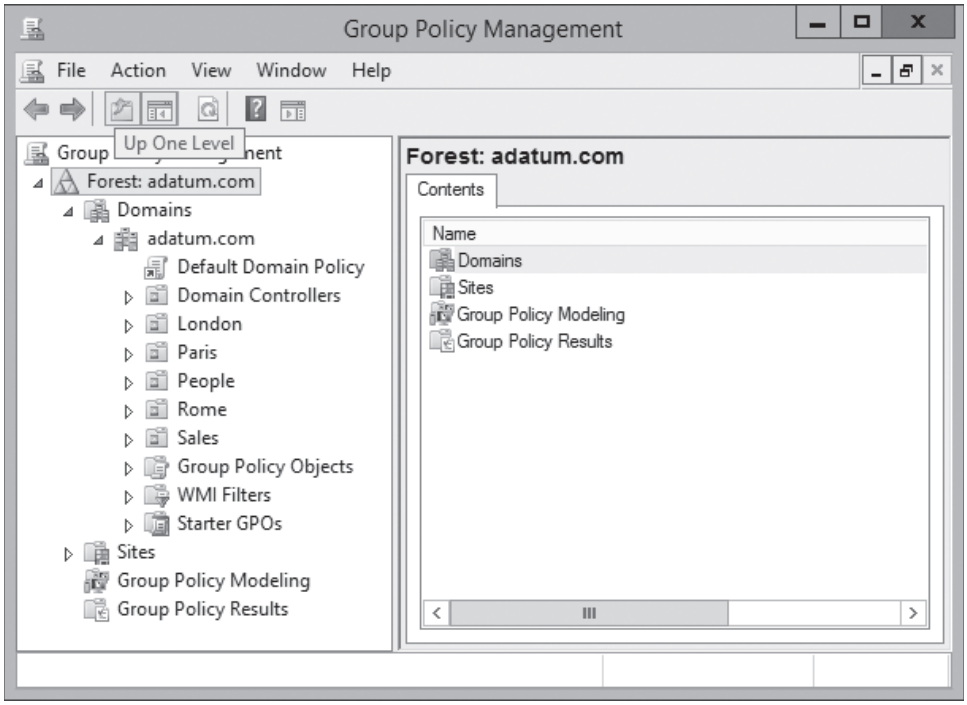
CREATE AND LINK A GPO TO AN OU

GET READY. Log on to a domain controller running Windows Server 2012 R2, using an account with domain Administrator privileges. The Server Manager console appears.

1. Open the Active Directory Administrative Center and create an OU called "Sales" in your domain.
2. From the Tools menu, select Group Policy Management. The Group Policy Management console appears, as shown in Figure 16-1.

Figure 16-1

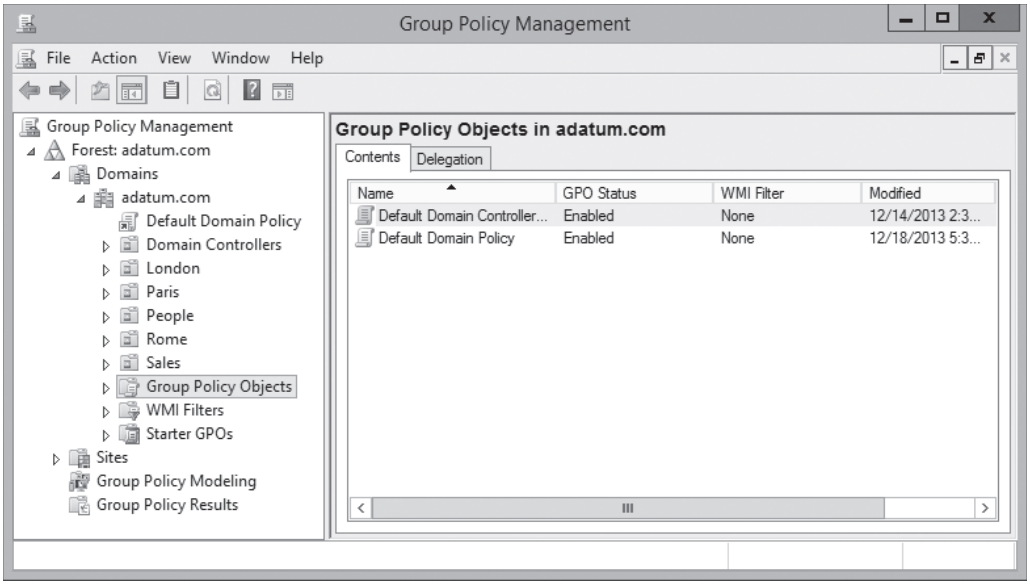
The Group Policy Management console



3. Expand the forest container and browse to your domain. Then expand the domain container and select the Group Policy Objects folder. The GPOs that currently exist in the domain appear in the *Contents* tab, as shown in Figure 16-2.

Figure 16-2

Contents of the Group Policy Objects folder



4. Right-click the Group Policy Objects folder and, from the context menu, select New. The *New GPO* dialog box appears.
5. In the Name text box, type a name for the new GPO and, if desired, select a *Source Starter GPO* from the drop-down list and click OK. The new GPO appears in the Contents list.
6. In the left pane, right-click the domain, site, or OU object to which you want to link the new GPO and, from the context menu, select Link an existing GPO. The *Select GPO* dialog box appears.
7. Select the GPO you want to link to the object and click OK. The GPO appears on the object's *Linked Group Policy Objects* tab.

CLOSE the Group Policy Management console.

You can also create and link a GPO to an AD container in a single step, by right-clicking an object and selecting Create a GPO in this Domain and Link it here from the context menu.

If you link a GPO to a domain object, it applies to all users and computers in the domain. On a larger scale, if you link a GPO to a site that contains multiple domains, the Group Policy settings are applied to all the domains and the child objects beneath them. This process is referred to as ***GPO inheritance***.

Using Security Filtering

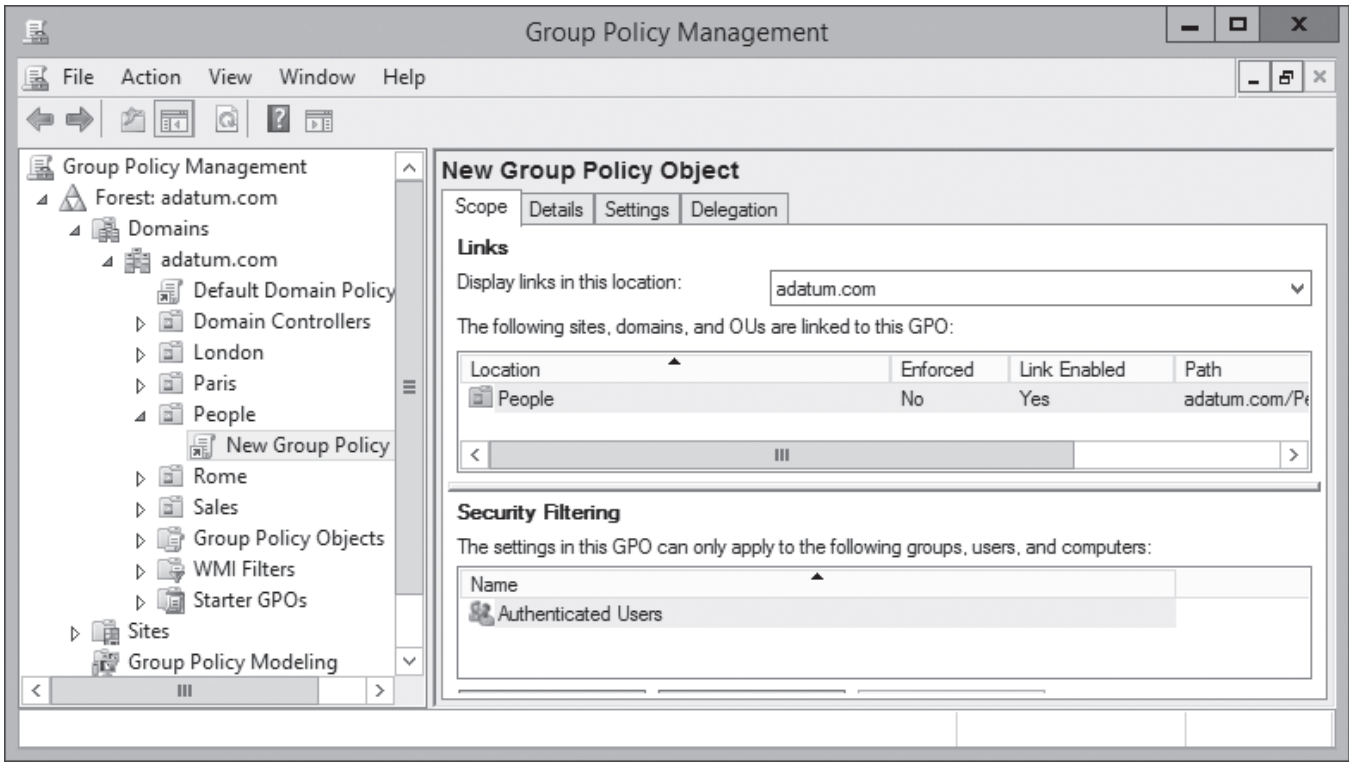
By default, linking a GPO to a container causes all the users and computers in that container to receive the GPO settings. The act of creating the link grants the Read and Apply Group Policy permissions for the GPO to the users and computers in the container.

In actuality, the system grants the permissions to the Authenticated Users special identity, which includes all the users and computers in the container. However, by using a technique called security filtering, you can modify the default permission assignments so that only certain users and computers receive the permissions and, consequently, the settings in the GPO.

To modify the default security filtering configuration for a GPO, select it in the left pane of the Group Policy Management console, as shown in Figure 16-3 . In the Security Filtering area, you can use the Add and Remove buttons to replace the Authenticated Users special identity with specific user, computer, or group objects. Of the users and computers in the container to which the GPO is linked, only those you select in the Security Filtering pane receive the settings from the GPO.

Figure 16-3

Security filtering in the Group Policy Management console



Managing Starter GPOs

Starter GPOs are templates that create multiple GPOs with the same set of baseline Administrative Templates settings.

You create and edit starter GPOs just as you would any other Group Policy object. In the Group Policy Management console, you right-click the Starter GPOs folder and, from the context menu, select New to create a blank starter GPO. You can then open the starter GPO in the Group Policy Management Editor and configure any settings you want to carry over to the new GPOs you create.

After you create and edit your starter GPOs, you can create new GPOs from them in two ways. You can right-click a starter GPO and select New GPO from Starter GPO from the context menu, or you can create a new GPO in the usual manner described previously and select the starter GPO in the Source Starter GPO drop-down list. This process copies the settings from the starter GPO to the new GPO, which you can continue to edit from there.

Configuring Group Policy Settings

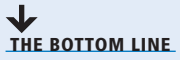
Group Policy settings enable you to customize the configuration of a user's desktop, environment, and security settings. The settings are divided into two subcategories: Computer Configuration and User Configuration. The subcategories are referred to as Group Policy nodes. A node is a parent structure that holds all related settings. In this case, the node is specific to computer configurations and user configurations.

Group Policy nodes provide a way to organize the settings according to where they are applied. The settings you define in a GPO can be applied to client computers, users, or member servers and domain controllers. The application of the settings depends on the container to which you link the GPO. By default, all objects within the container to which you link the GPO are affected by the GPO's settings.

The Computer Configuration and the User Configuration nodes contain three subnodes, or extensions, that further organize the available Group Policy settings. Within the Computer Configuration and User Configuration nodes, the subnodes are as follows:

- **Software Settings:** The Software Settings folder located under the Computer Configuration node contains Software Installation settings that apply to all users who log on to a domain using a specific computer. These settings are applied before any user is allowed to log on. Rather than being computer specific, the Software Settings folder located under the User Configuration node contains Software Installation settings applied to users designated by the Group Policy, regardless of the computer from which they log on.
- **Windows Settings:** The Windows Settings folder located under the Computer Configuration node in the Group Policy Management Editor contains security settings and scripts that apply to all users who log on to Active Directory Domain Services from that specific computer. This means that the settings are computer specific. The Windows Settings folder located under the User Configuration node contains settings related to folder redirection, security settings, and scripts that apply to specific users. The computer from which a user logs on does not affect these policy settings; the policies are applied regardless of the user's log on location.
- **Administrative Templates:** Windows Server 2012 R2 includes thousands of Administrative Template policies, which contain all registry-based policy settings. Administrative Templates are files with the .admx extension. They are used to generate the user interface for the Group Policy settings that you can set by using the Group Policy Management Editor. The Windows Server 2012 R2 .admx files are based on the eXtensible Markup Language (XML), unlike the Windows 2003 .adm files, which are token-based text files.

■ Creating Multiple Local GPOs



Computers that are members of an AD DS domain benefit from flexibility when it comes to Group Policy configuration. Standalone (non-AD DS) systems can achieve some of the flexibility, as long as they are running at least Windows Vista or Windows Server 2008 R2. These operating systems enable you to create multiple local GPOs that provide different settings for users, based on their identities.

Windows systems supporting multiple local GPOs have the following three layers of Group Policy support:

- **Local Group Policy:** Identical to the single local GPO supported by older operating system versions, the Local Group Policy layer consists of both Computer and User

settings and applies to all system users, administrative or not. Because this is the only local GPO that includes computer settings, you must use this GPO to apply Computer Configuration policies.

- **Administrators and Non-administrators Group Policy:** This layer consists of two GPOs, one of which applies to members of the local Administrators group and one that applies to all users that are not members of the local Administrators group. This enables you to easily create user settings that distinguish between administrative and non-administrative users. Unlike the Local Group Policy GPO, this layer does not include computer settings.
- **User-specific Group Policy:** This layer consists of GPOs that apply to specific local user accounts created on the computer. These GPOs can apply to individual users only, not to local groups. These GPOs also do not have computer configuration settings.

Windows applies the local GPOs in the order listed here. The Local Group Policy settings are applied first, then either the Administrators or Non-administrators GPO, and finally any user specific GPOs. As with nonlocal GPOs, the settings processed later can overwrite any earlier settings with which they conflict.

In the case of a system that is also a member of a domain, the three layers of local GPO processing come first and are followed by the standard order of nonlocal Group Policy application.

To create local GPOs, you use the Group Policy Object Editor, which is an MMC snap-in provided on all Windows computers, as in the following procedure.

CREATE LOCAL GPOS

GET READY. Log on to a Windows computer, using an account with Administrator privileges. The Server Manager console appears.

1. Open the Run dialog box and, in the Open text box, type mmc and click OK. An empty MMC console appears.
2. Click File > Add/Remove Snap-in. The *Add or Remove Snap-ins* dialog box appears.
3. Select Group Policy Object Editor from the *Available snap-ins* list and click Add. The *Select Group Policy Object* page appears.
4. To create the Local Group Policy GPO, click Finish. To create a secondary or t*ertiary GPO,* click Browse. The *Browse for a Group Policy Object* dialog box appears.
5. Click the Users tab, as shown in Figure 16-4.

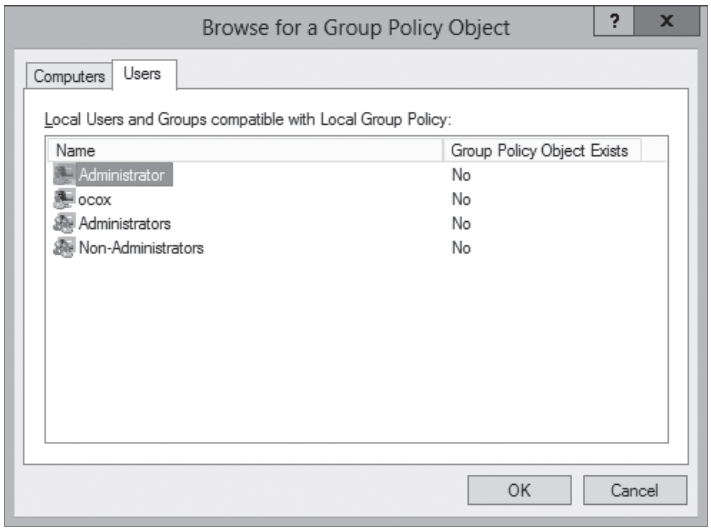


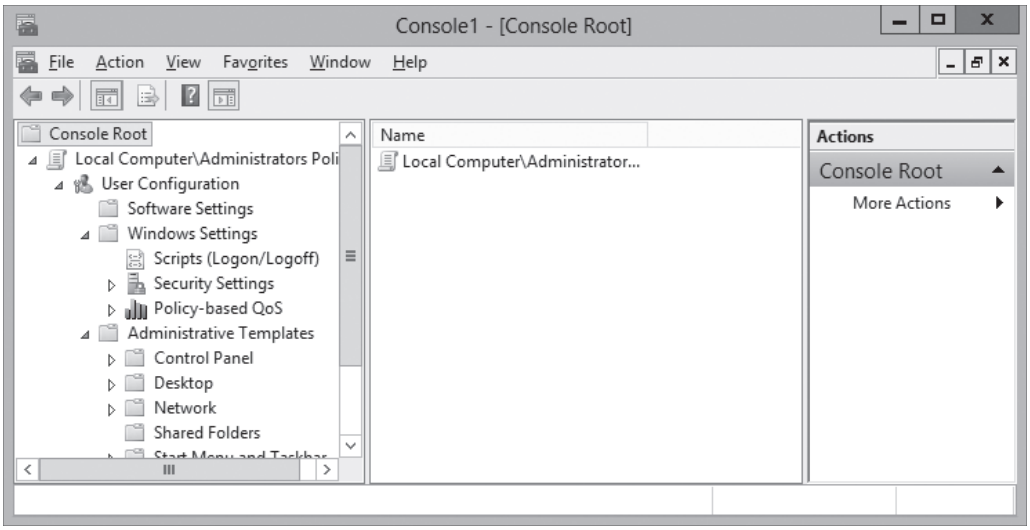
Figure 16-4

The Users tab of the *Browse for a Group Policy Object* dialog box

6. To create a secondary GPO, select either Administrators or Non-Administrators and click OK. To create tertiary GPO, select a user and click OK. The Group Policy object appears on the *Select Group Policy Object* page.
7. Click Finish. The snap-in appears in the *Add or Remove Snap-ins* dialog box.
8. Click OK. The snap-in appears in the MMC console, as shown in Figure 16-5.

Figure 16-5

A Group Policy Object Editor console



9. Click File > Save As. A *Save As* combo box appears.
10. Type a name for the console, to save it in the Administrative Tools program group.

CLOSE the MMC console.

You can now open this console to configure the settings in the GPO you created. A Local Group Policy GPO has both Computer Configuration and User Configuration settings, whereas the secondary and tertiary GPOs have only User Configuration settings.

■ Business Case Scenarios

Scenario 16-1: Creating Device Restrictions

After a recent incident in which an employee left the company with a substantial amount of confidential data, the IT director has given Alice the task of implementing Group Policy settings that prevent all users except administrators and members of the Executives group from installing any USB devices. Alice creates a GPO called Device Restrictions for this purpose and links it to the company's single domain object. The GPO contains the following settings:

- Allow administrators to override Device Installation Restriction policies—Enabled
- Prevent installation of devices not described by other policy sessions—Enabled

What else must Alice do to satisfy the requirements of her assignment?

Scenario 16-2: Deploying a GPO

Ralph has a number of Group Policy settings that he must deploy to the workstations of his firm's department managers, so they can run a timesheet application for their hourly employees. He has created a GPO containing those settings, and has named it HourlyTime. After linking the GPO to his company domain object in Active Directory Domain Services, Ralph quickly receives a number of trouble tickets referred to him from the help desk. The salaried employees are complaining that the application they use to file their weekly expenses has stopped working. Ralph's testing eventually establishes that it is the settings in the HourlyTime GPO that are causing the expense voucher application to malfunction. How can Ralph deploy the HourlyTiime GPO to the department managers only, without interfering with the other application?

Configuring Security Policies

LESSON 17

■ Configuring Security Policies Using Group Policy

THE BOTTOM LINE

In Lesson 16, "Creating Group Policy Objects," you learned how to create and deploy Group Policy objects (GPOs) by linking them to Active Directory Domain Services (AD DS) objects. This lesson focuses on configuring the settings in the Group Policy objects themselves, and particularly on the ones that can help secure your network.

It is important for administrators to know the difference between user and computer settings. In addition to learning about these settings and categorizing them based on where you apply them, this lesson also looks at the default Group Policy refresh process and how to invoke a manual refresh of Group Policy objects when necessary.

One of the primary aims of Group Policy is to provide centralized management of security settings for users and computers. Most of the settings that pertain to security are found in the Windows Settings folder within the Computer Configuration node of a GPO. You can use security settings to govern how users are authenticated to the network, the resources they are permitted to use, group membership policies, and events related to user and group actions recorded in the event logs. Table 17-1 briefly describes some of the security settings that you can configure within the Computer Configuration node.

Table 17-1

Computer Configuration Node Security Settings

Setting	Description
Account Policies	Includes settings for Password Policy, Account Lockout Policy, and Kerberos Policy. A domain-wide policy, such as the Default Domain Policy GPO, also includes Kerberos Policy settings. Prior to Windows Server 2008, you could configure only Password Policy and Account Lockout Policy settings at the domain level. Starting with Windows Server 2008, you can configure Fine-Grained Password Policies that enable you to specify multiple password policies in a single domain.
Local Policies	Contains three subcategories that pertain to the local computer policies: Audit Policy, User Rights Assignment, and Security Options.
Event Log Policy	These settings pertain to Event Viewer logs, their maximum size, retention settings, and accessibility.
Restricted Groups Policy	This setting gives you control over the Members property and the Members Of property for specific security groups.

(continued)

Table 17-1

(continued)

Setting	Description
System Services Policy	These settings can be used to define the startup mode and access permissions for all system services. You can configure each service to be disabled, to start automatically, or to start manually.
Registry and File System Policies	These settings configure access permissions and audit settings for specific registry keys or file system objects.
Wired Network (IEEE 802.3) Policies	Enables you to create policies specifying authentication settings for computers on wired networks running Windows Vista or later.
Windows Firewall with Advanced Security	Enables you to create inbound and outbound firewall filters and distribute them to network computers.
Network List Manager Policies	Specifies whether unidentified networks should be designated as public or private, causing them to receive a specific group of firewall rules.
Wireless Network (IEEE 802.11) Policies	Enables the creation of policies for IEEE 802.11 wireless networks. Settings include preferred networks and authentication types, in addition to other security-related options.
Public Key Policies	This node includes options to create an Encrypted File System (EFS), automatic certificate request, trusted root certificates, and an enterprise trust list.
Software Restriction Policies	This policy can specify software that you want to run on computers, and that you want to prevent from running, because it might pose a security risk to the computer or organization.
Network Access Protection	Enables you to create policies to configure Network Access Protection clients.
Application Control Policies	Configures access control policies for applications using AppLocker.
IP Security Policies on Active Directory	Includes policy settings that enable you to define mandatory rules applicable to computers on an IP-based network.
Advanced Audit Policy Configuration	Provides more granular audit policy settings than the Audit Policy node found under Local Policies.

Policy settings in the Computer Configuration node apply to a computer; it does not matter who logs on to it. You can apply more Computer Configuration security settings than settings to a specific user. Table 17-2 describes the security settings that you can apply within the User Configuration node of a Group Policy object.

Table 17-2

Security Settings Applied in the User Configuration Node

Setting	Description
Public Key Policies	Includes the Enterprise Trust policy that enables you to list the trusted sources for certificates. Also, auto-enrollment settings can be specified for the user within this node.
Software Restriction Policies	This policy can be used to specify software that you want to run for the user. Specifically, it can be used to disallow applications that might pose a security risk if run.

Defining Local Policies

Local Policies enable you to set user privileges on the local computer that govern what users can do on the computer and determine whether the system should track them in an event log. Tracking events that take place on the local computer, a process referred to as ***auditing***, is another important part of monitoring and managing activities on a computer running Windows Server 2012 R2.

The Local Policies node of a GPO has three subordinate nodes: User Rights Assignment, Security Options, and Audit Policy. As discussed in the following sections, keep in mind that Local Policies are local to a computer. When they are part of a GPO in Active Directory, they affect the local security settings of computer accounts to which the GPO is applied.

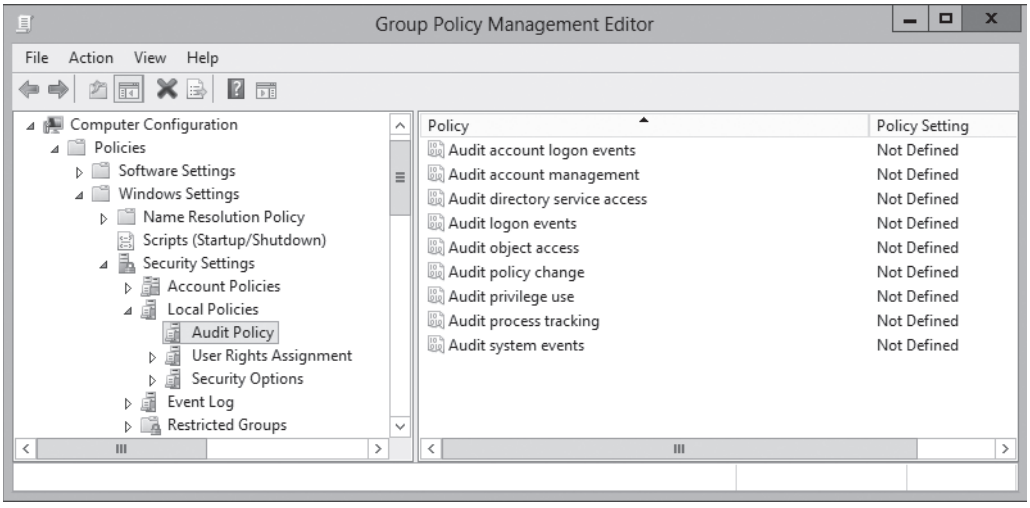
PLANNING AND CONFIGURING AN AUDIT POLICY

The Audit Policy section of a GPO enables you to log successful and failed security events, such as logon events, account access, and object access. You can use auditing to track both user activities and system activities. Planning to audit requires that you determine the computers to be audited and the types of events you want to track.

When you consider an event to audit, such as account logon events, you must decide whether to audit successful logon attempts, failed logon attempts, or both. Tracking successful events enables you to determine how often users access network resources. This information can be valuable when planning your resource usage and budgeting for new resources. Tracking failed events can help you determine when security breaches occur or are attempted. For example, if you notice frequent failed logon attempts for a specific user account, you might want to investigate further. The policy settings available for auditing are shown in Figure 17-1.

Figure 17-1

Audit Policies in the Default Domain Policy



When an audited event occurs, Windows Server 2012 R2 writes an event to the security log on the domain controller or the computer where the event took place. If it is a logon attempt or other Active Directory–related event, the event is written to the domain controller. If it is a computer event, such as a drive access, the event is written to the local computer's event log.

Implementation of your plan requires that you specify the categories to be audited and, if necessary, configure objects for auditing. To configure an audit policy, use the following procedure.

CONFIGURE AN AUDIT POLICY

GET READY. Log on to a domain controller running Windows Server 2012 R2, using an account with domain Administrator privileges. The Server Manager console appears.

1. From the Tools menu, select Group Policy Management. The Group Policy Management console appears.
2. Expand the forest container and browse to your domain. Then expand the domain container and select the Group Policy Objects folder. The GPOs that currently exist in the domain appear in the *Contents* tab.
3. Right-click the Default Domain Policy GPO and click Edit. A Group Policy Management Editor window for this policy appears.
4. Browse to the Computer Configuration\Policies\Windows Settings\Security Settings\Local Policies node and select Audit Policy. The audit policy settings appear in the right pane.
5. Double-click the Audit Policy setting you want to modify. The Properties sheet for the policy you chose appears.
6. Select the Define This Policy Setting check box.
7. Select the appropriate check box(es) to audit Success, Failure, or both.
8. Click OK to close the setting's Properties sheet.

CLOSE the Group Policy Management Editor and the Group Policy Management console.

You now configured an audit policy in the Default Domain Policy GPO, which will be propagated to all the computers in the domain during the next policy refresh.

Configuring objects for auditing is necessary when you configure either of the two following event categories:

- **Audit Directory Service Access:** logs user access to Active Directory objects, such as other user objects or OUs.
- **Audit Object Access:** logs user access to files, folders, registry keys, and printers.

Each of these event categories requires additional setup steps, which are described in the following procedure.

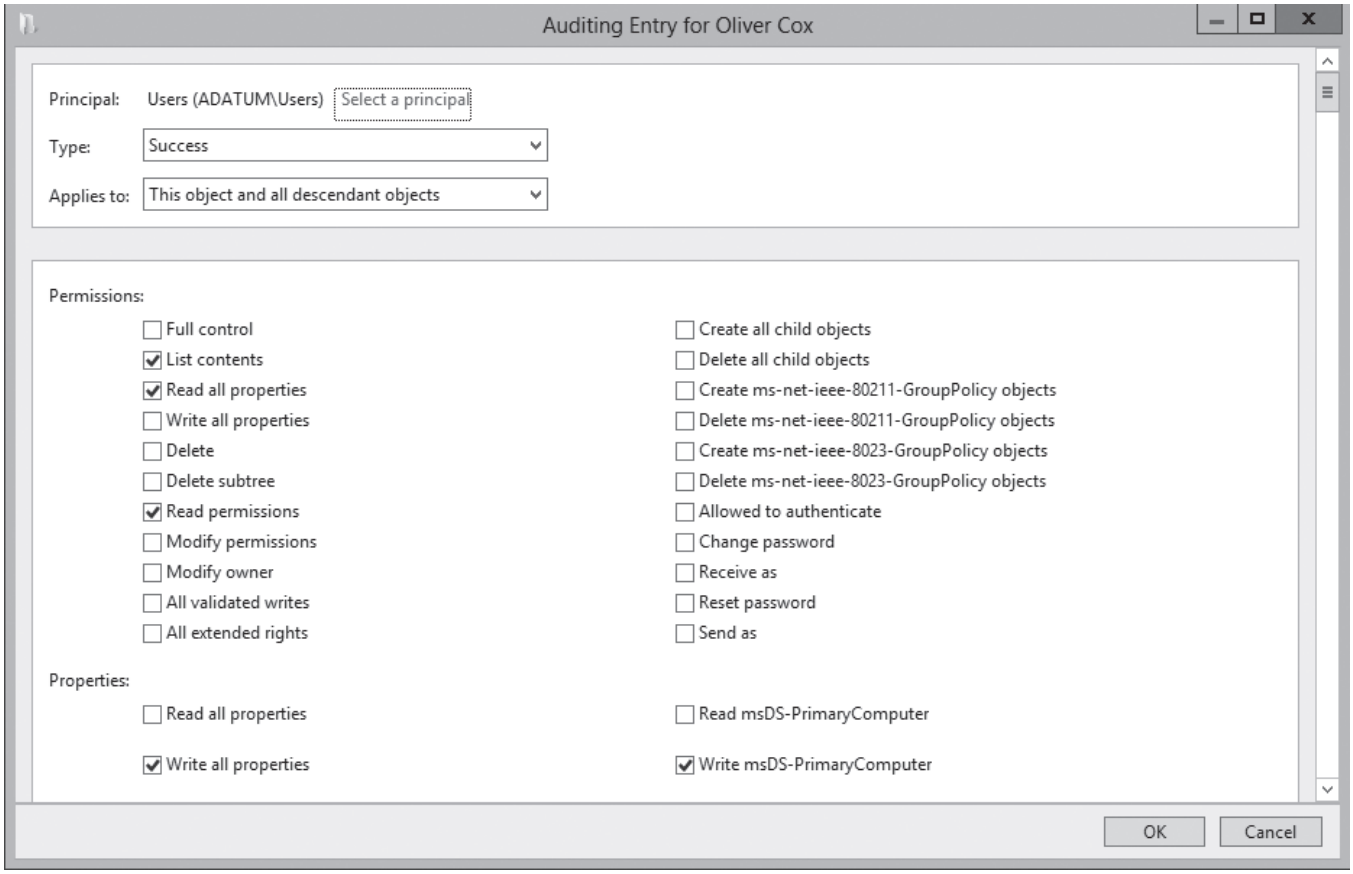
CONFIGURE AN ACTIVE DIRECTORY OBJECT FOR AUDITING

GET READY. Log on to a domain controller running Windows Server 2012 R2, using an account with domain Administrator privileges. The Server Manager console appears.

1. From the Tools menu, select Active Directory Users and Computers. The *Active Directory Users and Computers* console appears.
2. From the View menu, select Advanced Features.
3. Browse to the object that you want to audit. Right-click the object and, from the context menu, select Properties. The object's Properties sheet appears.
4. Click the Security tab, and then click Advanced. The *Advanced Security Settings* dialog box for the object appears.
5. Select the Auditing tab.
6. Click Add. The *Auditing Entry* page for the object appears.
7. Click Select a Principal. The *Select User, Computer, Service Account, or Group* dialog box appears. Select the users or groups to be audited for Active Directory object access and click OK. The users or groups appear in the *Auditing Entry* dialog box for the object, as shown in Figure 17-2.

Figure 17-2

The Auditing Entry dialog box for an object



8. From the *Type* drop-down list, specify whether you want to audit failures, successes, or both.
9. From the *Applies to* drop-down list, specify which descendent objects should be audited.
10. Select the Permissions and/or Properties you want to audit for this object and click OK. The new Auditing entry appears in the *Advanced Security Settings* dialog box.
11. Create additional auditing entries, if desired, and click OK.
12. Click OK to close the object's Advanced Security Settings dialog box.
13. Click OK to close the object's Properties sheet.

CLOSE the *Active Directory Users and Computers* console.

After you configure the auditing policy and the AD DS objects you want to audit, the systems monitor the objects and create entries in the Security log.

To audit access to objects, such as files and folders, use the following procedure.

CONFIGURE FILES AND FOLDERS FOR AUDITING

GET READY. Log on to a domain controller running Windows Server 2012 R2, using an account with domain Administrator privileges. The Server Manager console appears.

1. Open File Explorer, right-click the file or folder you want to audit and, from the context menu, select Properties. The Properties sheet for the file or folder appears.

2. Click the Security tab, and then click Advanced. The *Advanced Security Settings* dialog box appears.
3. Click the Auditing tab.
4. Click Add. The *Auditing Entry* page appears.
5. Click Select a Principal. The *Select User, Computer, Service Account, or Group* dialog box appears. Select the users or groups to be audited for Active Directory object access and click OK. The users or groups appear in the *Auditing Entry* dialog box for the object.
6. From the *Type* drop-down list, specify whether you want to audit failures, successes, or both.
7. From the *Applies to* drop-down list, specify which descendent objects should be audited.
8. Select the basic permissions you want to audit for this object and click OK. The new Auditing entry appears in the *Advanced Security Settings* dialog box.
9. Create additional auditing entries, if desired, and click OK.
10. Click OK to close the object's *Advanced Security Settings* dialog box.
11. Click OK to close the object's Properties sheet.

CLOSE the File Explorer window.

You now configured auditing for files and folders within the Windows operating system.

ASSIGNING USER RIGHTS

The User Rights Assignment settings in Windows Server 2012 R2 are extensive and include settings that pertain to rights users need to perform system-related tasks, as shown in Figure 17-3.

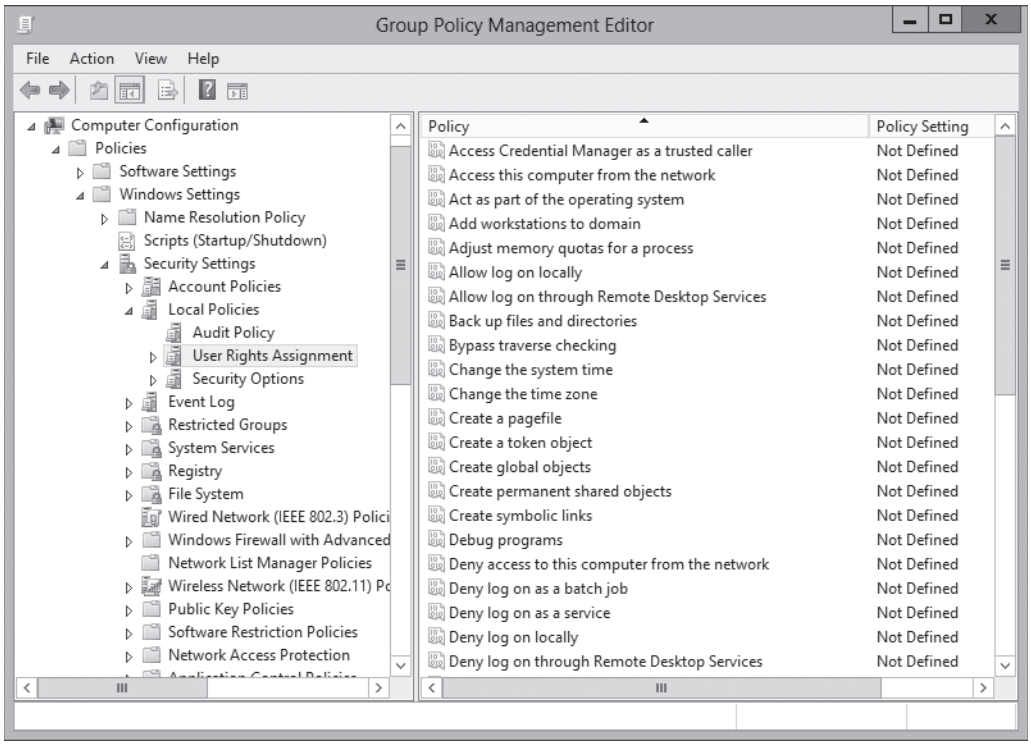


Figure 17-3

User rights assignment settings in a Group Policy object

For example, a user logging on locally to a domain controller must have the Allow Log On Locally right assigned to his or her account or be a member of one of the following AD DS groups:

- Account Operators
- Administrators
- Backup Operators
- Print Operators
- Server Operators

These group memberships enable users to log on locally because Windows Server 2012 R2 assigns the Allow Log On Locally user right to those groups in the Default Domain Controllers Policy GPO by default.

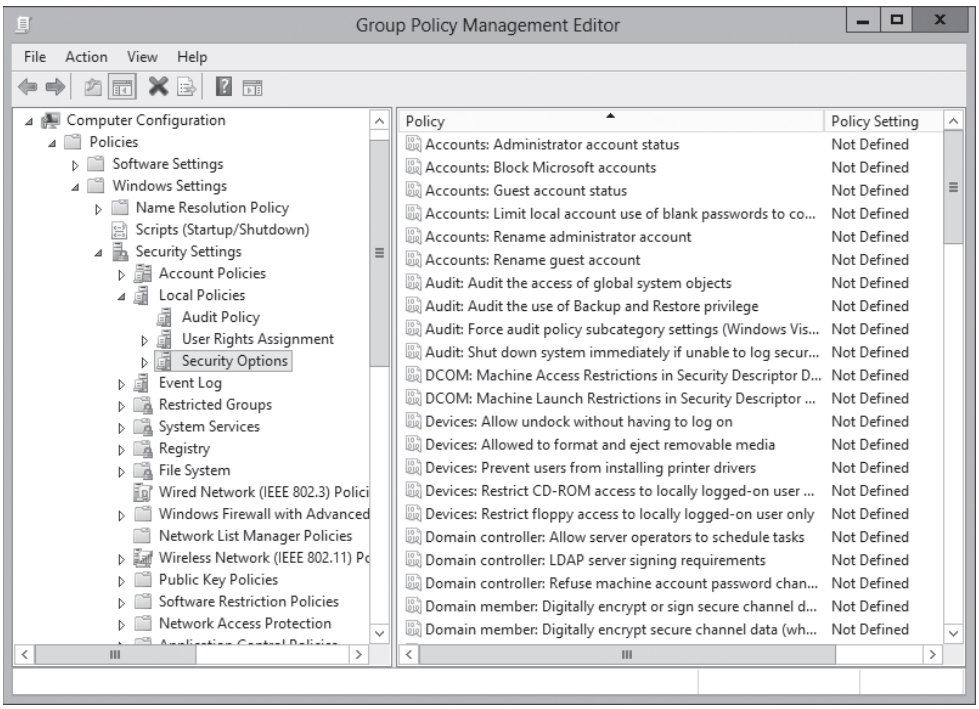
Other similar settings included in this collection are related to user rights associated with system shutdown, taking ownership privileges of files or objects, restoring files and directories, and synchronizing directory service data.

CONFIGURING SECURITY OPTIONS

The Security Options node in a GPO, shown in Figure 17-4, includes security settings related to interactive log on, digital signing of data, restrictions for access to floppy and CD-ROM drives, unsigned driver installation behavior, and logon dialog box behavior.

Figure 17-4

The Security Options node in a GPO



The Security Options category also includes options to configure authentication and communication security within Active Directory through the use of the following settings:

- **Domain controller:** LDAP server signing requirements controls whether LDAP traffic between domain controllers and clients must be signed. This setting can be configured with a value of None or Require signing.
- **Domain member:** Digitally sign or encrypt or sign secure channel data (always) controls whether traffic between domain members and the domain controllers are signed and encrypted at all times.

- **Domain member:** Digitally encrypt secure channel data (when client agrees) indicates that traffic between domain members and the domain controllers is encrypted only if the client workstations are able to do so.
- **Domain member:** Digitally sign secure channel data (when client agrees) indicates that traffic between domain members and the domain controllers is signed only if the client services are able to do so.
- **Microsoft network client:** Digitally sign communications (always) indicates that Server Message Block (SMB) signing is enabled by the SMB signing component of the SMB client at all times.
- **Microsoft network client:** Digitally sign communications (if server agrees) indicates that SMB signing is enabled by the SMB signing component of the SMB client only if the corresponding server service is able to do so.
- **Microsoft network server:** Digitally sign communications (always) indicates that SMB signing is enabled by the SMB signing component of the SMB server at all times.
- **Microsoft network server:** Digitally sign communications (if server agrees) indicates that SMB signing is enabled by the SMB signing component of the SMB server only if the corresponding client service is able to do so.

From this section, you can also enforce the level of NT LAN Manager (NTLM) authentication that is allowed on your network. Although Kerberos is the default authentication protocol in an AD DS network, NTLM authentication is used in certain situations. The original incarnation of NTLM authentication was called LAN Manager (LM) authentication, which is now considered a weak authentication protocol that can easily be decoded by network traffic analyzers. Microsoft has improved NTLM authentication over the years by introducing first NTLM and subsequently NTLMv2.

NTLM authentication levels are controlled by the *Network security: LAN Manager authentication level* security setting, which enables you to select one of the following options:

- Send LM and NTLM responses.
- Send LM and NTLM—use NTLMv2 session security if negotiated.
- Send NTLM response only.
- Send NTLMv2 response only.
- Send NTLMv2 response only. Refuse LM.
- Send NTLMv2 response only. Refuse LM and NTLM.

By allowing only the most stringent levels of NTLM authentication on your network, you can improve the overall communications security of Active Directory.

Using Security Templates

You learned previously how to deploy security and other system configuration settings on a Microsoft Windows network using Group Policy. Windows Server 2012 R2 also includes another mechanism for deploying security configuration settings called security templates.

A ***security template*** is a collection of configuration settings stored as a text file with an .inf extension. Security templates can contain many of the same security parameters as group policy objects. However, security templates present these parameters in a unified interface, enable you to save your configurations as files, and simplify the process of deploying them when and where they are needed.

The settings that you can deploy using security templates include many of the security policies covered in this lesson, including audit policies, user rights assignments, security options, event log policies, restricted groups, and others. By itself, a security template is a convenient way to configure the security of a single system. When you combine it with group policies or scripting, security templates enable you to maintain the security of networks consisting of hundreds or thousands of computers running various Microsoft Windows versions.

Using these tools together, you can create complex security configurations, and mix and match the configurations for each of the various roles computers serve in your organization. When deployed across a network, security templates enable you to implement consistent, scalable, and reproducible security settings throughout the enterprise.

USING THE SECURITY TEMPLATES CONSOLE

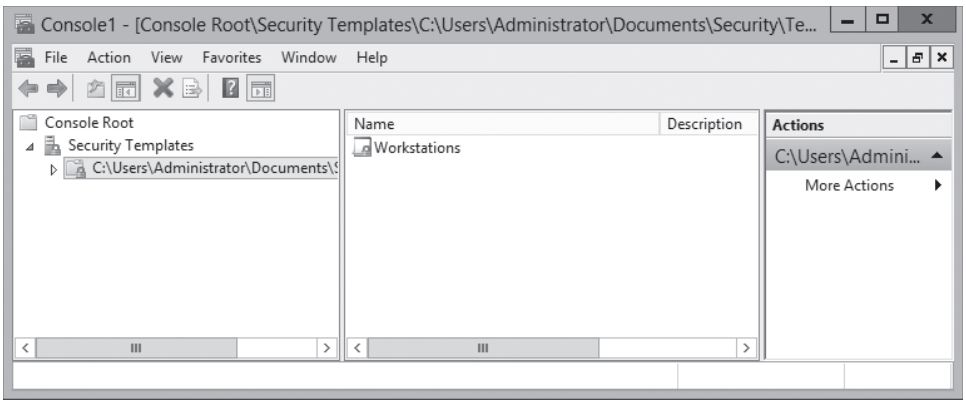
Security templates are plain text files that contain security settings in a variety of formats, depending on the nature of the individual settings. For example, many of the security policies are implemented by registry settings, and for these, a template file contains entries such as the following:

```
MACHINE\Software\Microsoft\Windows
NT\CurrentVersion\Winlogon\PasswordExpiryWarning=4,14
```

Although it is possible to work with security template files directly, by using any text editor, Windows Server 2012 R2 provides a graphical interface that makes the job much easier. To create and manage security templates, you use the Security Templates snap-in for Microsoft Management Console. By default, the Windows Server 2012 R2 Administrative Tools menu does not include an MMC console containing the Security Templates snap-in, so you need to create one by using the MMC Add or Remove Snap-ins dialog box. After you create a new template, the console provides an interface like the one shown in Figure 17-5.

Figure 17-5

The Security Templates snap-in



The left pane of the Security Templates snap-in points to a default folder in which the console stores the template files you create by default. The snap-in interprets any file in this folder with an .inf extension as a security template, even though the extensions do not appear in the console.

When you create a new template in the console, you see a hierarchical display of the policies in the template, as well as their current settings. Many of the policies are identical to those in a GPO, both in appearance and function. You can modify the policies in each template just as you would those in a GPO.

PLANNING A SECURITY TEMPLATE STRATEGY

When planning a security template strategy, think in terms of computer roles, rather than individual computers. It is possible to create a separate template for each computer and customize the settings for that particular computer's needs, but that defeats the purpose of

creating templates in the first place, because it is just as easy to configure each computer manually. By creating templates for specific roles, you can apply them to multiple computers, using combinations in cases where computers perform multiple roles.

CREATING SECURITY TEMPLATES

After the plan for a network's security templates is in place, you can proceed to create the templates. To create a new security template from scratch, use the following procedure.

CREATE A SECURITY TEMPLATE

GET READY. Log on to a Windows computer, using an account with Administrator privileges. The Server Manager console appears.

1. Open the Run dialog box and, in the Open text box, type mmc and click OK. An empty MMC console appears.
2. Click File > Add/Remove Snap-in. The *Add or Remove Snap-ins* dialog box appears.
3. Select Security Templates from the *Available snap-ins* list and click Add. The snap-in appears in the *Add or Remove Snap-ins* dialog box.
4. Click OK. The snap-in appears in the MMC console.
5. Click File > Save As. A *Save As* combo box appears.
6. Type a name for the console, to save it in the Administrative Tools program group.
7. Expand the *Security Templates* node.
8. Right-click the security template search path and, from the context menu, select New Template. A dialog box appears.
9. In the *Template name* field, type a name for the template and click OK. The new template appears in the console.

LEAVE the console open.

When you create a blank security template, there are no policies defined in it. Applying the blank template to a computer has no effect on it whatsoever.

WORKING WITH SECURITY TEMPLATE SETTINGS

Security templates contain many of the same settings as group policy objects, so you are already familiar with some of the elements of a template. For example, security templates contain the same local policy settings described previously; the template is just a different way to configure and deploy the policies. Security templates also provide a means for configuring the permissions associated with files, folders, registry entries, and services.

Security templates have more settings than Local Computer Policy, because a template includes options for both standalone computers and computers that participate in a domain.

■ Configuring Local Users and Groups

THE BOTTOM LINE

Windows Server 2012 R2 provides two separate interfaces for creating and managing local user accounts: the User Accounts control panel and the Local Users and Groups snap-in for MMC. Both of these interfaces provide access to the same Security Account Manager (SAM) where the user and group information is stored, so any changes you make using one interface will appear in the other.

Microsoft designed the User Accounts control panel and the Local Users and Groups snap-in for computer users with different levels of expertise, and they provide different degrees of access to the Security Account Manager, as follows:

- **User Accounts:** Microsoft designed the User Accounts control panel for relatively inexperienced end users; it provides a simplified interface with limited access to user accounts. With this interface, you can create local user accounts and modify their basic attributes, but you cannot create groups or manage group memberships (except for the Administrators group).
- **Local Users and Groups:** Microsoft includes this MMC snap-in as part of the Computer Management console; it provides full access to local users and groups, as well as all their attributes. Designed more for the technical specialist or system administrator, this interface is not difficult to use, but it does provide access to controls that beginning users generally do not need.

Using the User Accounts Control Panel

Windows Server 2012 R2 creates two local user accounts during the operating system installation process, the Administrator and Guest accounts. The setup program prompts the installer for an Administrator password during the installation, whereas the Guest account is disabled by default.

After the installation process is complete, the system restarts. Because only the Administrator account is available, the computer logs on using that account. This account has administrative privileges, so at this point you can create additional user accounts or modify the existing ones.

CREATING A NEW LOCAL USER ACCOUNT

To create a new user account with the User Accounts control panel, use the following procedure.

CREATE A NEW LOCAL USER ACCOUNT WITH THE CONTROL PANEL

GET READY. Log on to Windows Server 2012 R2, using an account with Administrator privileges. The Server Manager console appears.

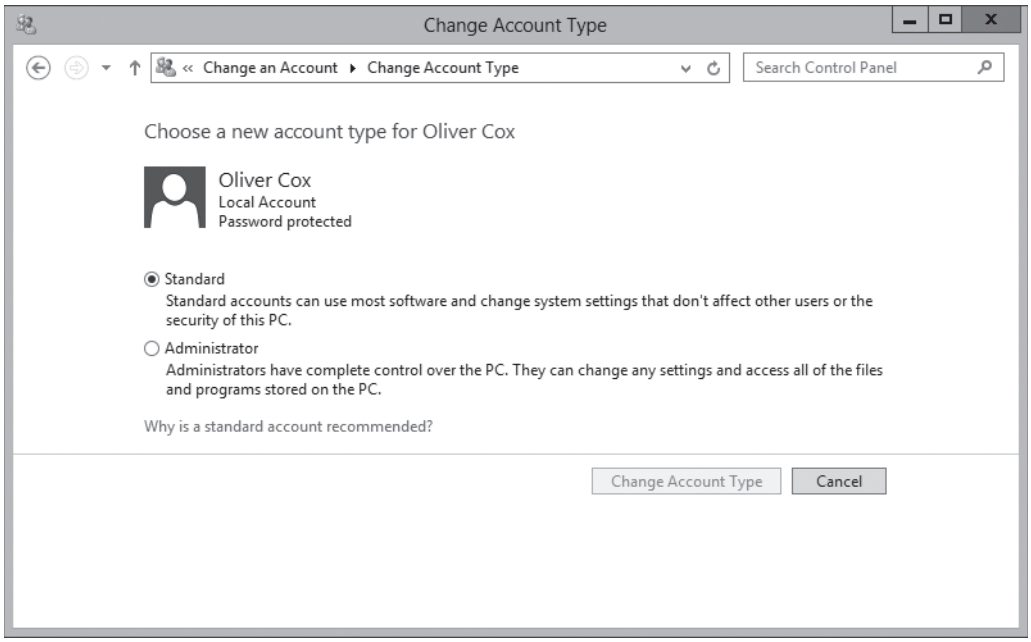
1. Open the Control Panel. The Control Panel window appears.
2. Click User Accounts. The User Accounts window appears.
3. Click User Accounts. The *Make changes to your user account* window appears.
4. Click Manage another account. The *Choose the user you would like to change* page appears.
5. Click Add a user account. The *Add a user* page appears.
6. Type a name for the new account in the User name text box, and a password for the account in the Password and Reenter password text boxes.
7. Type a phrase in the Password hint text box and click Next. The system creates the account.
8. Click Finish. The new account appears in the Manage Accounts window.

CLOSE the Manage Accounts window.

By default, this procedure creates standard accounts. To grant a local user administrative capabilities, you must change the account type, by using the interface shown in Figure 17-6.

Figure 17-6

The Change Account Type window



What the User Accounts control panel refers to as an account type is actually a group membership. Selecting the Standard user option adds the user account to the local Users group, whereas selecting the Administrator option adds the account to the Administrators group.

Using the Local Users and Groups Snap-In

The User Accounts control panel provides only partial access to local user accounts, and no access to groups other than the Users and Administrators groups. The Local Users and Groups snap-in, on the other hand, provides full access to all the local user and group accounts on the computer.

By default, the Local Users and Groups snap-in is part of the Computer Management console. However, you can also load the snap-in by itself, or create your own MMC console with any combination of snap-ins.

To create a local user account with the Local Users and Groups snap-in, use the following procedure.

CREATE A NEW LOCAL USER ACCOUNT WITH LOCAL USERS AND GROUPS

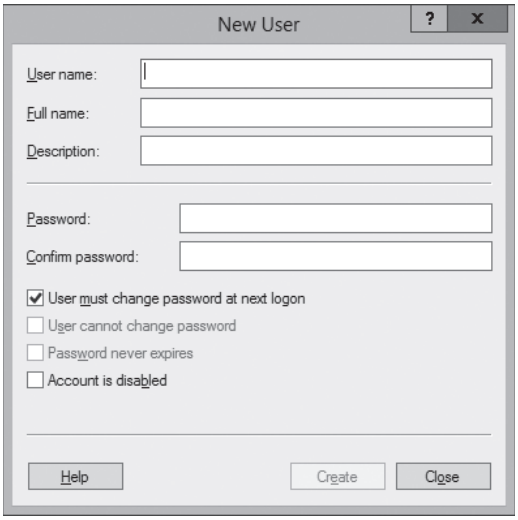
GET READY. Log on to Windows Server 2012 R2, using an account with Administrator privileges. The Server Manager console appears.

1. From the Tools menu, select Computer Management. The Computer Management console appears.
2. Expand the Local Users and Groups node and click Users. A list of the current local users appears.

3. Right-click the Users folder and, from the context menu, select New User. The New User dialog box appears, as shown in Figure 17-7.

Figure 17-7

The New User dialog box



4. In the User name text box, type the name you want to assign to the user account. This is the only required field in the dialog box.
5. Specify a Full name and a Description for the account, if desired.
6. In the Password and Confirm password text boxes, type a password for the account, if desired.
7. Select or clear the four checkboxes to control the following functions:
   - User must change password at next logon: forces the new user to change the password after logging on for the first time. Select this option to assign an initial password and enable users to control their own passwords after the first logon. You cannot select this option if you selected the *Password never expires* check box. Selecting this option automatically clears the *User cannot change password* check box.
   - User cannot change password: prevents the user from changing the account password. Select this option if you want to retain control over the account password, such as when multiple users log on with the same user account. This option is also commonly used to manage service account passwords. You cannot select this option if you selected the *User must change password at next logon* check box.
   - Password never expires: prevents the existing password from ever expiring. This option automatically clears the *User must change password at next logon* check box. This option is also commonly used to manage service account passwords.
   - Account is disabled: disables the user account, preventing anyone from using it to log on.
8. Click Create. The new account is added to the user list and the console clears the dialog box, leaving it ready for the creation of another user account.
9. Click Close.

CLOSE the Computer Management console.

CREATING A LOCAL GROUP

To create a local group with the Local Users and Groups snap-in, use the following procedure.

CREATE A LOCAL GROUP

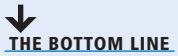
GET READY. Log on to Windows Server 2012 R2, using an account with Administrator privileges. The Server Manager console appears.

1. From the Tools menu, select Computer Management. The Computer Management console appears.
2. Expand the Local Users and Groups node and click Groups. A list of local groups appears.
3. Right-click the Groups folder and then, from the context menu, select New Group. The New Group dialog box appears.
4. In the Group name text box, type the name you want to assign to the group. This is the only required field in the dialog box. If desired, specify a Description for the group.
5. Click the Add button. The Select Users dialog box appears.
6. Type the names of the users that you want to add to the group, separated by semicolons, in the text box, and then click OK. The users are added to the Members list. You can also type part of a user name and click Check Names to complete the name or click Advanced to search for users.
7. Click Create to create the group and populate it with the user(s) you specify. The console clears the dialog box, leaving it ready for the creation of another group.
8. Click Close.

CLOSE the Computer Management console.

Local groups have no user-definable attributes other than a members list, so you only can add or remove members when you open an existing group. As noted previously, local groups cannot have other local groups as members. If the computer is a member of a Windows domain, a local group can have domain users and domain groups as members.

■ Configuring User Account Control



One of the most common Windows security problems arises from the fact that many users perform their everyday computing tasks with more system access than they need. Logging on as Administrator or user that is a member of the Administrators group grants the user full access to all areas of the operating system. This degree of system access is not necessary to run many of the applications and perform many of the tasks users require every day; it is needed only for certain administrative functions, such as installing system-wide software or configuring system parameters.

For most users, logging on with administrative privileges all the time is simply a matter of convenience. Microsoft recommends logging on as a standard user, and to use administrative privileges only when you need them. However, many technical specialists who do this frequently find themselves encountering situations in which they need administrative access. There is a surprisingly large number of common, and even mundane, Windows tasks that require administrative access, and the inability to perform the tasks can negatively affect a user's productivity.

Microsoft addressed this problem by keeping all Windows Server 2012 R2 users from accessing the system using administrative privileges unless the privileges are required to perform the task at hand. The mechanism that does this is called ***User Account Control (UAC)***.

Configuring User Account Control Settings

Windows Server 2012 R2 enables User Account Control by default, but it is possible to configure its properties, or even disable it completely.

In Windows Server 2012 R2, four UAC settings are available through the Control Panel. To configure UAC through the Control Panel, use the following procedure.

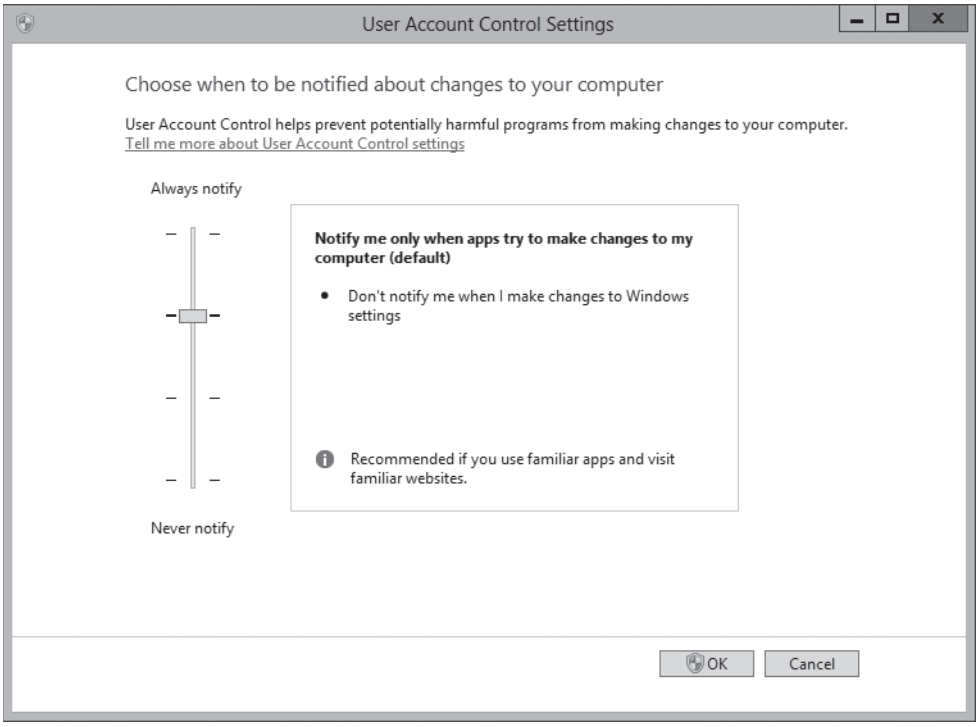
CONFIGURE UAC SETTINGS

GET READY. Log on to Windows Server 2012 R2, using an account with Administrator privileges. The Server Manager console appears.

1. Open the Control Panel. The Control Panel window appears.
2. Click System and Security > Action Center. The Action Center window appears.
3. Click Change User Account Control settings. The *User Account Control Settings* dialog box appears, as shown in Figure 17-8.

Figure 17-8

The User Account Control Settings dialog box



4. Adjust the slider to one of the following settings and click OK.
   - Always notify me
   - Notify me only when programs try to make changes to my computer
   - Notify me only when programs try to make changes to my computer (do not dim my desktop)
   - Never notify me

CLOSE the Action Center window.

Although the Control Panel provides some control over UAC, the most granular control over UAC properties is still through the Security Options node in Group Policy and Local Security Policy.

■ Business Case Scenarios

Scenario 17-1: Understanding Group Policy Planning

You are the administrator for Coho Winery, Inc. a large wine distribution company that has locations in the United States and Canada. In the last six months, Coho Winery, Inc. has purchased several smaller distribution companies. As you integrate them into your forest, you want to allow them to remain autonomous in their management of desktops and security. In the process of making their domains part of your corporate network, you have some policies that you want to become part of their environment, and others that you do not want to implement at this time. As you discuss this with your IT team, your manager asks you to explain which features in Windows Server 2012 R2 enable you to provide the Group Policy flexibility needed by the new Active Directory structure. List several of the features in Windows Server 2012 R2 Group Policy that will enable Coho Winery, Inc. to achieve its postacquisition goals.

Scenario 17-2: Deploying Security Templates

You are a network administrator planning a security template deloyment on a network that consists of 100 workstation. The workstations are all running various versions of Microsoft Windows, broken down as follows:

- Windows 7: 30 workstations
- Windows XP Professional: 40 workstations
- Windows XP Home Edition: 20 workstations
- Windows 2000 Professional: 10 workstations

In the past, some computers on the network have been compromised because end users modified their workstation security configurations. Your task is to deploy your security templates on the workstations in such a way that end users cannot modify them. To accomplish this goal, you decide to use Group Policy to deploy the templates to an Active Directory Domain Services OU object that contains all of the workstations.
Based on the information provided, answer the following questions.

1. How many of the workstations cannot receive their security template settings from a GPO linked to an AD DS container?
2. Which of the following methods can you use to deploy your security templates on the workstations that do not support Group Policy, while still accomplishing your assigned goals?
   a. Upgrade all of the computers that do not support Group Policy to Windows 7.
   b. Run the Security Templates snap-in on each computer and load the appropriate security template.
   c. Create a logon script that uses Secedit.exe to import the security template on each computer.
   d. Run the Security Configuration and Analysis snap-in on each computer and use it to import the appropriate security template.

LESSON 18

Configuring Application Restriction Policies

Installing Software with Group Policy

THE BOTTOM LINE

You can use Group Policy to install, upgrade, patch, or remove software applications when a computer starts, when a user logs on to the network, or when a user accesses a file associated with an application that is not currently on the user's computer.

In addition, you can use Group Policy to fix problems associated with applications. For example, if a user inadvertently deletes a file from an application, Group Policy can launch a repair process that will fix the application. To perform these tasks, Group Policy works together with the Windows Installer Service.

Windows Server 2012 R2 uses the Windows Installer with Group Policy to install and manage software that is packaged into Microsoft Installer files, with an .msi extension.

Windows Installer consists of two components: one for the client-side and another for the server-side. The client-side component is called the *Windows Installer Service*. This client-side component is responsible for automating the installation and configuration of the designated software.

The Windows Installer Service requires a package file that contains all the pertinent information about the software. This package file consists of the following information:

- An ***.msi file***, which is a relational database file that is copied to the target computer system, with the program files it deploys. In addition to providing installation information, this database file assists in the ***self-healing*** process for damaged applications and clean application removal.
- External source files that are required for software installation or removal.
- Summary information about the software and the package.
- A reference point to the path where the installation files are located.

To install software by using Group Policy, you must have Windows Installer–enabled applications. An application that has an approval stamp from Microsoft on its packaging, including the Certified for Windows Server 2012 R2 logo, is Windows Installer–enabled by default. This means the application provides support for Group Policy deployments using an .msi file.

At times, you might need to modify Windows Installer files to better suit the needs of your network. Modifications to .msi files require transform files, which have an .mst extension.

Windows Installer files with the .msp extension serve as patch files. You use ***patch files*** to apply service packs and hotfixes to installed software. Unlike an .msi file, a patch package does not include a complete database. Instead, it contains, at minimum, a database transform procedure that adds patching information to the target installation package database. For this reason, .msp files should be located in the same folder as the original .msi file when you want the patch to be applied as part of the Group Policy software installation.

■ Configuring Software Restriction Policies

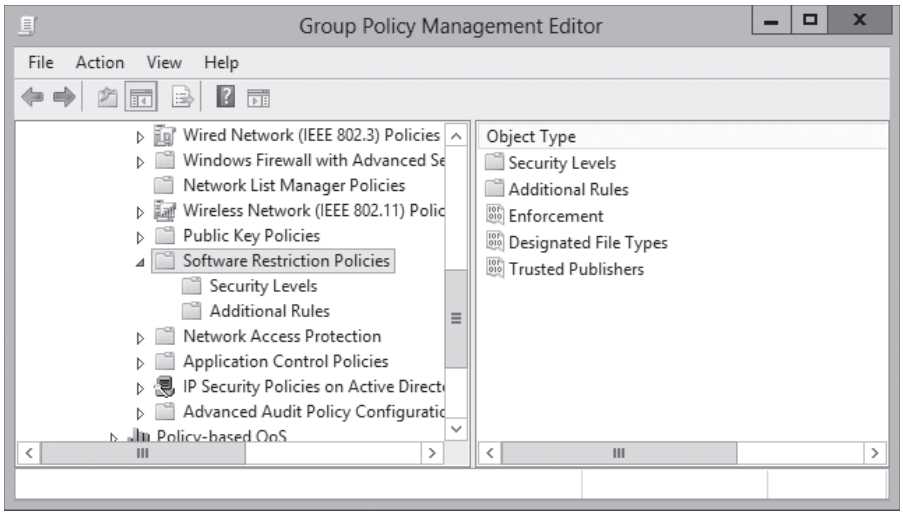
THE BOTTOM LINE

The options in the Software Restriction Policies node provide organizations greater control in preventing potentially dangerous applications from running. Software restriction policies are designed to identify software and control its execution. In addition, you can control who is affected by the policies.

The *Software Restriction Policies* node is found in the *Windows Settings\Security Settings* node of the *User Configuration or the Computer Configuration* node of a Group Policy object. By default, the Software Restriction Policies folder is empty. When you create a new policy, two subfolders appear: Security Levels and Additional Rules, as shown in Figure 18-1. The *Security Levels* folder enables you to define the default behavior from which all rules are created. The criteria for each executable program are defined in the Additional Rules folder.

Figure 18-1

The Software Restriction Policies folder



Enforcing Restrictions

Prior to creating any rules that govern the restriction or allowance of executable files, you should understand how the rules work by default. If a policy does not enforce restrictions, executable files run based on the permissions that users or groups have in the NTFS file system.

When considering software restriction policies, you must determine your approach to enforcing restrictions. You can use three basic strategies for enforcing restrictions, as follows:

- **Unrestricted:** enables all applications to run, except those that are specifically excluded.
- **Disallowed:** prevents all applications from running except those that are specifically allowed.
- **Basic User:** prevents any application from running that requires administrative rights, but enables programs to run that only require resources that are accessible by normal users.

The approach you take depends on the needs of your particular organization. By default, the Software Restriction Policies area has an Unrestricted value in the Default Security Level setting.

To modify the *Default Security Level* setting to *Disallowed*, use the following procedure.

MODIFY THE DEFAULT SECURITY LEVEL

GET READY. Log on to a server running Windows Server 2012 R2, using an account with domain Administrator privileges.

1. From the Tools menu in the *Server Manager* console, select Group Policy Management. The *Group Policy Management* console appears.
2. Expand the forest container and browse to your domain. Then expand the domain container and select the Group Policy Objects folder. The GPOs that currently exist in the domain appear in the *Contents* tab.
3. Right-click a GPO and click Edit. A *Group Policy Management Editor* window appears.
4. Browse to the Software Restriction Policies node under either Computer Configuration or User Configuration.
5. Right-click Software Restriction Policies and select New Software Restriction Policies. The folders containing the new policies appear.
6. In the details pane, double-click Security Levels. Note the checkmark on the *Unrestricted* icon, which is the default setting.
7. Right-click the Disallowed security level and, from the context menu, select Set As Default. A *Software Restriction Policies* warning message box appears, informing you of the consequences of more restrictive policies.

CLOSE the *Group Policy Management Editor* and *Group Policy Management* consoles.

You have now modified the Default Security Level for a software restriction policy.

Configuring Software Restriction Properties

Within the *Software Restriction Policies* folder, you can configure three specific properties to provide additional settings that apply to all policies when implemented.

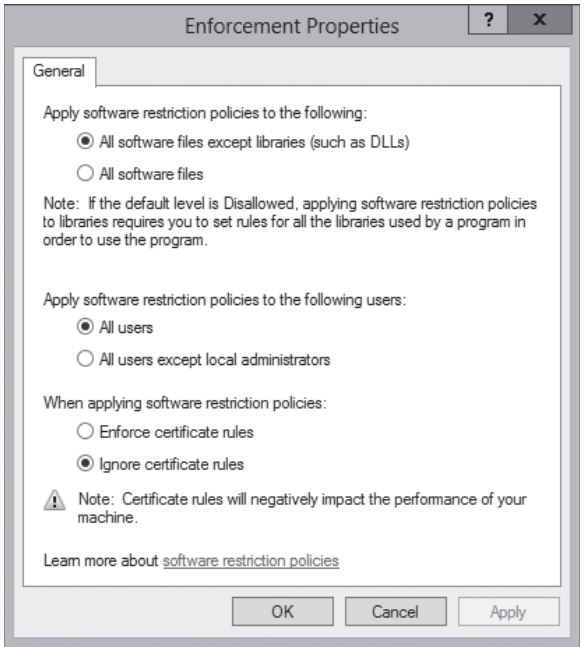
These three properties are enforcement, designated file types, and trusted publishers, as described in the following sections.

ENFORCEMENT

As shown in Figure 18-2, Enforcement properties enable you to determine whether the policies apply to all files or whether library files, such as Dynamic Link Library (DLL), are excluded. Excluding DLLs is the default, which is the most practical method of enforcement. For example, if the *Default Security Level* for the policy is set to *Disallowed* and the *Enforcement* properties are set to *All Software Files*, you would need to create a rule that checked every DLL before the program could be allowed or denied. By contrast, excluding DLL files using the default Enforcement property does not require you to define individual rules for each DLL file.

Figure 18-2

Configuring Enforcement properties



DESIGNATED FILE TYPES

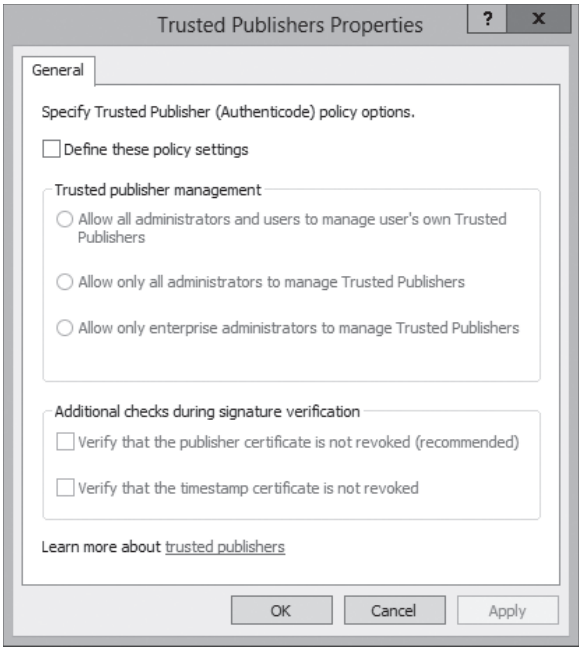
The Designated File Types properties within the Software Restriction Policies folder, specify file types that are executable. File types that are designated as executable or program files are shared by all rules, although you can specify a list for a computer policy that is different from one that is specified for a user policy.

TRUSTED PUBLISHERS

Finally, the Trusted Publishers properties enable you to control how systems handle certificate rules. In the Properties dialog box for Trusted Publishers, as shown in Figure 18-3, the first setting enables you to specify which users are permitted to manage trusted certificate sources. By default, local computer administrators have the right to specify trusted publishers on the local computer and enterprise administrators have the right to specify trusted publishers in an OU. From a security standpoint, in a high-security network, users should not be allowed to determine the sources from which certificates can be obtained.

Figure 18-3

Configuring Trusted Publishers properties



In addition, the Trusted Publisher Properties sheet also lets you verify that a certificate has not been revoked. If a certificate has been revoked, the user should not be permitted access to network resources. You have the option of checking either the *Publisher* or the *Timestamp* of the certificate to determine whether it has been revoked.

■ Using AppLocker

THE BOTTOM LINE

Software restriction policies can be a powerful tool, but they can also require a great deal of administrative overhead. If you elect to disallow all applications except those matching the rules you create, there are a great many programs in Windows Server 2012 R2 that need rules, in addition to the applications you want to install. You must also create the rules manually, which can be an onerous chore.

AppLocker, also known as ***application control policies***, is a Windows feature that is essentially an updated version of the concept implemented in software restriction policies. AppLocker also uses rules, which you must manage, but the process of creating the rules is much easier, thanks to a wizard-based interface.

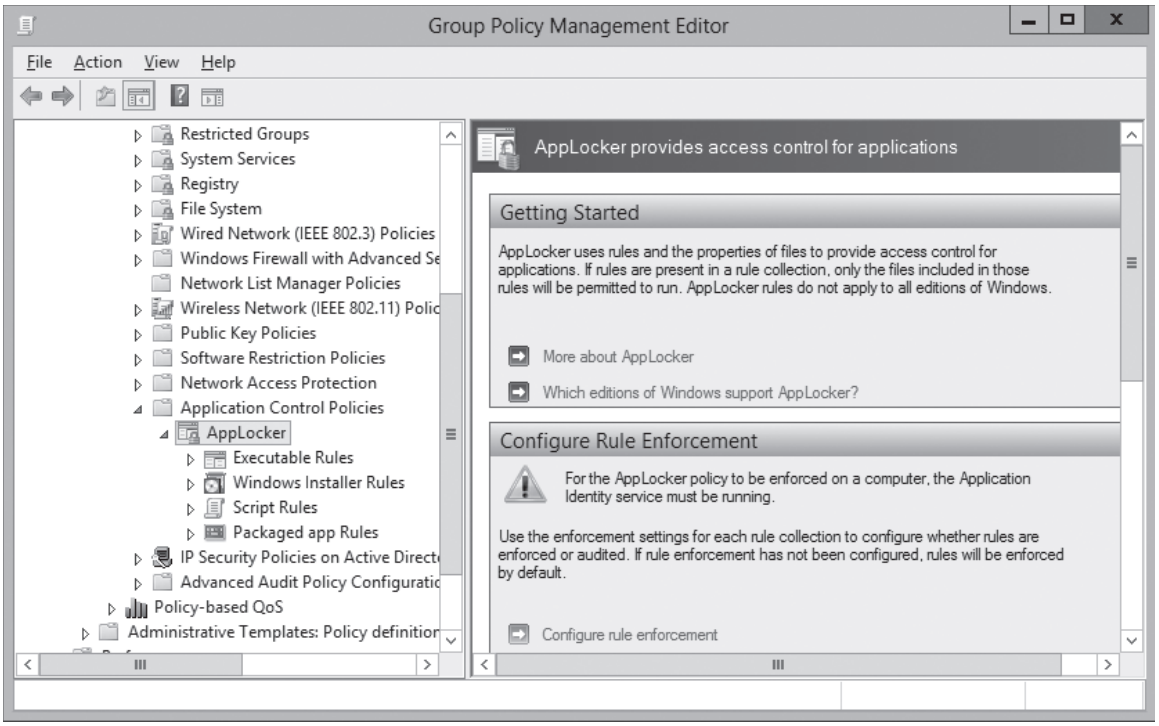
AppLocker is also more flexible than software restriction policies. You can apply AppLocker rules to specific users and groups and also create rules that support all future versions of an application. The primary disadvantage of AppLocker is that you can apply only the policies to computers running Windows 7 and Windows Server 2008 R2 or later.

Understanding Rule Types

The AppLocker settings are located in Group Policy objects in the Computer Configuration\Windows Settings\Security Settings\Application Control Policies\ AppLocker container, as shown in Figure 18-4.

Figure 18-4

The AppLocker container in a GPO



In the AppLocker container, four nodes contain the basic rule types, as follows:

- **Executable Rules:** contains rules that apply to files with .exe and .com extensions.
- **Windows Installer Rules:** contains rules that apply to Windows Installer packages with .msi and .msp extensions.
- **Script Rules:** contains rules that apply to script files with .ps1, .bat, .cmd, .vbs, and .js extensions.
- **Packaged app Rules:** contains rules that apply to applications purchased through the Windows Store.

Each of the rules you create in each of these containers can allow or block access to specific resources, based on one of the following criteria:

- **Publisher:** identifies code-signed applications by means of a digital signature extracted from an application file. You can also create publisher rules that apply to all future versions of an application.
- **Path:** identifies applications by specifying a file or folder name. The potential vulnerability of this type of rule is that any file can match the rule, as long as it is the correct name or location.
- **File Hash:** identifies applications based on a digital fingerprint that remains valid even when the name or location of the executable file changes. This type of rule functions much like its equivalent in software restriction policies; in AppLocker, however, the process of creating the rules and generating file hashes is much easier.

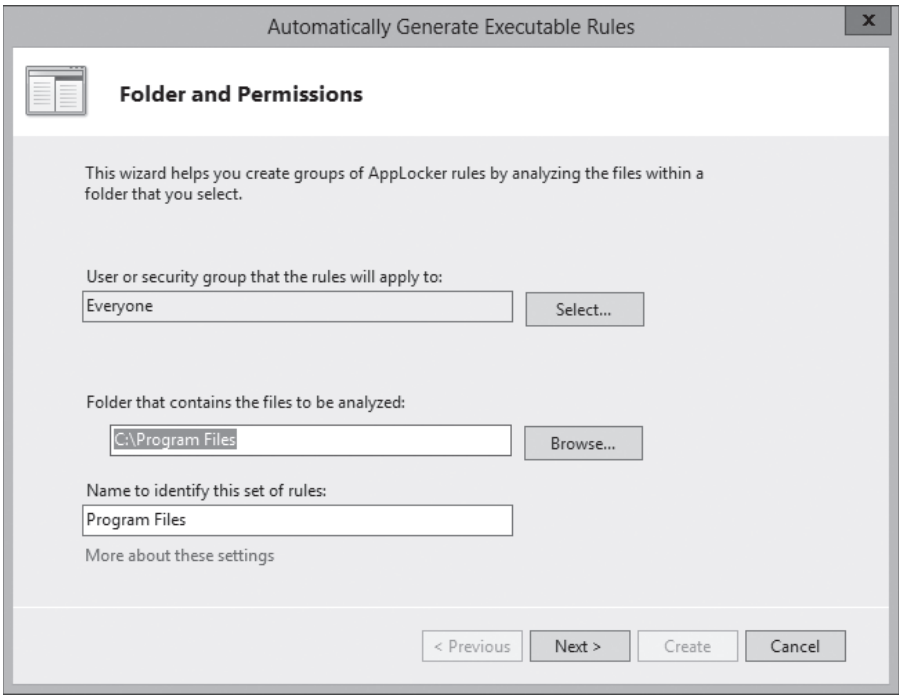
Creating Rules Automatically

The greatest advantage of AppLocker over software restriction policies is the capability to create rules automatically.

When you right-click one of the three rules containers and select *Create Rules Automatically* from the context menu, an *Automatically Generate Rules Wizard* appears, as shown in Figure 18-5.

Figure 18-5

The Automatically Generate Executable Rules Wizard

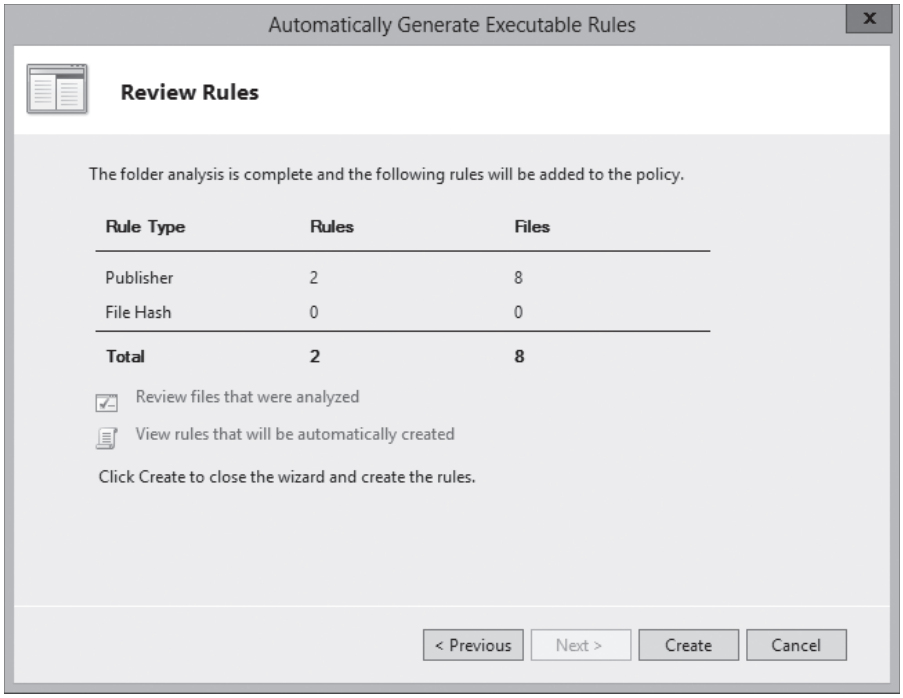


After specifying the folder to be analyzed and the users or groups to which the rules should apply, a *Rule Preferences* page appears, enabling you to specify the types of rules you want to create.

The wizard then displays a summary of its results in the *Review Rules* page, as shown in Figure 18-6, and adds the rules to the container.

Figure 18-6

The Review Rules page of the Automatically Generate Executable Rules Wizard



Creating Rules Manually

In addition to creating rules automatically, you can also do it manually, using a wizard-based interface you activate by selecting *Create New Rule* from the context menu for one of the three rule containers.

The wizard prompts you for the following information:

- **Action:** specifies whether you want to allow or deny the user or group access to the resource. In AppLocker, explicit deny rules always override allow rules.
- **User or group:** specifies the name of the user or group to which the policy should apply.
- **Conditions:** specifies whether you want to create a publisher, path, or file hash rule. The wizard generates an additional page for whichever option you select, enabling you to configure its parameters.
- **Exceptions:** enables you to specify exceptions to the rule you create, using any of the three conditions: publisher, path, or file hash.

■ Business Case Scenarios

Scenario 18-1: Planning Group Policy Software Deployments

Your company, a healthcare organization, is currently working toward compliance with new government standards on patient confidentiality. Your IT department has decided that using software restriction policies with standard user access permissions will help to fulfill the necessary security requirements. You are preparing an implementation plan that is based on user needs and security requirements. Users should not be able to access any programs with the exception of those that are pertinent to their jobs. In addition, the user needs within the organization are as follows:

- Users only need access to e-mail and a patient database.
- The patient database has its own built-in security access system that is configured for each user based on the user's needs within the program.
- All user accounts are located in containers based on the user's office location.

In addition, the following points should be considered in your implementation plan:

- Software restriction policy settings should not affect settings that are already in place within existing GPOs. If problems arise with software restriction policies, they should be easy to rectify, without affecting other security areas.
- Administrator accounts should not be affected by software restrictions.
- Other applications should not be affected by any of the restrictions.

List the key points that should be part of your implementation plan based on the information provided here.

Scenario 18-2: Using AppLocker

Sophie is planning on using AppLocker to control access to applications on a new network she has constructed for the Research and Development department at a major aerospace firm. The software developers in the department have recently deployed a new application called Virtual Wind Tunnel, which is based on government project research and is therefore classified. All of the full-time personnel have sufficient clearance to use the application, but the interns in the department do not. Sophie has placed the user accounts for everyone in the department into a security group called ResDev. The interns are also members of a group called RDint.

How can Sophie use AppLocker to provide everyone in the department with access to the Virtual Wind Tunnel application without changing the group memberships and without having to apply policies to individual users?

19 LESSON

Configuring Windows Firewall

■ Building a Firewall

THE BOTTOM LINE

You may have locked the door to the computer center in which the servers are located, but the computers are still connected to the network. A network is another type of door, or rather a series of doors, which can allow data out or allow it in. To provide services to your users, some of the doors must be open at least some of the time, but server administrators must make sure that only the right doors are left open.

A ***firewall*** is a software program that protects a computer or a network by allowing certain types of network traffic in and out of the system while blocking others. A firewall is essentially a series of filters that examine the contents of packets and the traffic patterns to and from the network to determine which packets they should allow to pass through the filter.

Some of the hazards that firewalls can protect against are as follows:

- Network scanner applications that probe systems for unguarded ports, which are essentially unlocked doors that attackers can use to gain access to the system.
- Trojan horse applications that open a connection to a computer on the Internet, enabling an attacker on the outside to run programs or store data on the system.
- Attackers that obtain passwords by illicit means, such as social engineering, and then use remote access technologies to log on to a computer from another location and compromise its data and programming.
- Denial of service attacks that use authorized access points to bombard a system with traffic, preventing legitimate traffic from reaching the computer.

The object of a firewall is to permit all the traffic in and out that legitimate users need to perform their assigned tasks, and block everything else. When you work with firewalls, you are not concerned with subjects such as authentication and authorization. These are mechanisms that control who is able to get through the server's open doors. The firewall is all about which doors are left open, and which doors are shut tight.

Understanding Windows Firewall Settings

Windows Server 2012 R2 includes a firewall program called *Windows Firewall*, which is activated by default on all systems.

By default, Windows Firewall blocks most network traffic from entering the computer. Firewalls examine the contents of each packet entering and leaving the computer and comparing the information they find to a series of rules, which specify which packets are allowed to pass through the firewall and which are blocked.

The TCP/IP (Transmission Control Protocol/Internet Protocol) protocols that Windows systems use to communicate function by packaging application data using a series of layered protocols that define where the data comes from and where it is going. The three most important criteria that firewalls can use in their rules are as follows:

- **IP addresses:** identify specific hosts on the network. You can use IP addresses to configure a firewall to allow only traffic from specific computers or networks in and out.
- **Protocol numbers:** specify whether the packet contains TCP or UDP (User Datagram Protocol) traffic. You can filter protocol numbers to block packets containing certain types of traffic. Windows computers typically use UDP for brief message exchanges, such as DNS (Domain Name System) and DHCP (Dynamic Host Configuration Protocol) transactions. TCP packets usually carry larger amounts of data, such as the files exchanged by web, file, and print servers.
- ***Port numbers*:** identify specific applications running on the computer. The most common firewall rules use port numbers to specify the types of application traffic the computer is allowed to send and receive. For example, a web server usually receives its incoming packets to port number 80. Unless the firewall has a rule opening port 80 to incoming traffic, the web server cannot function in its default configuration.

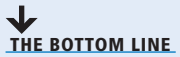
Firewall rules can function in two ways:

- Admit all traffic, except that which conforms to the applied rules
- Block all traffic, except that which conforms to the applied rules

Generally speaking, blocking all traffic by default is the more secure arrangement. From the server administrator's standpoint, you start with a completely blocked system, and then test your applications. When an application fails to function properly because network access is blocked, you create a rule that opens up the ports the application needs to communicate.

This is the method that Windows Firewall uses by default for incoming network traffic. There are default rules preconfigured into the firewall that are designed to admit the traffic used by standard Windows networking functions, such as file and printer sharing. For outgoing network traffic, Windows Firewall uses the other method, allowing all traffic to pass the firewall except that which conforms to a rule.

■ Using the Windows Firewall Control Panel



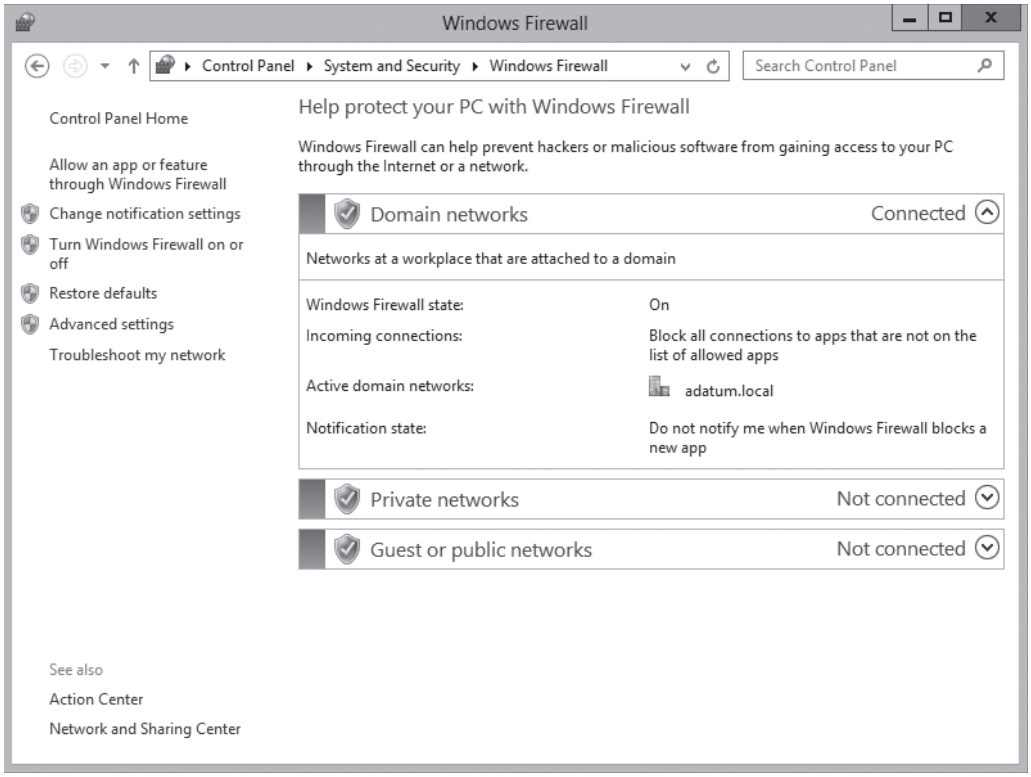
The Windows Firewall control panel provides the easiest and safest access to the firewall controls. These controls are usually sufficient for most server administrators, unless the system has special requirements or you work with custom server applications.

When you open the *Windows Firewall* window from the *Control Panel*, as shown in Figure 19-1, you see the following information:

- Whether the computer is connected to a domain, private, or public network
- Whether the Windows Firewall service is currently turned on or off
- Whether inbound and outbound connections are blocked
- The name of the currently active network
- Whether users are notified when a program is blocked

Figure 19-1

The Windows Firewall control panel window



On the left side of the window is a series of links, which provide the following functions:

- **Allow an app or feature through Windows Firewall:** displays the Allowed apps dialog box, in which you can select the applications that can send traffic through the firewall.
- **Change notification settings:** displays the Customize settings dialog box, in which you can adjust the notification settings for each of the three profiles.
- **Turn Windows Firewall on or off:** displays the Customize settings dialog box, in which you can toggle the state of the firewall in each of the three profiles.
- **Restore defaults:** returns all firewall settings to their installation defaults.
- **Advanced settings:** launches the Windows Firewall with Advanced Security console.
- **Troubleshoot my network:** launches the Network and Internet troubleshooter.

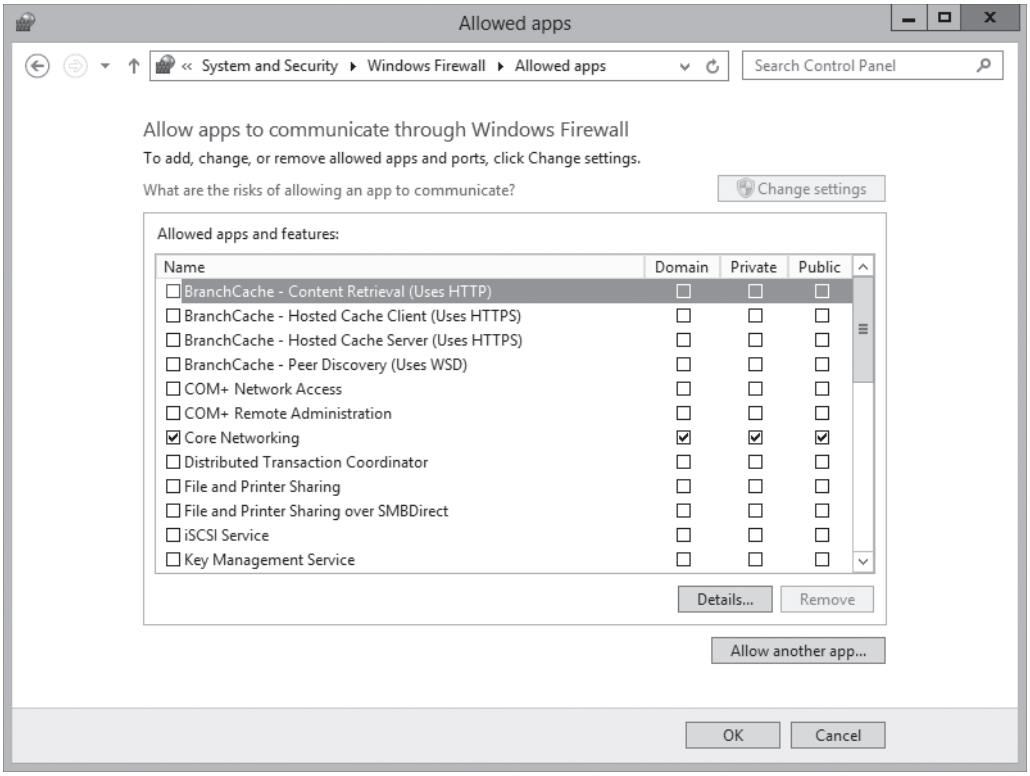
Allowing Applications

Sometimes, administrators might be required to modify the firewall settings in other ways, typically because a specific application requires access to a port not anticipated by the firewall's default rules.

To do this, you can use the Allowed Apps dialog box in the Windows Firewall control panel, as shown in Figure 19-2.

Figure 19-2

The Allowed Apps dialog box for Windows Firewall



Opening a port in a server's firewall is an inherently dangerous activity. The more open doors you put in a wall, the more opportunities that intruders can exploit to get in. Windows Firewall provides two basic methods for opening a hole in your firewall: opening a port and allowing an application. Both are risky, but the latter of the two is less so. This is because when you open a port by creating a rule in the Windows Firewall with Advanced Security console, the port stays open permanently. When you allow an application through the firewall using the Control Panel, the specified port is open only while the program is running. When you terminate the program, the firewall closes the port.

To allow an application through the firewall by using Control Panel, use the following procedure.

ALLOW AN APPLICATION

GET READY. Log on to a server running Windows Server 2012 R2, using an account with domain administrator privileges.

1. In the *Server Manager* console, open the Control Panel and click System and Security > Windows Firewall. The *Windows Firewall* window appears.
2. Click Allow an app or feature through Windows Firewall. The *Allowed Apps* dialog box appears.
3. Scroll down in the Allowed apps and features list and select the check box for the application you want to allow through the firewall.
4. Click OK to close the *Allowed Apps* dialog box.

CLOSE the *Windows Firewall* window.

The applications listed on the Allowed Apps dialog box are based on the roles and features installed on the server. Each listed application corresponds to one or more firewall rules, which the Control Panel activates and deactivates as needed.

Unlike earlier versions, the Windows Server 2012 and Windows Server 2012 R2 versions of the Windows Firewall control panel do not provide direct access to port numbers. For more precise control over the firewall, you must use the *Windows Firewall with Advanced Security* console, which you can access by clicking the *Advanced Settings* link in the *Windows Firewall* control panel, or by selecting it from the *Tools* menu in *Server Manager*.

■ Using the Windows Firewall with Advanced Security Console

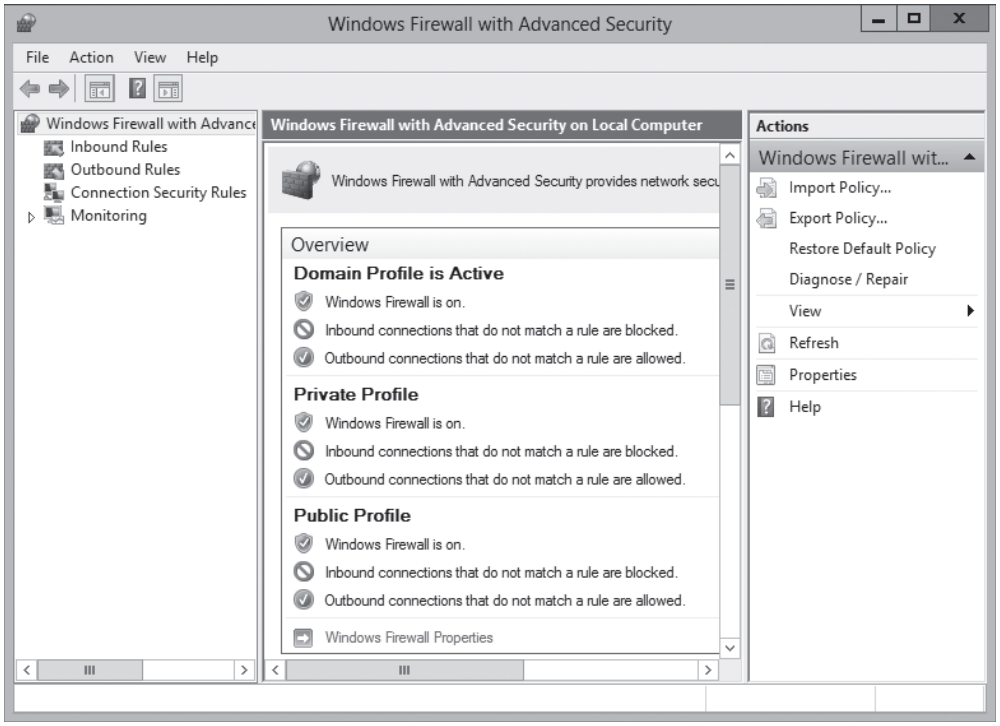
THE BOTTOM LINE

The Windows Firewall control panel is designed to enable administrators and advanced users to manage basic firewall settings. For full access to the Windows Firewall configuration settings, you must use the *Windows Firewall With Advanced Security* snap-in for the *Microsoft Management* console.

To open the console, open *Server Manager* and, from the *Tools* menu, select *Windows Firewall With Advanced Security*. The *Windows Firewall with Advanced Security* console appears, as shown in Figure 19-3.

Figure 19-3

The *Windows Firewall with Advanced Security* console



Configuring Profile Settings

At the top of the Windows Firewall With Advanced Security console's middle pane, in the Overview section, are status displays for the computer's three network location profiles. If you connect the computer to a different network (which is admittedly not likely with a server), Windows Firewall can load a different profile and a different set of rules.

The default Windows Firewall configuration calls for the same basic settings for all three profiles, as follows:

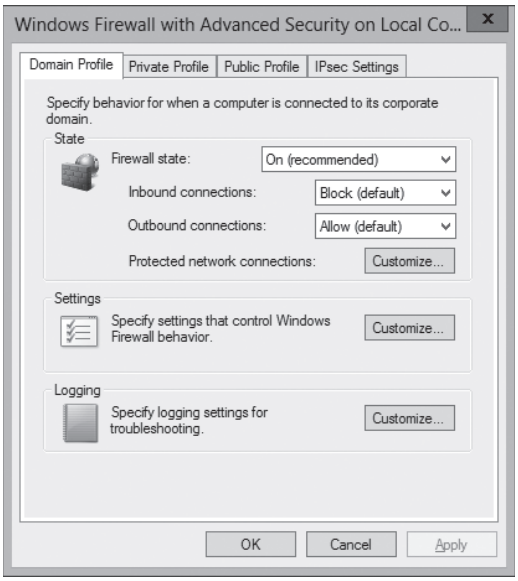
- The firewall is turned on.
- Incoming traffic is blocked unless it matches a rule.
- Outgoing traffic is allowed unless it matches a rule.

You can change this default behavior by clicking the *Windows Firewall Properties* link, which displays the *Windows Firewall With Advanced Security On Local Computer* dialog box, as shown in Figure 19-4.

In this dialog box, each of the three network location profile has a tab with identical controls that enables you to modify the default profile settings. You can, for example, configure the firewall to shut down completely when it is connected to a domain network, and turn the firewall on with its most protective settings when you connect the computer to a public network. You can also configure the firewall's notification options, its logging behavior, and how it reacts when rules conflict.

Figure 19-4

The *Windows Firewall with Advanced Security on Local Computer* dialog box



Creating Rules

The allowed applications that you can configure in the Windows Firewall control panel are a relatively friendly method for working with firewall rules. In the Windows Firewall With Advanced Security console, you can work with the rules in their raw form.

Selecting either *Inbound Rules* or *Outbound Rules* in the left pane displays a list of all the rules operating in that direction, as shown in Figure 19-5. The rules that are currently operational have a check mark in a green circle, whereas the rules not in force are grayed out.

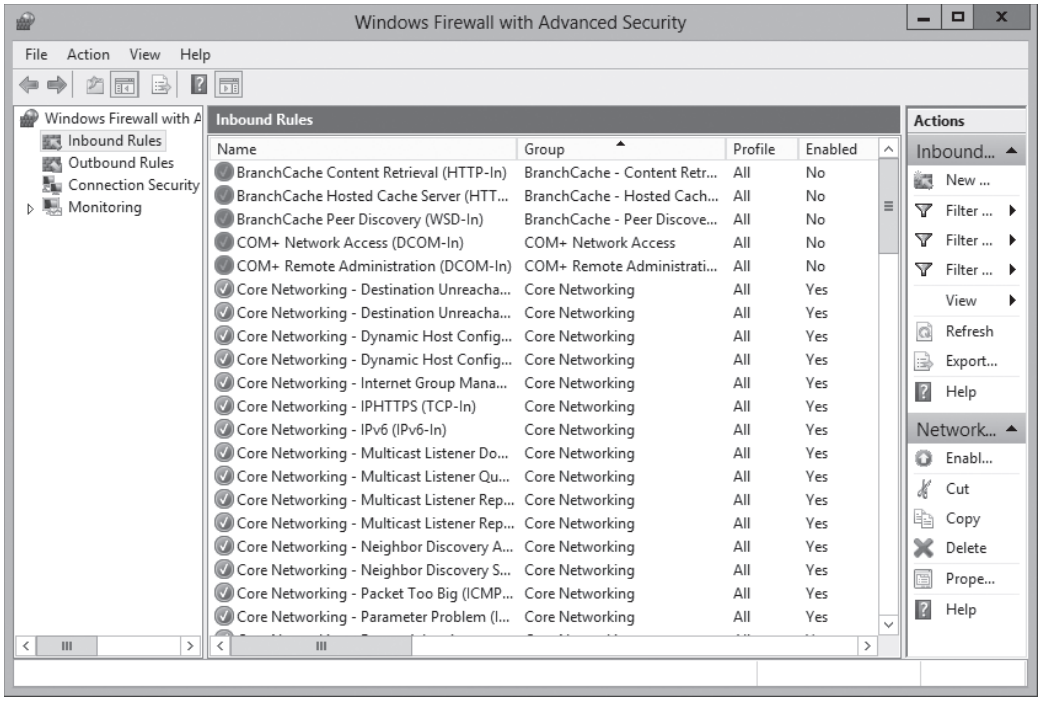


Figure 19-5

The Inbound Rules list in the Windows Firewall with Advanced Security console

Creating new rules with this interface provides more flexibility than the Windows Firewall control panel. When you right-click the *Inbound Rules* (or *Outbound Rules*) node and select *New Rule* from the context menu, the *New Inbound* (or *Outbound*) *Rule Wizard* takes you through the process of configuring the following sets of parameters:

- **Rule Type:** specifies whether you want to create a program rule, a port rule, a variant on one of the predefined rules, or a custom rule. This selection determines which of the following pages the wizard displays.
- **Program:** specifies whether the rule applies to all programs, to one specific program, or to a specific service. This is the equivalent of defining an allowed application in the Windows Firewall control panel, except that you must specify the exact path to the application.
- **Protocol and Ports:** specifies the network or transport layer protocol and the local and remote ports to which the rule applies. This enables you to specify the exact types of traffic that the rule should block or allow. To create rules in this way, you must be familiar with the protocols and ports that an application uses to communicate at both ends of the connection.
- **Predefined Rules:** specifies which predefined rules defining specific network connectivity requirements the wizard should create.
- **Scope:** specifies the IP addresses of the local and remote systems to which the rule applies. This enables you to block or allow traffic between specific computers.
- **Action:** specifies the action the firewall should take when a packet matches the rule. You configure the rule to allow traffic if it is blocked by default, or block traffic if it is allowed by default. You can also configure the rule to allow traffic only when the connection between the communicating computers is secured using IPsec.
- **Profile:** specifies the profile(s) to which the rule should apply: domain, private, and public.
- **Name:** specifies a name and (optionally) a description for the rule.

The rules you can create by using the wizards range from simple program rules, just like those you can create in the Windows Firewall control panel, to highly complex and specific rules that block or allow only specific types of traffic between specific computers. The more complicated the rules become, however, the more you need to know about TCP/IP communications in general and the specific behavior of your applications. Modifying the default firewall settings to accommodate some special applications is relatively simple, but creating an entirely new firewall configuration is a formidable task.

Importing and Exporting Rules

The process of creating and modifying rules in the Windows Firewall with Advanced Security console can be time-consuming, and repeating the process on multiple computers even more so. Therefore, the console makes it possible for you to save the rules and settings you create by exporting them to a policy file.

A policy file is a file with a .wfw extension that contains all the property settings in a Windows Firewall installation, as well as all of its rules, including the preconfigured rules and the ones you created or modified. To create a policy file, use the following procedure.

EXPORT WINDOWS FIREWALL RULES

GET READY. Log on to a server running Windows Server 2012 R2, using an account with domain administrator privileges.

1. In the *Server Manager* console, select *Windows Firewall With Advanced Security* from the Tools menu. The *Windows Firewall with Advanced Security* console appears.
2. Modify the inbound or outbound firewall rules or create new rules as needed.
3. In the left pane, select the Windows Firewall with Advanced Security on Local Computer node.
4. From the Action menu, select Export Policy. The Save As combo box appears.
5. In the File Name text box, type a name for the policy file and click Save.

CLOSE the Windows Firewall With Advanced Security console.

You can then duplicate the rules and settings on another computer by copying the file and using the *Import Policy* function to read in the contents, as in the following procedure.

IMPORT WINDOWS FIREWALL RULES

GET READY. Log on to a server running Windows Server 2012 R2, using an account with domain administrator privileges.

1. In the *Server Manager* console, select *Windows Firewall With Advanced Security* from the Tools menu. The *Windows Firewall with Advanced Security* console appears.
2. From the *Action* menu, select Import Policy. A message box appears, warning that importing a policy file will overwrite all existing firewall rules.
3. Click Yes. An *Open* combo box appears.
4. Locate and select the policy file you want to import and click Open.

5. A message box appears, stating that the policy was successfully imported.
6. Click OK.

CLOSE the Windows Firewall With Advanced Security console.

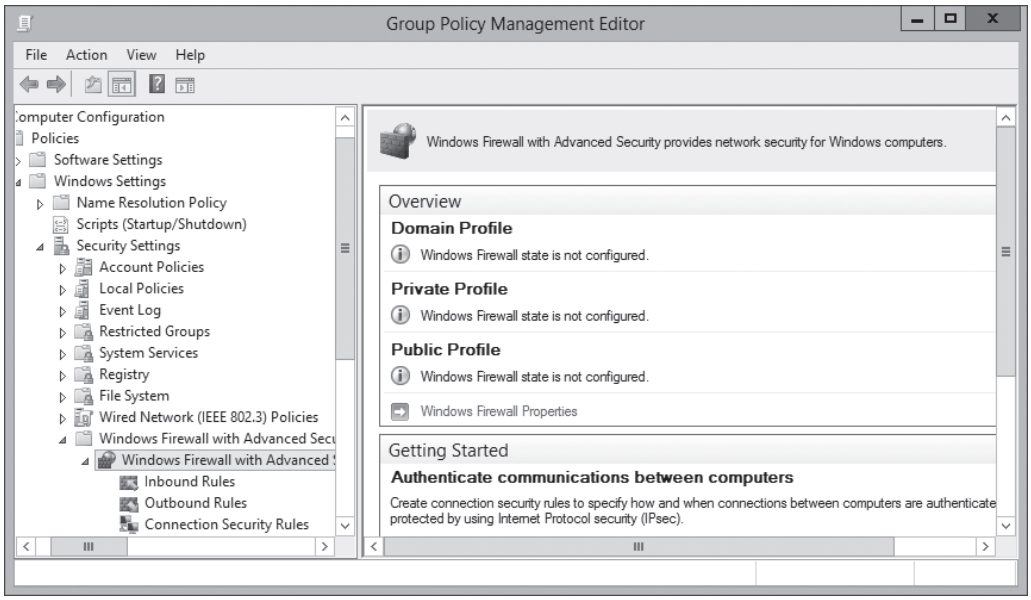
Creating Rules Using Group Policy

The Windows Firewall with Advanced Security console makes it possible to create complex firewall configurations, but Windows Firewall is still an application designed to protect a single computer from intrusion. If you have a large number of servers running Windows Server 2012 R2, manually creating a complex firewall configuration on each one can be a lengthy process. Therefore, as with most Windows configuration tasks, administrators can distribute firewall settings to computers throughout the network by using Group Policy.

When you edit a Group Policy object (GPO) and browse to the *Computer Configuration\Policies\Windows Settings\Security Settings\ Windows Firewall with Advanced Security* node, you see an interface, shown in Figure 19-6, which is similar to the Windows Firewall with Advanced Security console.

Figure 19-6

The *Windows Firewall with Advanced Security* node in a Group Policy object



You can configure Windows Firewall properties and create inbound, outbound, and connection security rules, just as you would in the console. The difference is that you can then deploy these settings to computers anywhere on the network by linking the GPO to Active Directory Domain Services object.

When you open a new GPO, the Windows Firewall with Advanced Security node contains no rules at all. The preconfigured rules that you find on every computer running Windows Server 2012 R2 are not there. You can create new rules from scratch to deploy to the network, or you can import settings from a policy file, just as you can in the Windows Firewall with Advanced Security console.

Group Policy does not overwrite the entire Windows Firewall configuration, as importing a policy file does. When you deploy firewall rules and settings by using Group Policy, the rules in the GPO are combined with the existing rules on the target computers. The only exception is when you deploy rules with the same names as existing rules. Then, the GPO settings overwrite these found on the target computers.

Creating Connection Security Rules

Windows Server 2012 R2 also includes a feature that incorporates IPsec data protection into the Windows Firewall. The IP Security (IPsec) standards are a collection of documents that define a method for securing data while it is in transit over a TCP/IP network. IPsec includes a connection establishment routine, during which computers authenticate each other before transmitting data, and a technique called ***tunneling***, in which data packets are encapsulated within other packets, for their protection.

In addition to inbound and outbound rules, the Windows Firewall With Advanced Security console enables you to create connection security rules, by using the New Connection Security Rule Wizard. Connection security rules define the type of protection you want to apply to the communications that conform to Windows Firewall rules.

When you right-click the *Connection Security Rules* node and select *New Rule* from the context menu, the *New Connection Security Rule Wizard* takes you through the process of configuring the following sets of parameters:

- **Rule Type:** specifies the basic function of the rule, such as to isolate computers based on authentication criteria, to exempt certain computers (such as infrastructure servers) from authentication, to authenticate two specific computers or groups of computers, or to tunnel communications between two computers. You can also create custom rules combining these functions.
- **Endpoints:** specifies the IP addresses of the computers that establish a secured connection before transmitting any data.
- **Requirements:** specifies whether authentication between two computers should be requested or required. If required, options include requiring authentication for inbound connections only or for both inbound and outbound connections.
- **Authentication Method:** specifies the type of authentication the computers should use when establishing a connection.
- **Profile:** specifies the profile(s) to which the rule should apply: domain, private, and public.
- **Name:** specifies a name and (optionally) a description for the rule.

■ Business Case Scenarios

Scenario 19-1: Configuring Windows Firewall

Ralph is a junior network administrator at Wingtip Toys, left in change of the IP department while everyone else is out of town at a conference. Ralph receives a call from the company's best customer, reporting that they are unable to place orders through the company's website. Ralph examines the logs for the Windows web server and notices a huge amount of incoming traffic that began that morning.

Ralph suspects that the server is the target of a denial of service (DoS) attack, but he doesn't have access to the network firewall, nor does he know anything about the firewall configuration his company uses. Ralph does have access to the Windows Firewall running on the web server, however. What temporary modifications can he make to that firewall to block the attack and allow the customer to submit orders as they normally do?

Scenario 19-2: Configuring Firewall Rules to Require IPsec Encryption

Alice is responsible for the file server where all of the company payroll and accounting data is stored, and as a result, security is one of her primary concerns. To prevent potential intruders from intercepting sensitive data in transit over the network, Alice wants to ensure that all of the company bookkeepers and accountants use IPsec to encrypt their traffic as they access the financial spreadsheets stored on the server.

Alice has located the inbound firewall rules on the server that enable users to access files using their spreadsheet application. How can she modify those rules to permit access only under the following conditions:

- The user must be a member of the Accounting group.
- The user must be using a computer on the company subnet.
- The user must connect using IPsec encryption.