

Contents at a Glance

1	Installation	1
2	Administration	19
3	Users.....	50
4	Groups.....	87
5	Computers.....	121
6	Group Policy Infrastructure	147
7	Group Policy Settings	189
8	Authentication	228
9	Integrating Domain Name System with AD DS.....	251
10	Domain Controllers	298
11	Sites and Replication	328
12	Domains and Forests.....	361

Chapter 1

Installation

Active Directory Domain Services (AD DS) and its related services form the foundation for enterprise networks running Microsoft Windows as, together, they act as tools to store information about the identities of users, computers, and services; to authenticate a user or computer; and to provide a mechanism with which the user or computer can access resources in the enterprise. In this chapter, you will begin your exploration of Windows Server 2008 Active Directory by installing the Active Directory Domain Services role and creating a domain controller in a new Active Directory forest. You will find that Windows Server 2008 continues the evolution of Active Directory by enhancing many of the concepts and features with which you are familiar from your experience with Active Directory.

This chapter focuses on the creation of a new Active Directory forest with a single domain in a single domain controller. The practice exercises in this chapter will guide you through the creation of a domain named *contoso.com* that you will use for all other practices in this training kit. Later, in Chapter 8, “Authentication,” Chapter 10, “Domain Controllers,” and Chapter 12, “Domains and Forests,” you will learn to implement other scenarios, including multidomain forests, upgrades of existing forests to Windows Server 2008, and advanced installation options. In Chapter 14, “Active Directory Lightweight Directory Services,” Chapter 15, “Active Directory Certificate Services and Public Key Infrastructures,” Chapter 16, “Active Directory Rights Management Services,” and Chapter 17, “Active Directory Federation Services,” you will learn the details of other Active Directory services such as Active Directory Lightweight Directory Services, Active Directory Certificate Services and public key infrastructure, Active Directory Rights Management Service, and Active Directory Federated Services.

Exam objectives in this chapter:

- Configuring the Active Directory Infrastructure
 - Configure a forest or a domain.

Before You Begin

To complete the lessons in this chapter, you must have done the following:

- Obtained two computers on which you will install Windows Server 2008. The computers can be physical systems that meet the minimum hardware requirements for Windows Server 2008 found at <http://technet.microsoft.com/en-us/windowsserver/2008/bb414778.aspx>. You will need at least 512 MB of RAM, 10 GB of free hard disk space, and an x86 processor with a minimum clock speed of 1GHz or an x64 processor with a minimum clock speed of 1.4 GHz. Alternatively, you can use virtual machines that meet the same requirements.
- Obtained an evaluation version of Windows Server 2008. At the time of writing, links to evaluation versions are available on the Windows Server 2008 Home Page at <http://www.microsoft.com/windowsserver2008>.

Real World

Dan Holme

Domain controllers perform identity and access management functions that are critical to the integrity and security of a Windows enterprise. Therefore, most organizations choose to dedicate the role of domain controller, meaning that a domain controller does not provide other functions such as file and print servers. In previous versions of Windows, however, when you promote a server to a domain controller, other services continue to be available whether or not they are in use. These additional unnecessary services increase the need to apply patches and security updates and expose the domain controller to additional susceptibility to attack. Windows Server 2008 addresses these concerns through its role-based architecture, so that a server begins its life as a fairly lean installation of Windows to which roles and their associated services and features are added. Additionally, the new Server Core installation of Windows Server 2008 provides a minimal installation of Windows that even forgoes a graphical user interface (GUI) in favor of a command prompt. In this chapter, you will gain firsthand experience with these important characteristics of Windows Server 2008 domain controllers. These changes to the architecture and feature set of Windows Server 2008 domain controllers will help you and other enterprises further improve the security, stability, and manageability of your identity and access management infrastructure.

Lesson 1: Installing Active Directory Domain Services

Active Directory Domain Services (AD DS) provides the functionality of an identity and access (IDA) solution for enterprise networks. In this lesson, you will learn about AD DS and other Active Directory roles supported by Windows Server 2008. You will also explore Server Manager, the tool with which you can configure server roles, and the improved Active Directory Domain Services Installation Wizard. This lesson also reviews key concepts of IDA and Active Directory.

After this lesson, you will be able to:

- Explain the role of identity and access in an enterprise network.
- Understand the relationship between Active Directory services.
- Configure a domain controller with the Active Directory Domain Services (AD DS) role, using the Windows interface.

Estimated lesson time: 60 minutes

Active Directory, Identity and Access

As mentioned in the introductions to the chapter and this lesson, Active Directory provides the IDA solution for enterprise networks running Windows. IDA is necessary to maintain the security of enterprise resources such as files, e-mail, applications, and databases. An IDA infrastructure should do the following:

- **Store information about users, groups, computers, and other identities** An identity is, in the broadest sense, a representation of an entity that will perform actions on the enterprise network. For example, a user will open documents from a shared folder on a server. The document will be secured with permissions on an access control list (ACL). Access to the document is managed by the security subsystem of the server, which compares the identity of the user to the identities on the ACL to determine whether the user's request for access will be granted or denied. Computers, groups, services, and other objects also perform actions on the network, and they must be represented by identities. Among the information stored about an identity are properties that uniquely identify the object, such as a user name or a security identifier (SID), and the password for the identity. The *identity store* is, therefore, one component of an IDA infrastructure. The Active Directory data store, also known as the directory, is an identity store. The directory itself is hosted on and managed by a domain controller—a server performing the AD DS role.

- **Authenticate an identity** The server will not grant the user access to the document unless the server can verify the identity presented in the access request as valid. To validate the identity, the user provides secrets known only to the user and the IDA infrastructure. Those secrets are compared to the information in the identity store in a process called *authentication*.

Kerberos Authentication in an Active Directory Domain

In an Active Directory domain, a protocol called Kerberos is used to authenticate identities. When a user or computer logs on to the domain, Kerberos authenticates its credentials and issues a package of information called a ticket granting ticket (TGT). Before the user connects to the server to request the document, a Kerberos request is sent to a domain controller along with the TGT that identifies the authenticated user. The domain controller issues the user another package of information called a service ticket that identifies the authenticated user to the server. The user presents the service ticket to the server, which accepts the service ticket as proof that the user has been authenticated.

These Kerberos transactions result in a single network logon. After the user or computer has initially logged on and has been granted a TGT, the user is authenticated within the entire domain and can be granted service tickets that identify the user to any service. All of this ticket activity is managed by the Kerberos clients and services built into Windows and is transparent to the user.

- **Control access** The IDA infrastructure is responsible for protecting confidential information such as the information stored in the document. Access to confidential information must be managed according to the policies of the enterprise. The ACL on the document reflects a security policy composed of permissions that specify access levels for particular identities. The security subsystem of the server in this example is performing the access control functionality in the IDA infrastructure.
- **Provide an audit trail** An enterprise might want to monitor changes to and activities within the IDA infrastructure, so it must provide a mechanism by which to manage auditing.

AD DS is not the only component of IDA that is supported by Windows Server 2008. With the release of Windows Server 2008, Microsoft has consolidated a number of previously separate components into an integrated IDA platform. Active Directory itself now includes five technologies, each of which can be identified with a keyword that identifies the purpose of the technology, as shown in Figure 1-1.

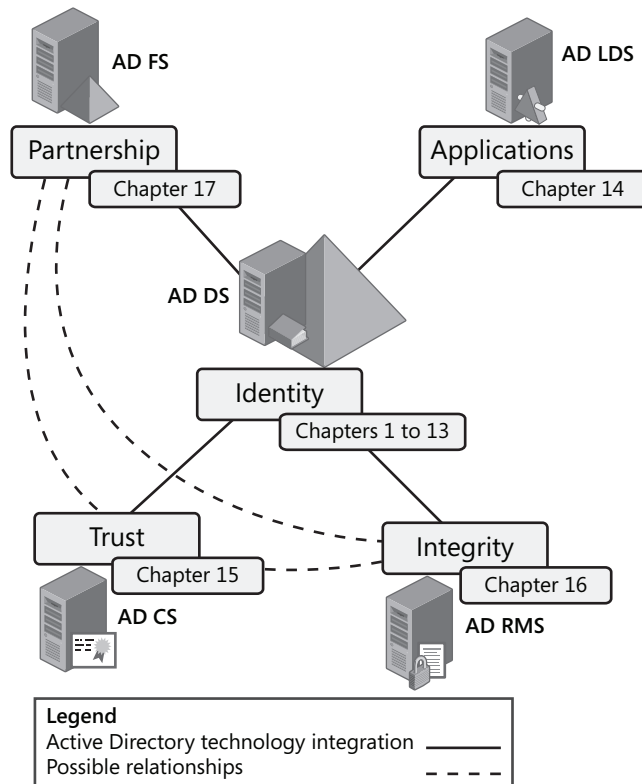


Figure 1-1 The integration of the five Active Directory technologies

These five technologies comprise a complete IDA solution:

- Active Directory Domain Services (Identity)** AD DS, as described earlier, is designed to provide a central repository for identity management within an organization. AD DS provides authentication and authorization services in a network and supports object management through Group Policy. AD DS also provides information management and sharing services, enabling users to find any component—file servers, printers, groups, and other users—by searching the directory. Because of this, AD DS is often referred to as a network operating system directory service. AD DS is the primary Active Directory technology and should be deployed in every network that runs Windows Server 2008 operating systems. AD DS is covered in chapters 1 through 13.

For a guide outlining best practices for the design of Active Directory, download the free “Chapter 3: Designing the Active Directory” from *Windows Server 2003, Best Practices for Enterprise Deployments* at http://www.reso-net.com/Documents/007222343X_Ch03.pdf.

MORE INFO AD DS design

For updated information on creating an Active Directory Domain Services design, look up *Windows Server 2008: The Complete Reference*, by Ruest and Ruest (McGraw-Hill Osborne, in press).

- **Active Directory Lightweight Directory Services (Applications)** Essentially a standalone version of Active Directory, the Active Directory Lightweight Directory Services (AD LDS) role, formerly known as Active Directory Application Mode (ADAM), provides support for directory-enabled applications. AD LDS is really a subset of AD DS because both are based on the same core code. The AD LDS directory stores and replicates only application-related information. It is commonly used by applications that require a directory store but do not require the information to be replicated as widely as to all domain controllers. AD LDS also enables you to deploy a custom schema to support an application without modifying the schema of AD DS. The AD LDS role is truly lightweight and supports multiple data stores on a single system, so each application can be deployed with its own directory, schema, assigned Lightweight Directory Access Protocol (LDAP) and SSL ports, and application event log. AD LDS does not rely on AD DS, so it can be used in a standalone or workgroup environment. However, in domain environments, AD LDS can use AD DS for the authentication of Windows security principals (users, groups, and computers). AD LDS can also be used to provide authentication services in exposed networks such as extranets. Once again, using AD LDS in this situation provides less risk than using AD DS. AD LDS is covered in Chapter 14.
- **Active Directory Certificate Services (Trust)** Organizations can use Active Directory Certificate Services (AD CS) to set up a certificate authority for issuing digital certificates as part of a public key infrastructure (PKI) that binds the identity of a person, device, or service to a corresponding private key. Certificates can be used to authenticate users and computers, provide Web-based authentication, support smart card authentication, and support applications, including secure wireless networks, virtual private networks (VPNs), Internet Protocol security (IPSec), Encrypting File System (EFS), digital signatures, and more. AD CS provides an efficient and secure way to issue and manage certificates. You can use AD CS to provide these services to external communities. If you do so, AD CS should be linked with an external, renowned CA that will prove to others you are who you say you are. AD CS is designed to create trust in an untrustworthy world; as such, it must rely on proven processes that certify that each person or computer that obtains a certificate has been thoroughly verified and approved. In internal networks, AD CS can integrate with AD DS to provision users and computers automatically with certificates. AD CS is covered in Chapter 15.

For more information on PKI infrastructures and how to apply them in your organization, visit <http://www.reso-net.com/articles.asp?m=8> and look for the “Advanced Public Key Infrastructures” section.

- **Active Directory Rights Management Services (Integrity)** Although a server running Windows can prevent or allow access to a document based on the document's ACL, there have been few ways to control what happens to the document and its content after a user has opened it. Active Directory Rights Management Services (AD RMS) is an information-protection technology that enables you to implement persistent usage policy templates that define allowed and unauthorized use whether online, offline, inside, or outside the firewall. For example, you could configure a template that allows users to read a document but not to print or copy its contents. By doing so, you can ensure the integrity of the data you generate, protect intellectual property, and control who can do what with the documents your organization produces. AD RMS requires an Active Directory domain with domain controllers running Windows 2000 Server with Service Pack 3 (SP3) or later; IIS; a database server such as Microsoft SQL Server 2008; the AD RMS client that can be downloaded from the Microsoft Download Center and is included by default in Windows Vista and Windows Server 2008; and an RMS-enabled browser or application such as Microsoft Internet Explorer, Microsoft Office, Microsoft Word, Microsoft Outlook, or Microsoft PowerPoint. AD RMS can rely on AD CS to embed certificates within documents as well as in AD DS to manage access rights. AD RMS is covered in Chapter 16.
- **Active Directory Federation Services (Partnership)** Active Directory Federation Services (AD FS) enables an organization to extend IDA across multiple platforms, including both Windows and non-Windows environments, and to project identity and access rights across security boundaries to trusted partners. In a federated environment, each organization maintains and manages its own identities, but each organization can also securely project and accept identities from other organizations. Users are authenticated in one network but can access resources in another—a process known as single sign-on (SSO). AD FS supports partnerships because it allows different organizations to share access to extranet applications while relying on their own internal AD DS structures to provide the actual authentication process. To do so, AD FS extends your internal AD DS structure to the external world through common Transmission Control Protocol/Internet Protocol (TCP/IP) ports such as 80 (HTTP) and 443 (Secure HTTP, or HTTPS). It normally resides in the perimeter network. AD FS can rely on AD CS to create trusted servers and on AD RMS to provide external protection for intellectual property. AD FS is covered in Chapter 17.

Together, the Active Directory roles provide an integrated IDA solution. AD DS or AD LDS provides foundational directory services in both domain and standalone implementations. AD CS provides trusted credentials in the form of PKI digital certificates. AD RMS protects the integrity of information contained in documents. And AD FS supports partnerships by eliminating the need for federated environments to create multiple, separate identities for a single security principal.

Beyond Identity and Access

Active Directory delivers more than just an IDA solution, however. It also provides the mechanisms to support, manage, and configure resources in distributed network environments.

A set of rules, the *schema*, defines the classes of objects and attributes that can be contained in the directory. The fact that Active Directory has user objects that include a user name and password, for example, is because the schema defines the *user* object class, the two attributes, and the association between the object class and attributes.

Policy-based administration eases the management burden of even the largest, most complex networks by providing a single point at which to configure settings that are then deployed to multiple systems. You will learn about such policies, including Group Policy, audit policies, and fine-grained password policies in Chapter 6, “Group Policy Infrastructure,” Chapter 7, “Group Policy Settings,” and Chapter 8.

Replication services distribute directory data across a network. This includes both the data store itself as well as data required to implement policies and configuration, including logon scripts. In Chapter 8, Chapter 11, “Sites and Replication,” and Chapter 10, you will learn about Active Directory replication. There is even a separate partition of the data store named *configuration* that maintains information about network configuration, topology, and services.

Several components and technologies enable you to query Active Directory and locate objects in the data store. A partition of the data store called the *global catalog* (also known as the *partial attribute set*) contains information about every object in the directory. It is a type of index that can be used to locate objects in the directory. Programmatic interfaces such as Active Directory Services Interface (ADSI) and protocols such as LDAP can be used to read and manipulate the data store.

The Active Directory data store can also be used to support applications and services not directly related to AD DS. Within the database, application partitions can store data to support applications that require replicated data. The domain name system (DNS) service on a server running Windows Server 2008 can store its information in a database called an Active Directory integrated zone, which is maintained as an application partition in AD DS and replicated using Active Directory replication services.

Components of an Active Directory Infrastructure

The first 13 chapters of this training kit will focus on the installation, configuration, and management of AD DS. AD DS provides the foundation for IDA in and management of an enterprise network. It is worthwhile to spend a few moments reviewing the components of an Active Directory infrastructure.

NOTE Where to find Active Directory details

For more details about Active Directory, refer to the product help installed with Windows Server 2008 and to the TechCenter for Windows Server 2008 located at <http://technet.microsoft.com/en-us/windowsserver/2008/default.aspx>.

- **Active Directory data store** As mentioned in the previous section, AD DS stores its identities in the directory—a data store hosted on domain controllers. The directory is a single file named `Ntds.dit` and is located by default in the `%SystemRoot%\Ntds` folder on a domain controller. The database is divided into several partitions, including the schema, configuration, global catalog, and the domain naming context that contains the data about objects within a domain—the users, groups, and computers, for example.
- **Domain controllers** Domain controllers, also referred to as DCs, are servers that perform the AD DS role. As part of that role, they also run the Kerberos Key Distribution Center (KDC) service, which performs authentication, and other Active Directory services. Chapter 10 details the roles performed by DCs.
- **Domain** One or more domain controllers are required to create an Active Directory *domain*. A domain is an administrative unit within which certain capabilities and characteristics are shared. First, all domain controllers replicate the domain's partition of the data store, which contains among other things the identity data for the domain's users, groups, and computers. Because all DCs maintain the same identity store, any DC can authenticate any identity in a domain. Additionally, a domain is a scope of administrative policies such as password complexity and account lockout policies. Such policies configured in one domain affect all accounts in the domain and do not affect accounts in other domains. Changes can be made to objects in the Active Directory database by any domain controller and will replicate to all other domain controllers. Therefore, in networks where replication of all data between domain controllers cannot be supported, it might be necessary to implement more than one domain to manage the replication of subsets of identities. You will learn more about domains in Chapter 12.
- **Forest** A *forest* is a collection of one or more Active Directory domains. The first domain installed in a forest is called the *forest root domain*. A forest contains a single definition of network configuration and a single instance of the directory schema. A forest is a single instance of the directory—no data is replicated by Active Directory outside the boundaries of the forest. Therefore, the forest defines a security boundary. Chapter 12 will explore the concept of the forest further.
- **Tree** The DNS namespace of domains in a forest creates trees within the forest. If a domain is a subdomain of another domain, the two domains are considered a tree. For example, if the *treyresearch.net* forest contains two domains, *treyresearch.net* and *antarctica.treyresearch.net*, those domains constitute a contiguous portion of the DNS namespace, so they are a single tree. If, conversely, the two domains are *treyresearch.net*

and *proseware.com*, which are not contiguous in the DNS namespace, the domain is considered to have two trees. Trees are the direct result of the DNS names chosen for domains in the forest.

Figure 1-2 illustrates an Active Directory forest for Trey Research, which maintains a small operation at a field station in Antarctica. Because the link from Antarctica to the headquarters is expensive, slow, and unreliable, Antarctica is configured as a separate domain. The DNS name of the forest is *treyresearch.net*. The Antarctica domain is a child domain in the DNS namespace, *antarctica.treyresearch.net*, so it is considered a child domain in the domain tree.

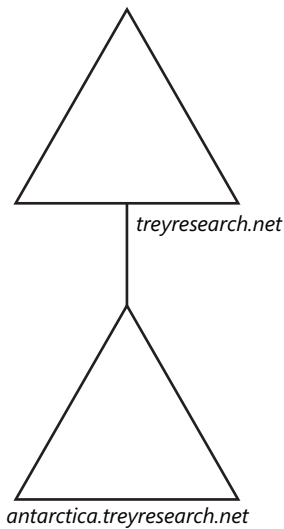


Figure 1-2 An Active Directory forest with two domains

- **Functional level** The functionality available in an Active Directory domain or forest depends on its *functional level*. The functional level is an AD DS setting that enables advanced domain-wide or forest-wide AD DS features. There are three domain functional levels, Windows 2000 native, Windows Server 2003, and Windows Server 2008 and two forest functional levels, Microsoft Windows Server 2003 and Windows Server 2008. As you raise the functional level of a domain or forest, features provided by that version of Windows become available to AD DS. For example, when the domain functional level is raised to Windows Server 2008, a new attribute becomes available that reveals the last time a user successfully logged on to a computer, the computer to which the user last logged on, and the number of failed logon attempts since the last logon. The important thing to know about functional levels is that they determine the versions of Windows permitted on domain controllers. Before you raise the domain functional level to Windows Server 2008, all domain controllers must be running Windows Server 2008. Chapter 12, details domain and forest functional levels.

- **Organizational units** Active Directory is a hierarchical database. Objects in the data store can be collected in containers. One type of container is the object class called *container*. You have seen the default containers, including Users, Computers, and Builtin, when you open the Active Directory Users and Computers snap-in. Another type of container is the organizational unit (OU). OUs provide not only a container for objects but also a scope with which to manage the objects. That is because OUs can have objects called Group Policy objects (GPOs) linked to them. GPOs can contain configuration settings that will then be applied automatically by users or computers in an OU. In Chapter 2, “Administration,” you will learn more about OUs, and in Chapter 6, you will explore GPOs.
- **Sites** When you consider the network topology of a distributed enterprise, you will certainly discuss the network’s sites. Sites in Active Directory, however, have a very specific meaning because there is a specific object class called *site*. An Active Directory site is an object that represents a portion of the enterprise within which network connectivity is good. A site creates a boundary of replication and service usage. Domain controllers within a site replicate changes within seconds. Changes are replicated between sites on a controlled basis with the assumption that intersite connections are slow, expensive, or unreliable compared to the connections within a site. Additionally, clients will prefer to use distributed services provided by servers in their site or in the closest site. For example, when a user logs on to the domain, the Windows client first attempts to authenticate with a domain controller in its site. Only if no domain controller is available in the site will the client attempt to authenticate with a DC in another site. Chapter 11 details the configuration and functionality of Active Directory sites.

Each of these components is discussed in detail later in this training kit. At this point, if you are less familiar with Active Directory, it is important only that you have a basic understanding of the terminology, the components, and their relationships.

Preparing to Create a New Windows Server 2008 Forest

Before you install the AD DS role on a server and promote it to act as a domain controller, plan your Active Directory infrastructure. Some of the information you will need to create a domain controller includes the following:

- The domain’s name and DNS name. A domain must have a unique DNS name, for example, *contoso.com*, as well as a short name, for example, CONTOSO, called a NetBIOS name. NetBIOS is a network protocol that has been used since the first versions of Microsoft Windows NT and is still used by some applications.
- Whether the domain will need to support domain controllers running previous versions of Windows. When you create a new Active Directory forest, you will configure the functional level. If the domain will include only Windows Server 2008 domain controllers,

you can set the functional level accordingly to benefit from the enhanced features introduced by this version of Windows.

- Details for how DNS will be implemented to support Active Directory. It is a best practice to implement DNS for your Windows domain zones by using Windows DNS Service, as you will learn in Chapter 9, “Integrating Domain Name System with AD DS”; however, it is possible to support a Windows domain on a third-party DNS service.
- IP configuration for the domain controller. Domain controllers require static IP addresses and subnet mask values. Additionally, the domain controller must be configured with a DNS server address to perform name resolution. If you are creating a new forest and will run Windows DNS Service on the domain controller, you can configure the DNS address to point to the server’s own IP address. After DNS is installed, the server can look to itself to resolve DNS names.
- The user name and password of an account in the server’s Administrators group. The account must have a password—the password cannot be blank.
- The location in which the data store (including *Ntds.dit*) and system volume (SYSVOL) should be installed. By default, these stores are created in %SystemRoot%, for example, C:\Windows, in the NTDS and SYSVOL folders, respectively. When creating a domain controller, you can redirect these stores to other drives.

MORE INFO Deployment of AD DS

This list comprises the settings that you will be prompted to configure when creating a domain controller. There are a number of additional considerations regarding the deployment of AD DS in an enterprise setting. See the Windows Server 2008 Technical Library at <http://technet2.microsoft.com/windowsserver2008/en/library/bab0f1a1-54aa-4cef-9164-139e8bcc44751033.mspx> for more information.

Adding the AD DS Role Using the Windows Interface

After you have collected the prerequisite information listed earlier, you are ready to add the AD DS role. There are several ways to do so. In this lesson, you will learn how to create a domain controller by using the Windows interface. In the next lesson, you will learn to do so using the command line.

Windows Server 2008 provides role-based configuration, installing only the components and services required for the roles a server plays. This role-based server management is reflected in the new administrative console, Server Manager, shown in Figure 1-3. Server Manager consolidates the information, tools, and resources needed to support a server’s roles.

You can add roles to a server by using the Add Roles link on the home page of Server Manager or by right-clicking the Roles node in the console tree and choosing Add Roles. The Add Roles Wizard presents a list of roles available for installation and steps you through the installation of selected roles.

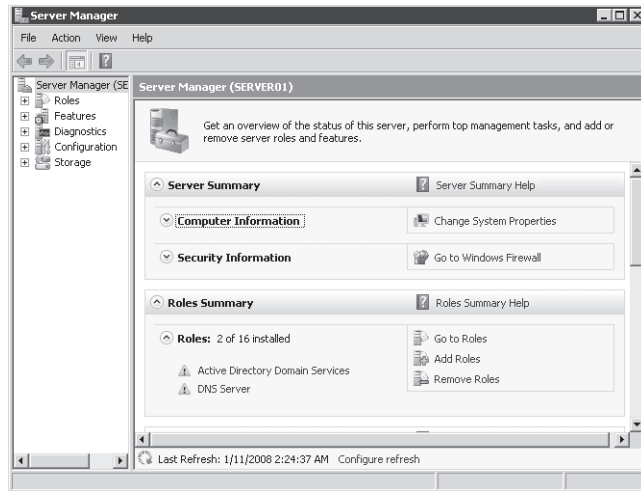


Figure 1-3 Server Manager

Practice It Exercise 3, “Install a New Windows Server 2008 Forest with the Windows Interface,” at the end of this lesson guides you through adding the AD DS role, using the Windows interface.

Creating a Domain Controller

After you add the AD DS role, the files required to perform the role are installed on the server; however, the server is not yet acting as a domain controller. You must subsequently run the Active Directory Domain Services Installation Wizard, which can be launched using the *Dcpromo.exe* command, to configure, initialize, and start Active Directory.

Practice It Exercise 4, “Install a New Windows Server 2008 Forest,” at the end of this lesson guides you through configuration of AD DS, using the Active Directory Domain Services Installation Wizard.

Quick Check

- You want to use a new server running Windows Server 2008 as a domain controller in your Active Directory domain. Which command do you use to launch configuration of the domain controller?

Quick Check Answer

- *Dcpromo.exe*

Lesson 2: Active Directory Domain Services on Server Core

Many organizations want to implement the maximum available security for servers acting as domain controllers because of the sensitive nature of information stored in the directory—particularly user passwords. Although the role-based configuration of Windows Server 2008 reduces the security surface of a server by installing only the components and services required by its roles, it is possible to reduce its servers and security surface further by installing Server Core. A Server Core installation is a minimal installation of Windows that forgoes even the Windows Explorer GUI and the Microsoft .NET Framework. You can administer a Server Core installation remotely, using GUI tools; however, to configure and manage a server locally, you must use command-line tools. In this lesson, you will learn to create a domain controller from the command line within a Server Core installation. You will also learn how to remove domain controllers from a domain.

After this lesson, you will be able to:

- Identify the benefits and functionality of installing Server Core.
- Install and configure Server Core.
- Add and remove Active Directory Domain Services (AD DS), using command-line tools.

Estimated lesson time: 60 minutes

Understanding Server Core

Windows Server 2008 (Server Core Installation), better known as Server Core, is a minimal installation of Windows that consumes about 3 GB of disk space and less than 256 MB of memory. Server Core installation limits the server roles and features that can be added but can improve the security and manageability of the server by reducing its attack surface. The number of services and components running at any one time are limited, so there are fewer opportunities for an intruder to compromise the server. Server Core also reduces the management burden of the server, which requires fewer updates and less maintenance.

Server Core supports nine server roles:

- Active Directory Domain Services
- Active Directory Lightweight Directory Services (AD LDS)
- Dynamic Host Configuration Protocol (DHCP) Server
- DNS Server
- File Services
- Print Server

- Streaming Media Services
- Web Server (IIS) (as a static Web server—ASP.NET cannot be installed)
- Hyper-V (Windows Server Virtualization)

Server core also supports these 11 optional features:

- Microsoft Failover Cluster
- Network Load Balancing
- Subsystem for UNIX-based applications
- Windows Backup
- Multipath I/O
- Removable Storage Management
- Windows Bitlocker Drive Encryption
- Simple Network Management Protocol (SNMP)
- Windows Internet Naming Service (WINS)
- Telnet client
- Quality of Service (QoS)

Installing Server Core

You can install Server Core by using the same steps presented in Exercise 1 of Lesson 1. The differences between a full installation and a Server Core installation are, first, that you select Server Core Installation in the Installing Windows Wizard shown in Figure 1-9, and that when the installation is complete and you log on, a command prompt appears rather than the Windows Explorer interface.

NOTE The blank initial Administrator password

When you install Windows Server 2008 from the installation DVD, the initial password for the Administrator account is blank. When you log on to the server for the first time, use a blank password. You will be prompted to change the password on first log on.

Practice It Exercise 1, “Install Server Core,” in the practice at the end of this lesson, steps you through the installation of Server Core.

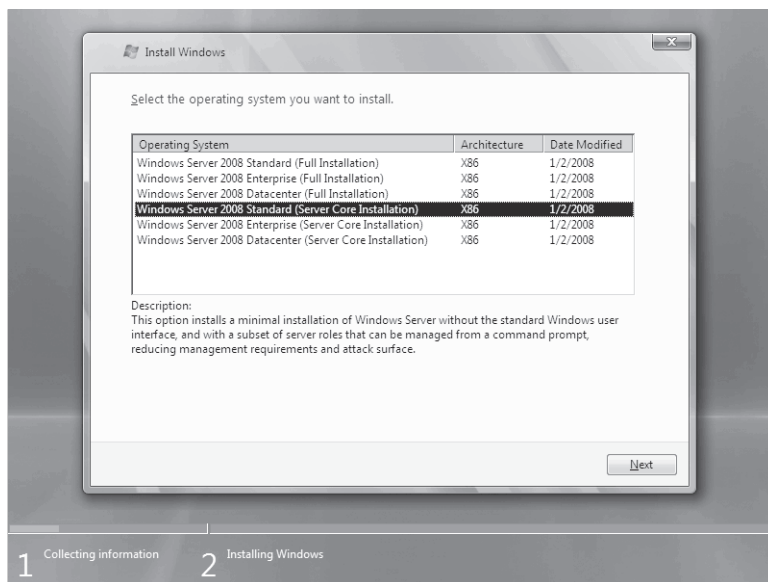


Figure 1-9 The Operating Systems selection page of the Install Windows Wizard

Performing Initial Configuration Tasks

On a full installation of Windows Server 2008, the Initial Configuration Tasks window appears to guide you through post-installation configuration of the server. Server Core provides no GUI, so you must complete the tasks by using command-line tools. Table 1-1 lists common configuration tasks and the commands you can use. To learn more about any command, open a command prompt and type the name of the command followed by `/?`.

Table 1-1 Server Core Configuration Commands

Task	Command
Change the Administrator password	When you log on with Ctrl + Alt + Del, you will be prompted to change the password. You can also type the following command: <i>Net user administrator *</i>
Set a static IPv4 configuration	<i>Netsh interface ipv4</i>
Activate Windows Server	<i>Cscript c:\windows\system32\slmgr.vbs -ato</i>
Join a domain	<i>Netdom</i>
Add Server Core roles, components, or features	<i>Ocsetup.exe</i> package or feature Note that the package or feature names are case sensitive.

Table 1-1 Server Core Configuration Commands

Task	Command
Display installed roles, components, and features	<i>Oclist.exe</i>
Enable Remote Desktop	<i>Cscript c:\windows\system32\scregedit.wsf /AR 0</i>
Promote a domain controller	<i>Dcpromo.exe</i>
Configure DNS	<i>Dnscmd.exe</i>
Configure DFS	<i>Dfscmd.exe</i>

Practice It Exercise 2, “Perform Post-Installation Configuration on Server Core,” in the practice at the end of this lesson, steps you through the initial configuration of a Server Core installation of Windows Server 2008.

The *Ocsetup.exe* command is used to add supported Server Core roles and features to the server. The exception to this rule is AD DS. Do not use *Ocsetup.exe* to add or remove AD DS. Use *Dcpromo.exe* instead.

Adding AD DS to a Server Core Installation

Because there is no Active Directory Domain Services Installation Wizard in Server Core, you must use the command line to run *Dcpromo.exe* with parameters that configure AD DS. To learn about the parameters of *Dcpromo.exe*, open a command line and type **dcpromo.exe /?**. Each configuration scenario has additional usage information. For example, type **dcpromo.exe /?:Promotion** for detailed usage instructions for promoting a domain controller.

MORE INFO Unattended installation parameters

You can find a listing of unattended installation parameters at <http://technet2.microsoft.com/windowsserver2008/en/library/bcd89659-402d-46fb-8535-8da1feb8d4111033.mspx>.

Practice It You will add AD DS to a Server Core installation during Exercise 3, “Create a Domain Controller with Server Core,” in the practice at the end of this lesson.

Removing Domain Controllers

Occasionally, you might have a reason to take a domain controller offline for extended maintenance or to remove it permanently. It is important that you remove a domain controller correctly so that the information about the domain controller is cleaned up in Active Directory.

To remove a domain controller, use the *Dcpromo.exe* command. If you run the command on a domain controller by using the Windows interface, the Active Directory Domain Services Installation Wizard will step you through the process. If you want to use the command line or are removing AD DS from a Server Core installation, type **dcpromo.exe /?:Demotion** for usage information regarding parameters for the demotion operation.

Practice It In Exercise 4, "Remove a Domain Controller," in the practice at the end of the lesson, you will remove a domain controller by using the *Dcpromo.exe* command.

When you demote a domain controller, you must provide a password that will be assigned to the local Administrator account of the server after demotion.

Chapter 2

Administration

Most administrators first experience Active Directory Domain Services (AD DS) by opening Active Directory Users And Computers and creating user, computer, or group objects within the organizational units (OUs) of a domain. Such tasks are fundamental to the job requirements of an IT professional in an Active Directory environment, so now that you have created a domain in Chapter 1, “Installation,” you can address the tools, tips, and best practices regarding the creation of these objects. Later chapters will explore each of these object classes in detail.

In this chapter, you will also look at two important, higher-level concerns within an enterprise: how to locate objects in the directory and how to ensure that Active Directory is secure while enabling support personnel to perform the tasks required of their roles.

Exam objectives in this chapter:

- Creating and Maintaining Active Directory Objects
 - Maintain Active Directory accounts

Before You Begin

To complete the lessons in this chapter, you must have installed Windows Server 2008 on a physical computer or virtual machine. The machine should be named SERVER01 and should be a domain controller in the *contoso.com* domain. The details for this setup are presented in Chapter 1.

Real World*Dan Holme*

You are certainly familiar with administrative tools, such as the Active Directory Users and Computers snap-in, and the basic skills required to create organizational units, users, computers, and groups. This chapter reviews those tools and skills so that you can fill in any gaps in your knowledge. More important, however, this chapter introduces ways you can elevate your productivity and effectiveness as an administrator. I find that many administrators continue to use the default consoles and, therefore, have to open multiple tools to do their jobs, instead of creating a single, customized Microsoft Management Console (MMC) that contains all the snap-ins they need. I also see administrators diving deep into their OU structure to locate and manage objects rather than taking advantage of the power of Saved Queries to virtualize the view of their domains. Although this chapter covers only one exam objective, “Maintain Active Directory accounts,” the tips and guidance I provide here is some of the most valuable in the book because it will enable you to work more efficiently and more securely every day in the real world of your enterprise.

Lesson 1: Working with Active Directory Snap-ins

The Active Directory administrative tools, or snap-ins, expose the functionality you require to support the directory service. In this lesson, you will identify and locate the most important Active Directory snap-ins. You will also learn how to work effectively with them, using alternate credentials, and how to build custom consoles that can be distributed to administrators in your organization.

After this lesson, you will be able to:

- Work with Microsoft Management Console.
- Identify the most important Active Directory administrative snap-ins.
- Install the Remote Server Administration Tools (RSAT) on Windows Server 2008 and Windows Vista.
- Launch administrative tools with alternate credentials, using Run As Administrator.
- Create, manage, and distribute a custom MMC.

Estimated lesson time: 35 minutes

Understanding the Microsoft Management Console

Windows administrative tools share a common framework called the Microsoft Management Console (MMC). The MMC displays tools in a customizable window with a left pane that displays the console tree (similar to the Windows Explorer tree) and a center pane that displays details. An Actions pane on the right exposes commands, called actions by MMC. Figure 2-1 shows an example.

To control the visibility of the left and right panes, use the Show/Hide Console Tree and Show/Hide Action Pane buttons or the *Customize* command on the View menu.

Administrative tools, called *snap-ins*, use the console tree and details pane of the console to provide administrative functionality. You can think of an MMC as a tool belt to which you can attach one or more tools (snap-ins). Snap-ins cannot be launched directly; they can function within the context of an MMC only. Most of the tools in the Administrative Tools folder constitute a single console with a single snap-in. These tools include Event Viewer, Services, and Task Scheduler. Other tools, such as Computer Management, are consoles that contain multiple snap-ins, including some that exist as standalone consoles. For example, the Computer Management console contains Event Viewer, Services, and Task Scheduler.

As you are administering Windows with snap-ins, you will be performing commands, called *actions* by the MMC, that you can find in the console's Action menu, on the context menu that appears when you right-click, and in the Actions pane on the right side of the console. Most experienced administrators find the context menu to be the most productive way to perform

actions in an MMC snap-in. If you use the context menu exclusively, you can turn off the Actions pane so that you have a larger area to display information in the details pane.

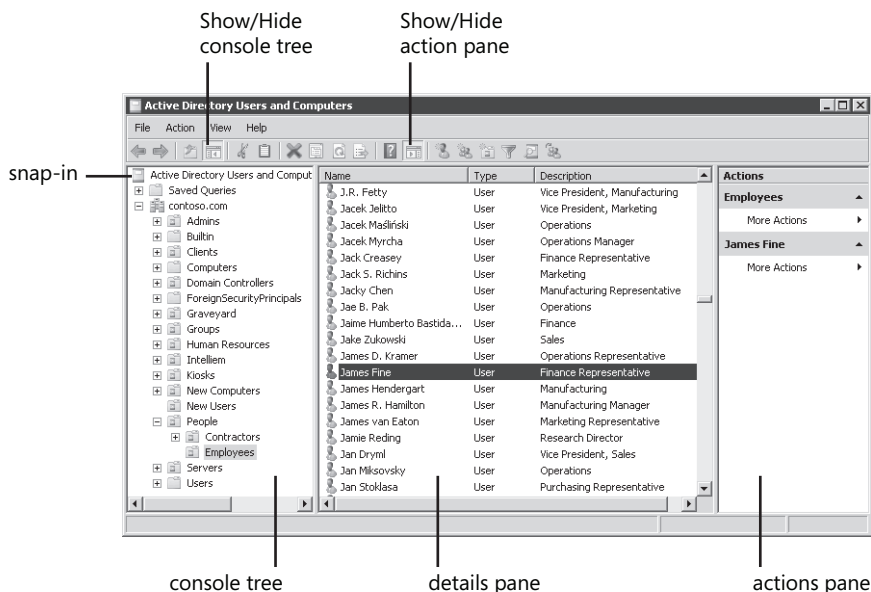


Figure 2-1 An MMC and snap-in

There are two types of MMC: preconfigured and custom. Preconfigured consoles are installed automatically when you add a role or feature, to support administration of that role or feature. They function in user mode, so you cannot modify them or save them. The user, however, can create custom consoles to provide exactly the tools and functionality required. In the following sections, you will look at both preconfigured and custom consoles.

Active Directory Administration Tools

Most Active Directory administration is performed with the following snap-ins and consoles:

- **Active Directory Users and Computers** Manage most common day-to-day resources, including users, groups, computers, printers, and shared folders. This is likely to be the most heavily used snap-in for an Active Directory administrator.
- **Active Directory Sites and Services** Manage replication, network topology, and related services. You will use this snap-in heavily in Chapter 11, “Sites and Replication.”
- **Active Directory Domains and Trusts** Configure and maintain trust relationships and the domain and forest functional levels. This tool will be discussed in Chapter 13, “Domains and Forests.”

- **Active Directory Schema** Examine and modify the definition of Active Directory attributes and object classes. This schema is the “blueprint” for Active Directory. It is rarely viewed and even more rarely changed. Therefore, the Active Directory Schema snap-in is not installed by default.

Active Directory snap-ins and consoles are installed when you add the AD DS role to a server. Two commonly used Active Directory administrative tools are added to Server Manager when you install the AD DS role: the Active Directory Users and Computers snap-in and the Active Directory Sites and Services snap-in. However, to administer Active Directory from a system that is not a domain controller, you must install the RSAT, a feature that can be installed from the Features node of Server Manager on Windows Server 2008. It can be downloaded from Microsoft and installed on clients running Windows Vista Service Pack 1.

Finding the Active Directory Administrative Tools

You can find two Active Directory snap-ins in Server Manager by expanding Roles and Active Directory Domain Services. All tools, however, can be found in the Administrative Tools folder, which itself is found in Control Panel. In the classic view of Control Panel, you will see the Administrative Tools folder displayed. Using the Control Panel Home view, you can find administrative tools in System And Maintenance.

Adding the Administrative Tools to Your Start Menu

By default, administrative tools are not added to the Start menu on Windows Vista clients. You can make the administrative tools easier to access by adding them to your Start menu.

1. Right-click the Start button and choose Properties.
2. Click Customize.
3. If you are using the default Start menu, scroll to System Administrative Tools and select Display On The All Programs Menu And The Start Menu or Display On The All Programs Menu. If you are using the Classic Start menu, select Display Administrative Tools.
4. Click OK twice.

Running Administrative Tools with Alternate Credentials

Many administrators log on to their computers by using their administrative accounts. This practice is dangerous because an administrative account has more privileges and access to more of the network than a standard user account. Therefore, malware that is launched with administrative credentials can cause significant damage. To avoid this problem, do not log on as an administrator. Instead, log on as a standard user and use the Run As Administrator feature to launch administrative tools in the security context of an administrative account:

1. Right-click the shortcut for an executable, Control Panel applet, or MMC that you want to launch, and then choose Run As Administrator. If you do not see the command, try holding down the Shift key and right-clicking.

The User Account Control dialog box appears, as shown in Figure 2-2.

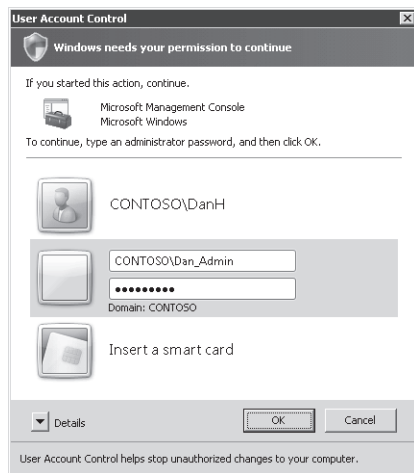


Figure 2-2 The User Account Control dialog box prompting for administrative credentials

2. Enter the user name and password of your administrative account.
3. Click OK.

If you will be running an application regularly as an administrator, create a new shortcut that preconfigures Run As Administrator. Create a shortcut and open the Properties dialog box for the shortcut. Click the Advanced button and select Run As Administrator. When you launch the shortcut, the User Account Control dialog box will appear.

Creating a Custom Console with Active Directory Snap-ins

It's easier to administer Windows when the tools you need are in one place and can be customized to meet your needs. You can achieve this by creating a custom administrative MMC which, continuing our tool belt metaphor, is a tool belt made just for you. When you create a custom MMC, you can:

- Add multiple snap-ins so that you do not have to switch between consoles to perform your job tasks and so that you have to launch only one console with Run As Administrator.
- Save the console to be used regularly.
- Distribute the console to other administrators.
- Centralize consoles in a shared location for unified, customized administration.

To create a custom MMC, open an empty MMC by clicking the Start button. Then, in the Start Search box, type **mmc.exe** and press Enter. The *Add/Remove Snap-in* command in the File menu enables you to add, remove, reorder, and manage the console's snap-ins.

Practice It Exercise 1, "Create a Custom MMC," Exercise 2, "Add a Snap-in to an MMC," and Exercise 3, "Manage the Snap-ins of an MMC," in the practice at the end of this lesson step you through the skills related to creating a custom MMC with multiple snap-ins.

Saving and Distributing a Custom Console

If you plan to distribute a console, it is recommended to save the console in user mode. To change a console's mode, choose Options from the File menu. By default, new consoles are saved in author mode, which enables adding and removing snap-ins, viewing all portions of the console tree, and saving customizations. User mode, by contrast, restricts the functionality of the console so that it cannot be changed. There are three types of user modes, described in Table 2-1. User Mode – Full Access is commonly selected for a console provided to skilled administrators with diverse job tasks requiring broad use of the console snap-ins. User Mode – Limited Access (multiple window and single window) is a locked-down mode and is, therefore, selected for a console provided to administrators with a more narrow set of job tasks.

Table 2-1 MMC Console Modes

Mode	Use when
Author	You want to continue customizing the console.
User Mode – Full Access	You want users of the console to be able to navigate between and use all snap-ins. Users will not be able to add or remove snap-ins or change the properties of snap-ins or the console.
User Mode – Limited Access, multiple window	You want users to navigate to and use only the snap-ins that you have made visible in the console tree, and you want to preconfigure multiple windows that focus on specific snap-ins. Users will not be able to open new windows.
User Mode – Limited Access, single window	You want users to navigate to and use only the snap-ins that you have made visible in the console tree within a single window.

After a console is no longer saved in author mode, you—the original author—can make changes to the console by right-clicking the saved console and choosing Author.

Practice It Exercise 4, "Prepare a Console for Distribution to Users," in the practice at the end of the lesson, guides you through saving a console in user mode so that it can be locked down for deployment to other administrators.

Consoles are saved with the .msc file extension. The default location to which consoles are saved is the Administrative Tools folder, but not the folder in Control Panel. Rather, they are saved in the Start menu folder of your user profile: %userprofile%\AppData\Roaming\Microsoft\Windows\StartMenu.

This location is problematic because it is secured with permissions so that only your user account has access to the console. The best practice is to log on to your computer with an account that is not privileged and then run administrative tools such as your custom console with alternate credentials that have sufficient privilege to perform administrative tasks. Because two accounts will be involved, saving the console to the Start menu subfolder of one account's user profile will mean additional navigation, at a minimum, and access-denied errors in a worst-case scenario.

Save your consoles to a location that can be accessed by both your user and your administrative credentials. It is recommended to save consoles to a shared folder on the network so that you can access your tools when you are logged on to other computers. Optionally, the folder can be made accessible by other administrators to create a centralized store of customized consoles. You can also save consoles to a portable device such as a USB drive, or you can even send a console as an e-mail attachment.

It is important to remember that consoles are basically a set of instructions that are interpreted by *mmc.exe*—instructions that specify which snap-ins to add and which computers to manage with those snap-ins. Consoles do not contain the snap-ins themselves. Therefore, a console will not function properly if the snap-ins it contains have not been installed, so be sure you have installed appropriate snap-ins from RSAT on systems on which you will use the console.

Quick Check

- Describe the difference between a console saved in user mode and in author mode.

Quick Check Answer

- Author mode enables a user to add and remove snap-ins and thoroughly customize the console. User mode prevents users from making changes to the console.

Lesson 2: Creating Objects in Active Directory

Active Directory is a directory service, and it is the role of a directory service to maintain information about enterprise resources, including users, groups, and computers. Resources are divided into OUs to facilitate manageability and visibility—that is, they can make it easier to find objects. In this lesson, you will learn how to create OUs, users, groups, and computers. You will also learn important skills to help you locate and find objects when you need them.

If you are experienced with Active Directory, you will be able to review the first few sections in this lesson quickly, but you might want to pay particular attention to the later sections, beginning with “Find Objects in Active Directory,” because they will help you make better use of Active Directory tools.

The practice exercises at the end of this lesson will be important for you to complete because they create some of the objects that will be used in future practices.

After this lesson, you will be able to:

- Create users, groups, computers, and organizational units.
- Disable protection to delete an organizational unit.
- Customize and take advantage of views and features of the Active Directory Users and Computers snap-in to work effectively with objects in the directory.
- Create saved queries to provide rule-based views of objects in the directory.

Estimated lesson time: 45 minutes

Creating an Organizational Unit

Organizational units (OUs) are administrative containers within Active Directory that are used to collect objects that share common requirements for administration, configuration, or visibility. What this means will become clearer as you learn more about OU design and management. For now, just understand that OUs provide an administrative hierarchy similar to the folder hierarchy of a disk drive: OUs create *collections* of objects that belong together for *administration*. The term *administration* is emphasized here because OUs are not used to assign permissions to resources—that is what groups are for. Users are placed into groups that are given permission to resources. OUs are administrative containers within which those users and groups can be managed by administrators.

To create an organizational unit:

1. Open the Active Directory Users And Computers snap-in.
2. Right-click the Domain node or the OU node in which you want to add the new OU, choose New, and then select Organizational Unit.

3. Type the name of the organizational unit.
Be sure to follow the naming conventions of your organization.
4. Select Protect Container From Accidental Deletion.
You'll learn more about this option later in this section.
5. Click OK.
OUs have other properties that can be useful to configure. These properties can be set after the object has been created.
6. Right-click the OU and choose Properties.
Follow the naming conventions and other standards and processes of your organization. You can use the *Description* field to explain the purpose of an OU.
If an OU represents a physical location, such as an office, the OU's address properties can be useful.
The Managed By tab can be used to link to the user or group that is responsible for the OU. Click the Change button underneath the Name box. By default, the Select User, Contact, Or Group dialog box that appears does not, despite its name, search for groups; to search for groups, you must first click the Object Types button and select Groups. You'll learn about the Select Users, Contacts, Or Groups dialog box later in this lesson. The remaining contact information on the Managed By tab is populated from the account specified in the Name box. The Managed By tab is used solely for contact information—the specified user or group does not gain any permissions or access to the OU.
7. Click OK.

The Windows Server 2008 administrative tools add a new option: the Protect Container From Accidental Deletion. This option adds a safety switch to the OU so that it cannot be accidentally deleted. Two permissions are added to the OU: Everyone::Deny::Delete and Everyone::Deny::Delete Subtree. No user, not even an administrator, will be able to delete the OU and its contents accidentally. It is highly recommended that you enable this protection for all new OUs.

If you want to delete the OU, you must first turn off the safety switch. To delete a protected OU, follow these steps:

1. In the Active Directory Users And Computers snap-in, click the View menu and select Advanced Features.
2. Right-click the OU and choose Properties.
3. Click the Object tab.
If you do not see the Object tab, you did not enable Advanced Features in step 1.
4. Clear the check box labeled Protect Object From Accidental Deletion.
5. Click OK.

6. Right-click the OU and choose Delete.
7. You will be prompted to confirm that you want to delete the OU. Click Yes.
8. If the OU contains any other objects, you will be prompted by the Confirm Subtree Deletion dialog box to confirm that you want to delete the OU and all the objects it contains. Click Yes.

Quick Check

- You attempt to delete an OU and receive an insufficient privileges error. You are logged on as a member of Domain Admins, so you are certain you should have permission to delete an OU. What is happening and what must you change to delete the OU?

Quick Check Answer

- The OU is protected from accidental deletion. You must deselect the option to protect the object from accidental deletion. The option is located on the Object tab of the OU's Properties dialog box, which is accessible only when Advanced Features is enabled.

Creating a User Object

To create a new user in Active Directory, perform the following steps. Be certain to follow the naming conventions and processes specified by your organization.

1. Open the Active Directory Users And Computers snap-in.
2. In the console tree, expand the node that represents your domain (for instance, *contoso.com*) and navigate to the OU or container (for example, Users) in which you want to create the user account.
3. Right-click the OU or container, choose New, and then select User.
The New Object – User dialog box appears, as shown in Figure 2-5.
4. In First Name, type the user's first name.
5. In Initials, type the user's middle initial(s).
Note that this property is, in fact, meant for the initials of a user's middle name, not the initials of the user's first and last name.
6. In Last Name, type the user's last name.
7. The *Full Name* field is populated automatically. Make modifications to it if necessary.
The *Full Name* field is used to create several attributes of a user object, most notably the common name (CN), and to display name properties. The CN of a user is the name displayed in the details pane of the snap-in. It must be unique within the container or OU. Therefore, if you are creating a user object for a person with the same name as an existing user in the same OU or container, you will need to enter a unique name in the *Full Name* field.

The screenshot shows the 'New Object - User' dialog box. At the top, it says 'Create in: contoso.com/People/Employees'. Below that, there are several input fields: 'First name' with 'James', 'Last name' with 'Fine', and 'Full name' with 'James Fine'. There is an 'Initials' field which is empty. Under 'User logon name', there is a text box with 'jfine' and a dropdown menu showing '@contoso.com'. Below that, 'User logon name (pre-Windows 2000):' has a text box with 'CONTOSO\jfine'. At the bottom, there are three buttons: '< Back', 'Next >', and 'Cancel'.

Figure 2-5 New Object – User dialog box

8. In User Logon Name, type the name that the user will log on with and, from the drop-down list, select the user principle name (UPN) suffix that will be appended to the user logon name following the @ symbol.

User names in Active Directory can contain some special characters (including periods, hyphens, and apostrophes), which enable you to generate accurate user names such as O'Hara and Smith-Bates. However, certain applications can have other restrictions, so it is recommended to use only standard letters and numerals until you have fully tested the applications in your enterprise for compatibility with special characters in logon names.

The list of available UPN suffixes can be managed using the Active Directory Domains And Trusts snap-in. Right-click the root of the snap-in, Active Directory Domains And Trusts, choose Properties, and then use the UPN Suffixes tab to add or remove suffixes. The DNS name of your Active Directory domain will always be available as a suffix and cannot be removed.

9. In the User logon name (Pre-Windows 2000) box of the Active Directory Users And Computers snap-in, enter the pre-Windows 2000 logon name, often called the down-level logon name.

In Chapter 3, "Users," you will learn about the two different logon names.

10. Click Next.
11. Enter an initial password for the user in the Password and Confirm Password boxes.
12. Select User Must Change Password At Next Logon.

It is recommended that you always select this option so that the user can create a new password unknown to the IT staff. Appropriate support staff members can always reset the user's password at a future date if they need to log on as the user or access the user's resources. However, only users should know their passwords on a day-to-day basis.

13. Click Next.
14. Review the summary and click Finish.

The New Object – User interface enables you to configure a limited number of account-related properties such as name and password settings. However, a user object in Active Directory supports dozens of additional properties. These can be configured after the object has been created.
15. Right-click the user object you created and choose Properties.
16. Configure user properties.

Be certain to follow the naming conventions and other standards of your organization. You will learn more about many of the user properties in Chapter 3 and Chapter 8, “Authentication.”
17. Click OK.

Creating a Group Object

Groups are an important class of object because they are used to collect users, computers, and other groups to create a single point of management. The most straightforward and common use of a group is to grant permissions to a shared folder. If a group has been given read access to a folder, for example, then any of the group’s members will be able to read the folder. You do not have to grant read access directly to each individual member; you can manage access to the folder simply by adding and removing members of the group.

To create a group:

1. Open the Active Directory Users And Computers snap-in.
2. In the console tree, expand the node that represents your domain (for instance, *contoso.com*) and navigate to the OU or container (such as Users) in which you want to create the group.
3. Right-click the OU or container, choose New, and then select Group.

The New Object – Group dialog box appears, as shown in Figure 2-6.
4. Type the name of the new group in the Group Name box.

Most organizations have naming conventions that specify how group names should be created. Be sure to follow the guidelines of your organization.

By default, the name you type is also entered as the pre-Windows 2000 name of the new group. It is very highly recommended that you keep the two names the same.
5. Do not change the name in the Group Name (Pre-Windows 2000) box.
6. Choose the Group type.
 - ❑ A Security group can be given permissions to resources. It can also be configured as an e-mail distribution list.

- ❑ A Distribution group is an e-mail-enabled group that cannot be given permissions to resources and is, therefore, used only when a group is an e-mail distribution list that has no possible requirement for access to resources.

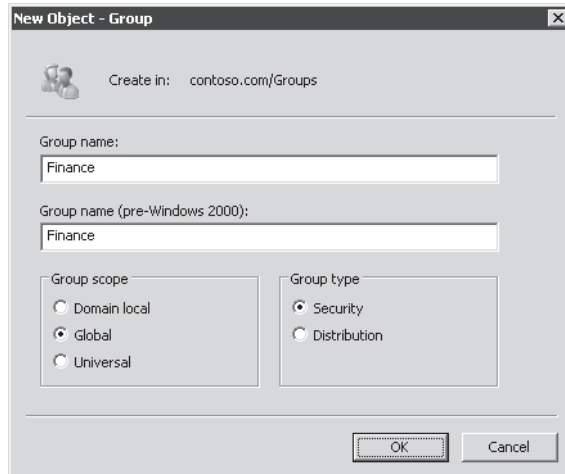


Figure 2-6 The New Object – Group dialog box

7. Select the Group Scope.
 - ❑ A Global group is used to identify users based on criteria such as job function, location, and so on.
 - ❑ A Domain local group is used to collect users and groups who share similar resource access needs, such as all users who need to be able to modify a project report.
 - ❑ A Universal group is used to collect users and groups from multiple domains.

Group scope will be discussed in more detail in Chapter 4, “Groups.”

Note that if the domain in which you are creating the group object is at a mixed or interim domain functional level, you can select only Domain Local or Global scopes for security groups. Domain functional levels will be discussed in Chapter 13, “Domains and Forests.”

8. Click OK.
9. Right-click the group and choose Properties.
10. Enter the properties for the group.

Be sure to follow the naming conventions and other standards of your organization. The group’s Members and Member Of tabs specify who belongs to the group and what groups the group itself belongs to. Group membership will be discussed in Chapter 4.

The group's *Description* field, because it is easily visible in the details pane of the Active Directory Users And Computers snap-in, is a good place to summarize the purpose of the group and the contact information for the individual(s) responsible for deciding who is and is not a member of the group.

The group's *Notes* field can be used to provide more detail about the group.

The Managed By tab can be used to link to the user or group that is responsible for the group. Click the Change button underneath the Name box. To search for a group, you must first click the Object Types button and select Groups. The Select User, Contact, Or Group dialog box will be discussed later in this lesson.

The remaining contact information on the Managed By tab is populated from the account specified in the Name box. The Managed By tab is typically used for contact information so that if a user wants to join the group, you can decide who in the business should be contacted to authorize the new member. However, if you select the Manager Can Update Membership List option, the account specified in the Name box will be given permission to add and remove members of the group. This is one method to delegate administrative control over the group. Other delegation options are discussed in Lesson 3.

11. Click OK.

Creating a Computer Object

Computers are represented as accounts and objects in Active Directory, just as users are. In fact, behind the scenes, a computer logs on to the domain just as a user does. The computer has a user name—the computer's name with a dollar sign appended, for instance, DESKTOP101\$—and a password that is established when you join the computer to the domain, and it's changed automatically every thirty days or so thereafter. To create a computer object in Active Directory:

1. Open the Active Directory Users And Computers snap-in.
2. In the console tree, expand the node that represents your domain (such as *contoso.com*) and navigate to the OU or container (for instance, Users) in which you want to create the computer.
3. Right-click the OU or container, choose New, and then select Computer.
The New Object – Computer dialog box appears, as seen in Figure 2-7.
4. In the Computer Name box, type the computer's name.
Your entry will automatically populate the Computer Name (Pre-Windows 2000) box.
5. Do not change the name in the Computer Name (Pre-Windows 2000) box.
6. The account specified in the *User Or Group* field will be able to join the computer to the domain. The default value is Domain Admins. Click Change to select another group or user.

Generally, you will select a group that represents your deployment, desktop support, or help desk team. You can also select the user to whom the computer is assigned. You will explore the issues related to joining the computer to the domain in Chapter 5, “Computers.”

7. Do not select the check box labeled Assign This Computer Account As A Pre-Windows 2000 Computer unless the account is for a computer running Microsoft Windows NT 4.0.

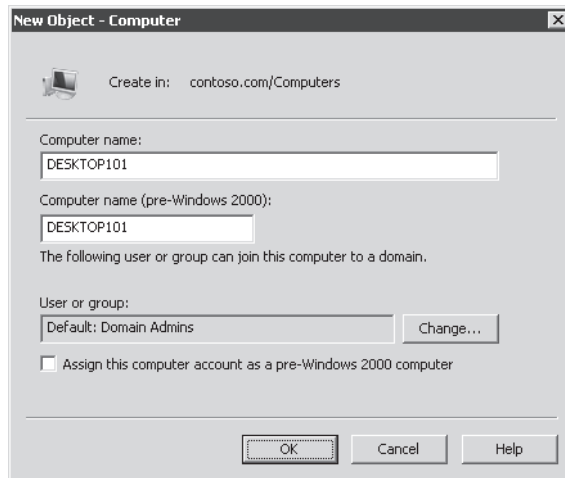


Figure 2-7 The New Object – Computer dialog box

8. Click OK.
9. Right-click the computer and choose Properties.
10. Enter the properties for the computer.

Be sure to follow the naming conventions and other standards of your organization.

The computer’s *Description* field can be used to indicate who the computer is assigned to, its role (for instance, a training-room computer), or other descriptive information. Because *Description* is visible in the details pane of the Active Directory Users And Computers snap-in, it is a good place to store the information you find most useful to know about a computer.

There are several properties that describe the computer, including DNS Name, DC Type, Site, Operating System Name, Version, and Service Pack. These properties will be populated automatically when the computer joins the domain.

The *Managed By* tab can be used to link to the user or group responsible for the computer. Click the *Change* button underneath the *Name* box. To search for groups, you must first click the *Object Types* button and select *Groups*. The *Select Users, Contacts,*

Or Groups dialog box is discussed later in this lesson. The remaining contact information on the Managed By tab is populated from the account specified in the Name box. The Managed By tab is typically used for contact information. Some organizations use the tab to indicate the support team (group) responsible for the computer. Others use the information to track the user to whom the computer is assigned.

11. Click OK.

Finding Objects in Active Directory

You have learned how to create objects in Active Directory, but what good is information in a directory service if you can't get it out of the directory as well? You will need to locate objects in Active Directory on many occasions:

- **Granting permissions** When you configure permissions for a file or folder, you must select the group (or user) to which permissions should be assigned.
- **Adding members to groups** A group's membership can consist of users, computers, groups, or any combination of the three. When you add an object as a member of a group, you must select the object.
- **Creating links** Linked properties are properties of one object that refer to another object. Group membership is, in fact, a linked property. Other linked properties, such as the *Managed By* attribute discussed earlier, are also links. When you specify the *Managed By* name, you must select the appropriate user or group.
- **Looking up an object** You can search for any object in your Active Directory domain.

There are many other situations that will entail searching Active Directory, and you will encounter several user interfaces. In this section, you'll learn some techniques for working with each.

Controlling the View of Objects in the Active Directory Users and Computers Snap-in

The details pane of the Active Directory Users and Computers snap-in can be customized to help you work effectively with the objects in your directory. Use the *Add/Remove Columns* command on the View menu to add columns to the details pane. Not every attribute is available to be displayed as a column, but you are certain to find columns that will be useful to display such as User Logon Name. You might also find columns that are unnecessary. If your OUs have only one type of object (user or computer, for example), the Type column might not be helpful.

When a column is visible, you can change the order of columns by dragging the column headings to the left or right. You can also sort the view in the details pane by clicking the column: the first click will sort in ascending order, the second in descending order, just like Windows Explorer. A common customization is to add the Last Name column to a view of users so they

can be sorted by last name. It is generally easier to find users by last name than by the Name column, which is the CN and generally first name/last name.

Using Saved Queries

Windows Server 2003 introduced the Saved Queries node of the Active Directory Users and Computers snap-in. This powerful function enables you to create rule-driven views of your domain, displaying objects across one or more OUs. To create a saved query:

1. Open the Active Directory Users And Computers snap-in.
Saved Queries is not available in the Active Directory Users And Computers snap-in that is part of Server Manager. You must use the Active Directory Users And Computers console or a custom console with the snap-in.
2. Right-click Saved Queries, choose New, and then select Query.
3. Type a name for the query.
4. Optionally, enter a description.
5. Click Browse to locate the root for the query.
The search will be limited to the domain or OU you select. It is recommended to narrow your search as much as possible to improve search performance.
6. Click Define Query to define your query.
7. In the Find Common Queries dialog box, select the type of object you want to query.
The tabs in the dialog box and the input controls on each tab change to provide options that are appropriate for the selected query.
8. Click OK.

After your query is created, it is saved within the instance of the Active Directory Users And Computers snap-in, so if you open the Active Directory Users And Computers console (*dsa.msc*), your query will be available the next time you open the console. If you created the saved query in a custom console, it will be available in that custom console. To transfer saved queries to other consoles or users, you can export the saved query as an XML file and then import it to the target snap-in.

The view in the details pane of the saved query can be customized as described earlier, with specific columns and sorting. A very important benefit of saved queries is that the customized view is specific to each saved query. When you add the Last Name column to the “normal” view of an OU, the Last Name column is actually added to the view of every OU, so you will see an empty Last Name column even for an OU of computers or groups. With saved queries, you can add the Last Name column to a query for user objects and other columns for other saved queries.

Saved queries are a powerful way to virtualize the view of your directory and monitor for issues such as disabled or locked accounts. Learning to create and manage saved queries is a worthwhile use of your time.

MORE INFO Saved queries

The following site is highly recommended for details and examples of saved queries: http://www.petri.co.il/saved_queries_in_windows_2003_dsa.htm.

Using the Select Users, Contacts, Computers, Or Groups Dialog Box

When you add a member to a group, assign a permission, or create a linked property, you are presented with the Select Users, Contacts, Computers, Or Groups dialog box shown in Figure 2-8. This dialog box is referred to as the *Select dialog box* throughout this training kit. If you'd like to see an example, open the properties of a group object, click the Members tab, and then click the Add button.

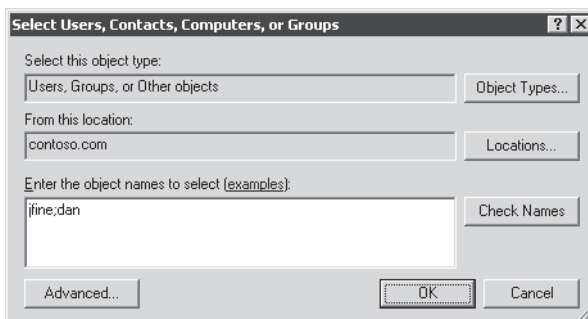


Figure 2-8 Select Users, Contacts, Computers, Or Groups dialog box

If you know the names of the objects you need, you can type them directly into the Enter The Object Names To Select text box. Multiple names can be entered, separated by semicolons, as shown in Figure 2-8. When you click OK, Windows looks up each item in the list and converts it into a link to the object, then closes the dialog box. The Check Names button also converts each name to a link but leaves the dialog box open, as shown in Figure 2-9.

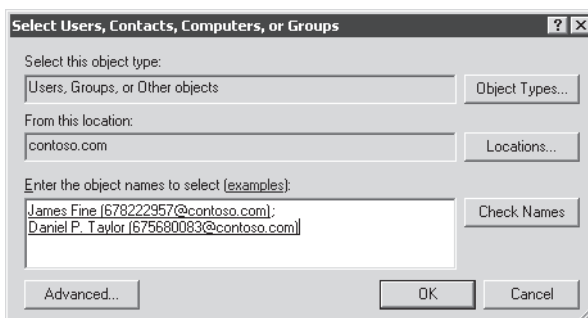


Figure 2-9 Names resolved to links using the Check Names button

You do not need to enter the full name; you can enter partial names instead. For example, Figure 2-8 shows the names `jfine` and `dan`. When you click OK or Check Names, Windows will attempt to convert your partial name to the correct object. If there is only one matching object, such as the logon name `jfine`, the name will be resolved as shown in Figure 2-9. If there are multiple matches, such as the name `Dan`, the Multiple Names Found box, shown in Figure 2-10, appears. Select the correct name(s) and click OK. The selected name appears as shown in Figure 2-9.

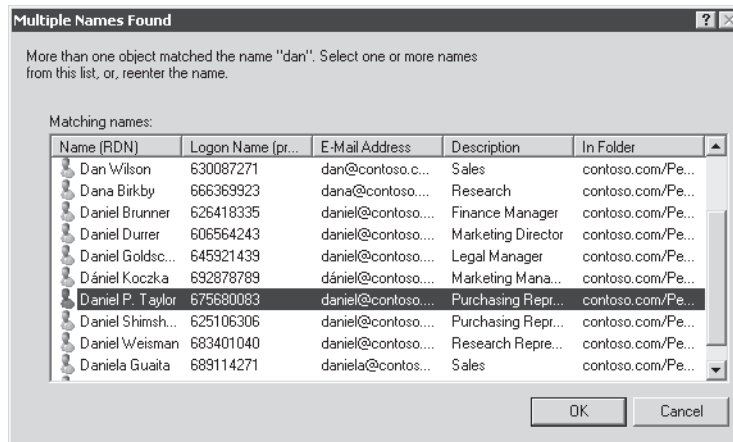


Figure 2-10 The Multiple Names Found dialog box

By default, the Select dialog box searches the entire domain. If you are getting too many results and wish to narrow the scope of your search, or if you need to search another domain or the local users and groups on a domain member, click Locations.

Additionally, the Select dialog box, despite its full name—Select Users, Contacts, Computers, Or Groups—rarely searches all four object types. When you add members to a group, for example, computers are not searched by default. If you enter a computer name, it will not be resolved correctly. When you specify Name on the Managed By tab, groups are not searched by default. You must make sure that the Select dialog box is scoped to resolve the types of objects you want to select. Click the Object Types button, use the Object Types dialog box shown in Figure 2-11 to select the correct types, and then click OK.

If you are having trouble locating the objects you want, click the Advanced button on the Select dialog box. The advanced view, shown in Figure 2-12, enables you to search both name and description fields as well as disabled accounts, nonexpiring passwords, and stale accounts that have not logged on for a specific period of time. Some of the fields on the Common Queries tab might be disabled, depending on the object type you are searching. Click the Object Types button to specify exactly the type of object you want.

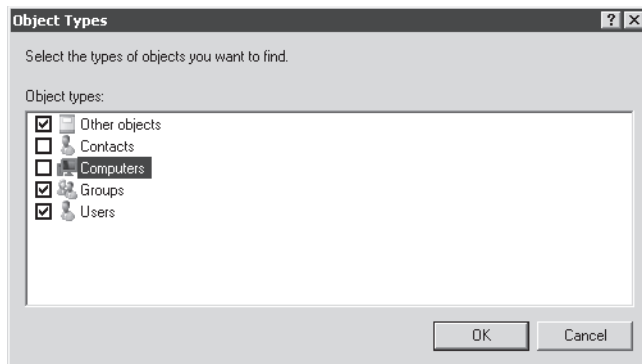


Figure 2-11 The Object Types dialog box

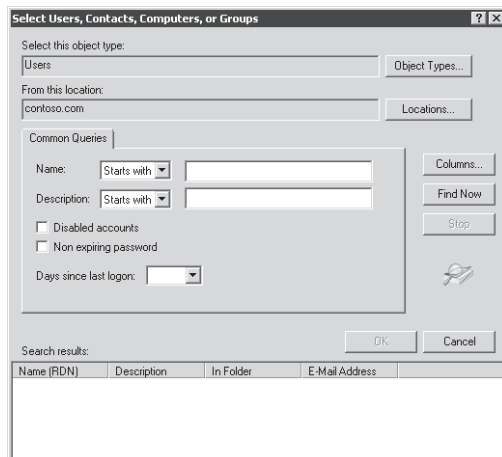


Figure 2-12 The advanced view of the Select dialog box

Using the *Find* Commands

Windows systems also provide the Active Directory query tool, called the Find box by many administrators. One way to launch the Find box is to click the Find Objects In Active Directory Domain Services button on the toolbar in the Active Directory Users And Computers snap-in. The button and the resulting Find box are shown in Figure 2-13.

Use the Find drop-down list to specify the type(s) of objects you want to query or select Common Queries or Custom Search. The In drop-down list specifies the scope of the search. It is recommended that, whenever possible, you narrow the scope of the search to avoid the performance impact of a large, domain-wide search. Together, the Find and the In lists define the scope of the search.

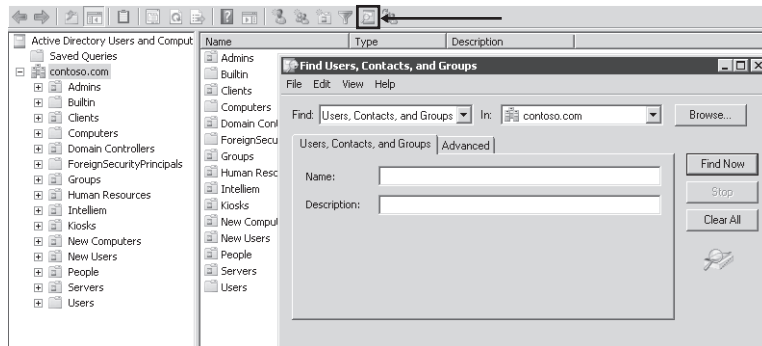


Figure 2-13 The Find box

Next, configure the search criteria. Commonly used fields are available as criteria based on the type of query you are performing. For the most complete, advanced control over the query, choose Custom Search in the Find drop-down list. If you choose Custom Search and then click the Advanced tab, you can build powerful LDAP queries. For example, the query **OU=*main*** searches for any OU with a name that contains *main* and would return the Domain Controllers OU. Without the custom search, you can search based on the text at the *beginning* of the name only; the custom search with wildcards enables you to build a “contains” search.

When you have specified your search scope and criteria, click Find Now. The results will appear. You can then right-click any item in the results list and perform commands such as *Move*, *Delete*, and *Properties*.

The Find box also appears in other Windows locations, including the Add Printer Wizard when locating a network printer. The Network folder also has a Search Active Directory button. You can add a custom shortcut, perhaps to your Start menu or desktop, to make searches even more accessible. The target of the shortcut should be `rundll32 dsquery,OpenQueryWindow`.

Finding Objects by Using *Dsquery*

Windows provides command-line utilities that perform functionality similar to that of user interface tools, such as the Active Directory Users and Computers snap-in. Many of those commands begin with the letters DS, so they are often referred to as *the DS commands*. *Dsquery* can locate objects in Active Directory.

Dsquery, like other DS commands, is well documented. Type **dsquery.exe /?** to learn its syntax and usage. Most DS commands are used by specifying the object class you want the command to work against. For example, you would type **dsquery user** to look for a user, whereas *Dsquery computer*, *Dsquery group*, and *Dsquery ou* would query for their respective object types. Following the object type specifier, you can use switches to indicate the criteria for the query. Each object can be located by its name, for example, with the *-name* switch. Most objects can

be queried based on the description (*-desc*). Security principals can be located based on their pre-Windows 2000 logon name (*-samid*). To learn which properties may be queried, type **dsquery objecttype /?**.

For example, if you want to locate all users whose names begin with “Jam,” you would type **dsquery user -name jam***. After the property switch, *name* in this case, you can enter the criteria, which are not case sensitive and can include wildcards such as the asterisk, which represents any zero or more characters. The *Dsquery* command returns matching objects with their distinguished names (DNs) by default, as you can see in Figure 2-14.

```
c:\>dsquery user -name jam*
"CN=James D. Kramer,OU=Employees,OU=People,DC=contoso,DC=com"
"CN=James Fine,OU=People,DC=contoso,DC=com"
"CN=James Hendergart,OU=Employees,OU=People,DC=contoso,DC=com"
"CN=James R. Hamilton,OU=Employees,OU=People,DC=contoso,DC=com"
"CN=James van Eaton,OU=Employees,OU=People,DC=contoso,DC=com"
"CN=Jamie Reding,OU=Employees,OU=People,DC=contoso,DC=com"
```

Figure 2-14 The *Dsquery* command

If DN is not the way you’d like to see the results, add the *-o* switch to the *Dsquery* command. You can add *-o samid*, for example, to return the results with pre-Windows 2000 logon names, or *-o upn* to return the list as user logon names, called UPNs.

Understanding DNs, RDNs, and CNs

DNs are a kind of path to an object in Active Directory. Each object in Active Directory has a completely unique DN. Our user, James Fine, has the DN CN=James Fine,OU=People,DC=contoso,DC=com.

You can see what is happening: the DN is a path starting at the object and working up to the top-level domain in the *contoso.com* DNS namespace. As mentioned earlier, CN stands for common name, and when you create a user, the Full Name box is used to create the CN of the user object. OU means organizational unit, not surprisingly. And DC means domain component.

The portion of the DN prior to the first OU or container is called the *relative distinguished name*, or RDN. In the case of James Fine, the RDN of the object is CN=James Fine. Not every RDN is a CN. The DN of the People OU is OU=People,DC=contoso,DC=com. The RDN of the People OU is, therefore, OU=People.

Because the DN of an object must be unique within the directory service, the RDN of an object must be unique within its container. That’s why if you hire a second James Fine, and if both user objects should be in the same OU, you will have to give that user a different CN. The same logic applies as files in a folder: you cannot have two files with identical names in a single folder.

You will encounter DNs regularly as you work with Active Directory, just as you encounter file paths regularly if you work with files and folders. It’s very important to be able to read them and interpret them.

Lesson 3: Delegation and Security of Active Directory Objects

In previous lessons of this chapter, you've learned how to create users, groups, computers, and OUs and how to access the properties of those objects. Your ability to perform those actions was dependent on your membership in the Administrators group of the domain. You would not want every user on your help desk team to be a member of the domain's Administrators group just to reset user passwords and unlock user accounts. Instead, you should enable the help desk and each role in your organization to perform the tasks that are required of the role and no more. In this lesson, you'll learn how to delegate specific administrative tasks within Active Directory, which is achieved by changing the access control lists (ACLs) on Active Directory objects.

After this lesson, you will be able to:

- Describe the business purpose of delegation.
- Assign permissions to Active Directory objects by using the security editor user interfaces and the Delegation of Control Wizard.
- View and report permissions on Active Directory objects by using user interface and command-line tools.
- Evaluate effective permissions for a user or group.
- Reset the permissions on an object to its default.
- Describe the relationship between delegation and OU design.

Estimated lesson time: 35 minutes

Understanding Delegation

In most organizations, there is more than one administrator, and as organizations grow, administrative tasks are often distributed to various administrators or support organizations. For example, in many organizations, the help desk is able to reset user passwords and unlock the accounts of users who are locked out. This capability of the help desk is a delegated administrative task. The help desk cannot, usually, create new user accounts, but it can make specific changes to existing user accounts.

All Active Directory objects, such as the users, computers, and groups you created in the previous lesson, can be secured using a list of permissions, so you could give your help desk permission to reset passwords on user objects. The permissions on an object are called *access control entries (ACEs)*, and they are assigned to users, groups, or computers (called *security principals*). ACEs are saved in the object's discretionary access control list (DACL). The DACL is a part of the object's ACL, which also contains the system access control list (SACL) that includes auditing settings. This might sound familiar to you if you have studied the permissions on files and folders—the terms and concepts are identical.

The delegation of administrative control, also called the delegation of control or just delegation, simply means assigning permissions that manage access to objects and properties in Active Directory. Just as you can give a group the ability to change files in a folder, you can give a group the ability to reset passwords on user objects.

Viewing the ACL of an Active Directory Object

At the lowest level is the ACL on an individual user object in Active Directory. To view the ACL on an object:

1. Open the Active Directory Users And Computers snap-in.
2. Click the View menu and select Advanced Features.
3. Right-click an object and choose Properties.
4. Click the Security tab.

If Advanced Features is not enabled, you will not see the Security tab in an object's Properties dialog box.

The Security tab of the object's Properties dialog box is shown in Figure 2-15.

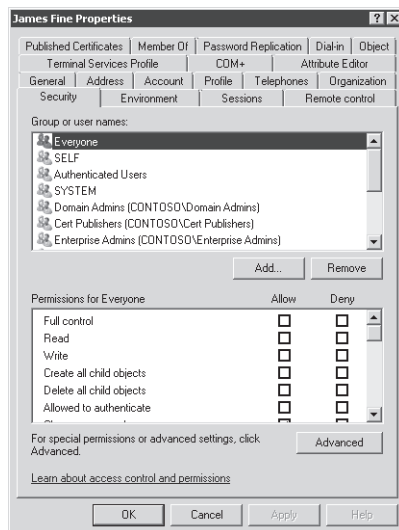


Figure 2-15 The Security tab of an Active Directory object's Properties dialog box

5. Click the Advanced button.

The Security tab shows a very high-level overview of the security principals that have been given permissions to the object, but in the case of Active Directory ACLs, the Security tab is rarely detailed enough to provide the information you need to interpret or manage the ACL. You should always click Advanced to open the Advanced Security Settings dialog box.

The dialog box showing Advanced Security Settings for an object appears, shown in Figure 2-16.

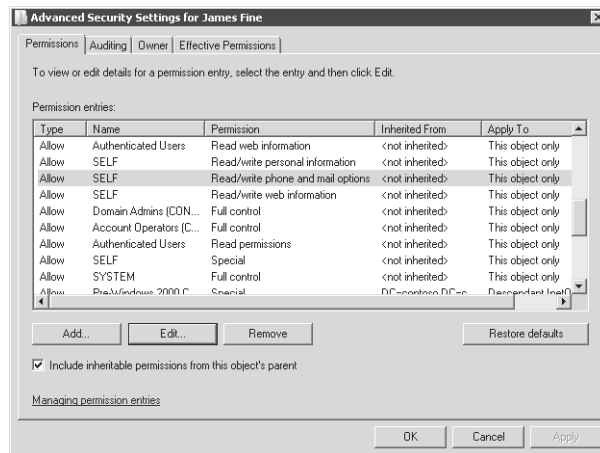


Figure 2-16 The Advanced Security Settings dialog box for an Active Directory object

The Permissions tab of the Advanced Security Settings dialog box shows the DACL of the object. You can see in Figure 2-16 that ACEs are summarized on a line of the Permission entries list. In this dialog box, you are not seeing the granular ACEs of the DACL. For example, the permission entry that is selected in Figure 2-16 is actually composed of two ACEs.

- To see the granular ACEs of a permission entry, select the entry and click Edit. The Permission Entry dialog box appears, detailing the specific ACEs that make up the entry, as in Figure 2-17.

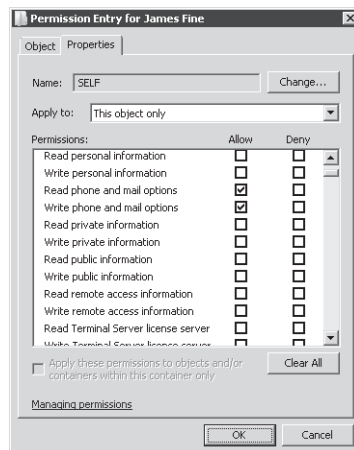


Figure 2-17 The Permission Entry dialog box

Quick Check

- You want to view the permissions assigned to an OU. You open the OU's Properties dialog box and there is no Security tab visible. What must you do?

Quick Check Answer

- In the Active Directory Users And Computers snap-in, click the View menu and select Advanced Features.

Object, Property, and Control Access Rights

The DACL of an object enables you to assign permissions to specific properties of an object. As you saw in Figure 2-17, you can allow (or deny) permission to change phone and e-mail options. This is in fact not just one property; it is a property set that includes multiple specific properties. Property sets make it easier to manage permissions to commonly used collections of properties. But you could get even more granular and allow or deny permission to change just the mobile telephone number or just the home street address.

Permissions can also be assigned to manage control access rights, which are actions such as changing or resetting a password. The difference between those two control access rights is important to understand. If you have the right to *change* a password, you must know and enter the current password before making the change. If you have the right to *reset* a password, you are not required to know the previous password.

Finally, permissions can be assigned to objects. For example, the ability to change permissions on an object is controlled by the Allow::Modify Permissions ACE. Object permissions also control whether you are able to create child objects. For example, you might give your desktop support team permissions to create computer objects in the OU for your desktops and laptops. The Allow::Create Computer Objects ACE would be assigned to the desktop support team at the OU.

The type and scope of permissions are managed using the two tabs, Object and Properties, and the Apply To drop-down lists on each tab.

Assigning a Permission Using the Advanced Security Settings Dialog Box

Imagine a scenario in which you want to allow the help desk to change the password on James Fine's account. In this section, you will learn to do it the most complicated way first: by assigning the ACE on the DACL of the user object. Later, you'll learn how to perform the delegation by using the Delegation Of Control Wizard for the entire OU of users, and you'll see why this latter practice is recommended.

1. Open the Active Directory Users And Computers snap-in.
2. Click the View menu and select Advanced Features.
3. Right-click an object and choose Properties.
4. Click the Security tab.
5. Click the Advanced button.
6. Click the Add button.

If you have User Account Control enabled, you might need to click Edit and, perhaps, enter administrative credentials before the Add button will appear.

7. In the Select dialog box, select the security principal to which permissions will be assigned.

It is an important best practice to assign permissions to groups, not to individual users. In your example, you would select your Help Desk group.

8. Click OK.

The Permission Entry dialog box appears.

9. Configure the permissions you want to assign.

For our example, on the Object tab, scroll down the list of Permissions and select Allow::Reset Password.

10. Click OK to close each dialog box.

Understanding and Managing Permissions with Inheritance

You can imagine that assigning the help desk permission to reset passwords for each individual user object would be quite time-consuming. Luckily, you don't have to and, in fact, it's a terrible practice to assign permissions to individual objects in Active Directory. Instead, you will assign permissions to organizational units. The permissions you assign to an OU will be inherited by all objects in the OU. Thus, if you give the help desk permission to reset passwords for user objects, and you attach that permission to the OU that contains your users, all user objects within that OU will inherit that permission. With one step, you'll have delegated that administrative task.

Inheritance is an easy concept to understand. Child objects inherit the permissions of the parent container or OU. That container or OU in turn inherits its permissions from its parent container, OU, or, if it is a first-level container or OU, from the domain itself. The reason child objects inherit permissions from their parents is that, by default, each new object is created with the Include Inheritable Permissions From This Object's Parent option enabled. You can see the option in Figure 2-16.

Note, however, that as the option indicates, only *inheritable* permissions will be inherited by the child object. Not every permission, however, is inheritable. For example, the permission to reset passwords assigned to an OU would not be inherited by group objects because group

objects do not have a password attribute. So inheritance can be scoped to specific object classes: passwords are applicable to user objects, not to groups. Additionally, you can use the Apply To box of the Permission Entry dialog box to scope the inheritance of a permission. The conversation can start to get very complicated. What you should know is that, by default, new objects inherit inheritable permissions from their parent object—usually an OU or container.

What if the permission being inherited is not appropriate? Two things can be done to modify the permissions that a child object is inheriting. First, you can disable inheritance by deselecting the Include Inheritable Permissions From This Object's Parent option in the Advanced Security Settings dialog box. When you do, the object will no longer inherit any permissions from its parent—all permissions will be explicitly defined for the child object. This is generally not a good practice because it creates an exception to the rule that is being created by the permissions of the parent containers.

The second option is to allow inheritance but override the inherited permission with a permission assigned specifically to the child object—an explicit permission. Explicit permissions always override permissions that are inherited from parent objects. This has an important implication: an explicit permission that *allows* access will actually override an inherited permission that *denies* the same access. If that sounds counterintuitive to you, it is not: the rule is being defined by a parent (deny), but the child object has been configured to be an exception (allow).

Exam Tip Look out for scenarios in which access or delegation are not performing as expected either because inheritance has been broken—the child is no longer inheriting permissions from its parent—or because the child object has an explicit permission that overrides the permissions of the parent.

Delegating Administrative Tasks with the Delegation Of Control Wizard

You've seen the complexity of the DACL, and you've probably gleaned that managing permissions by using the Permission Entry dialog box is not a simple task. Luckily, the best practice is not to manage permissions by using the security interfaces but, rather, to use the Delegation of Control Wizard. The following procedure details the use of the wizard.

1. Open the Active Directory Users And Computers snap-in.
2. Right-click the node (Domain or OU) for which you want to delegate administrative tasks or control and choose Delegate Control.

In this example, you would select the OU that contains your users.

The Delegation of Control Wizard is displayed to guide you through the required steps.

3. Click Next.
You will first select the administrative group to which you are granting privileges.
4. On the Users or Groups page, click the Add button.
5. Use the Select dialog box to select the group and click OK.
6. Click Next.
Next, you will specify the specific task you wish to assign that group.
7. On the Tasks To Delegate page, select the task.
In this example, you would select Reset User Passwords and Force Password Change at Next Logon.
8. Click Next.
9. Review the summary of the actions that have been performed and click Finish.
The Delegation of Control Wizard applies the ACEs that are required to enable the selected group to perform the specified task.

Reporting and Viewing Permissions

There are several other ways to view and report permissions when you need to know who can do what. You've already seen that you can view permissions on the DACL by using the Advanced Security Settings and Permission Entry dialog boxes.

Dsacls.exe is also available as a command-line tool that reports on directory service objects. If you type the command, followed by the distinguished name of an object, you will see a report of the object's permissions. For example, this command will produce a report of the permissions associated with the People OU:

```
dsac1s.exe "ou=People,dc=contoso,dc=com"
```

Dsacls can also be used to set permissions—to delegate. Type **dsac1s.exe /?** for help regarding the syntax and usage of *Dsacls*.

Removing or Resetting Permissions on an Object

How do you remove or reset permissions that have been delegated? Unfortunately, there is no undelegate command. You must use the Advanced Security Settings and Permission Entry dialog boxes to remove permissions. If you want to reset the permissions on the object back to the defaults, open the Advanced Security Settings dialog box and click Restore Defaults. The default permissions are defined by the Active Directory schema for the class of object. After you've restored the defaults, you can reconfigure the explicit permissions you want to add to the DACL. *Dsacls* also provides the */s* switch to reset permissions to the schema-defined defaults, and the */t* switch makes the change for the entire tree—the object and all its child

objects. For example, to reset permissions on the People OU and all its child OUs and objects, you would type:

```
dsac1s "ou=People,dc=contoso,dc=com" /resetDefaultDACL
```

Understanding Effective Permissions

Effective permissions are the resulting permissions for a security principal, such as a user or group, based on the cumulative effect of each inherited and explicit ACE. Your ability to reset a user's password, for example, can be due to your membership in a group that was allowed Reset Password permission on an OU several levels above the user object. The inherited permission assigned to a group to which you belong resulted in an effective permission of Allow::Reset Password. Your effective permissions can be complicated when you consider allow and deny permissions, explicit and inherited ACEs, and the fact that you might belong to multiple groups, each of which might be assigned different permissions.

Permissions, whether assigned to your user account or to a group to which you belong, are equivalent. In the end, an ACE applies to you, the user. The best practice is to manage permissions by assigning them to groups, but it is also possible to assign ACEs to individual users or computers. Just because a permission has been assigned directly to you, the user, doesn't mean that permission is either more important or less important than a permission assigned to a group to which you belong.

Permissions that allow access (allow permissions) are cumulative. When you belong to several groups, and those groups have been granted permissions that allow a variety of tasks, you will be able to perform all the tasks assigned to all those groups as well as tasks assigned directly to your user account.

Permissions that deny access (deny permissions) override an equivalent allow permission. If you are in one group that has been allowed the permission to reset passwords, and another group that has been denied permission to reset passwords, the deny permission will prevent you from resetting passwords.

NOTE Use Deny permissions sparingly

It is generally unnecessary to assign deny permissions. If you simply do not assign an allow permission, users cannot perform the task. Before assigning a deny permission, check to see whether you could achieve your goal by removing an allow permission instead. Use deny permissions rarely and thoughtfully.

Each permission is granular. Even though you've been denied the ability to reset passwords, you might still have the ability, through other allow permissions, to change the user's logon name or e-mail address.

Chapter 3

Users

Chapter 1, “Installation,” introduced Active Directory Domain Services (AD DS) as an identity and access solution. User accounts stored in the directory are the fundamental component of identity. Because of their importance, knowledge of user accounts and the tasks related to support them is critical to the success of an administrator in a Microsoft Windows enterprise.

Your ability to work effectively with user accounts can make a big difference in your overall productivity. Skills that are effective to create or modify a single user account, such as the procedures described in Chapter 2, “Administration,” can become clumsy and inefficient when you are working with large numbers of accounts, such as when creating the accounts of newly hired employees.

In this chapter, you will learn how to apply tools and techniques to automate the creation and management of users and to locate and manipulate user objects and their attributes. Along the way, you will be introduced to Microsoft Windows PowerShell, which represents the future of command line–based and automated administration for Windows technologies. You will learn a variety of options for performing each of the most common administrative tasks.

The certification exam will expect you to have a very basic understanding of the purpose and syntax of command-line utilities, Windows PowerShell, and Microsoft Visual Basic Script (VBScript). However, this chapter goes beyond the expectations of the exam to provide a solid introduction to scripting and automation. Practice what you learn in this chapter, not because you’ll need to be a scripting guru to pass the exam but because the more you can automate those tedious administrative tasks, the more you can elevate your productivity and your success.

Exam objectives in this chapter:

- Creating and Maintaining Active Directory Objects
 - Automate creation of Active Directory accounts.
 - Maintain Active Directory accounts.

Lessons in this chapter:

Before You Begin

To complete the practices in this chapter, you must have created a domain controller named SERVER01 in a domain named *contoso.com*. See Chapter 1 for detailed steps for this task.

Real World

Dan Holme

It's really amazing to stop and consider how much of our time as Windows administrators is spent performing basic tasks related to user objects. Each day in an enterprise network brings with it a unique set of challenges related to user management. Employees are hired, moved, married, and divorced, and most eventually leave the organization. As human beings, they make mistakes like forgetting passwords or locking out their accounts by logging on incorrectly.

Administrators must respond to all these changes, and user accounts are so complicated, with so many properties, that even the most well-intentioned administrators often stray from the procedures and conventions they've established. I believe that the key to efficient, effective, consistent, and secure user environments begins with raising the skill set of administrators.

Lesson 1: Automating the Creation of User Accounts

In Chapter 2, you learned how to create a user account in the Active Directory Users and Computers snap-in. Although the procedures discussed in Chapter 2 can be applied to create a small number of users, you will need more advanced techniques to automate the creation of user accounts when a large number of users must be added to the domain. In this lesson, you will learn several of these techniques.

After this lesson, you will be able to:

- Create users from user account templates.
- Import users with *CSVDE*.
- Import users with *LDIFDE*.

Estimated lesson time: 30 minutes

Creating Users with Templates

Users in a domain often share many similar properties. For example, all sales representatives can belong to the same security groups, log on to the network during similar hours, and have home folders and roaming profiles stored on the same server. When you create a new user, you can simply copy an existing user account rather than create a blank account and populate each property.

Since the days of Microsoft Windows NT 4.0, Windows has supported the concept of user account templates. A user account template is a generic user account prepopulated with common properties. For example, you can create a template account for sales representatives that is preconfigured with group memberships, logon hours, a home folder, and roaming profile path.

NOTE **Disable template user accounts**

The template account should not be used to log on to the network, so be sure to disable the account.

To create a user based on the template, select **Copy** from the shortcut menu. The **Copy Object – User Wizard** appears. You are prompted for the name, logon name, and password settings of the new user. A number of properties of the template are copied to the new user account. After a user account is created, you can view its properties, grouped by tab, in the **Properties** dialog box. Some of the tabs and properties that appear are the following:

- **General** No properties are copied from the **General** tab
- **Address** P.O. box, city, state or province, zip or postal code, and country or region. Note that the street address itself is not copied
- **Account** Logon hours, logon workstations, account options, and account expiration
- **Profile** Profile path, logon script, home drive, and home folder path

- **Organization** Department, company, and manager
- **Member Of** Group membership and primary group

NOTE What you see isn't all you get

User accounts have additional properties that are not visible on the standard tabs in the Active Directory Users and Computers snap-in. These hidden attributes include useful properties such as assistant, division, employee type, and employee ID. To view these properties, click the View menu in the Active Directory Users and Computers snap-in and select the Advanced Features option. Then open the properties of a user account and click the Attribute Editor tab. Several of these attributes, including assistant, division, and employee type, are also copied from a template to a new account.

What Is Copied Is Not Enough

Many administrators consider the list of copied attributes to be somewhat limited. For example, you might want the job title and street address attributes to be copied. You can actually modify the Active Directory schema to include additional attributes when duplicating a user. See Knowledge Base article 827832 at <http://support.microsoft.com/kb/827832> for instructions.

However, you will be well served to use more advanced methods for automating the creation of user accounts. Later in this chapter, you will learn to use directory service (DS) commands, Comma-Separated Values Data Exchange (CSVDE), LDAP Data Interchange Format Data Exchange (LDIFDE), and Windows PowerShell to automate administrative tasks. With these tools, you will have full control over the process used to provision a new account.

Using Active Directory Command-Line Tools

In Chapter 2, you were introduced to *Dsquery.exe*, one of a suite of Active Directory command-line tools collectively called *DS commands*. The following DS commands are supported in Windows Server 2008:

- **Dsadd** Creates an object in the directory.
- **Dsget** Returns specified attributes of an object.
- **Dsmod** Modifies specified attributes of an object.
- **Dsmove** Moves an object to a new container or OU.
- **Dsrm** Removes an object, all objects in the subtree beneath a container object, or both.
- **Dsquery** Performs a query based on parameters provided at the command line and returns a list of matching objects. By default, the result set is presented as the distinguished

names (DNs) of each object, but you can use the `-o` parameter with modifiers such as `dn`, `rdn`, `upn`, or `samid` to receive the results as DN, relative DN, user principal name (UPN), or pre-Windows 2000 logon names (security accounts manager [SAM] IDs).

Most of the DS commands take two modifiers after the command itself: the object type and the object's DN. For example, the following command adds a user account for Mike Fitzmaurice:

```
dsadd user "cn=Mike Fitzmaurice,ou=People,dc=contoso,dc=com"
```

The object type, `user`, immediately follows the command. After the object type is the object's DN. When the object's DN includes a space, surround the DN with quotes. The following command removes the same user:

```
dsrm user "cn=Mike Fitzmaurice,ou=People,dc=contoso,dc=com"
```

DS commands that read or manipulate attributes of objects include `Dsquery.exe`, `Dsget.exe`, and `Dsmode.exe`. To specify an attribute, include it as a parameter after the object's DN. For example, the following command retrieves the home folder path for Mike Fitzmaurice:

```
dsget user "cn=Mike Fitzmaurice,ou=People,dc=contoso,dc=com" hmdir
```

The parameter of a DS command that represents an attribute, for example, `hmdir`, is not always the same as the name of the attribute in the Active Directory Users and Computers snap-in or in the schema.

Creating Users with *Dsadd*

Use the `Dsadd` command to create objects in Active Directory. The `DSADD USER UserDN` command creates a user object and accepts parameters that specify properties of the user. The following command shows the basic parameters required to create a user account:

```
dsadd user "User DN" dsamid pre-Windows 2000 logon name
-pwd {Password | *} dmustchpwd yes
```

The `pwd` parameter specifies the password. If it is set to an asterisk (*), you are prompted for a user password. The `mustchpwd` parameter specifies that the user must change the password at next logon.

`DSADD USER` accepts a number of parameters that specify properties of the user object. Most parameter names are self-explanatory: `-email`, `-profile`, and `-company`, for example. Type `DSADD USER /?` or search the Windows Server 2008 Help And Support Center for thorough documentation of the `DSADD USER` parameters.

The special token `$username$` represents the SAM ID in the value of the `-email`, `-hmdir`, `-profile`, and `-webpg` parameters. For example, to configure a home folder for a user when creating the user with the `DSADD USER` command shown earlier, add the following parameter:

```
-hmdir \\server01\users\$username$\documents
```

Importing Users with *CSVDE*

CSVDE is a command-line tool that imports or exports Active Directory objects from or to a comma-delimited text file (also known as a comma-separated value text file, or .csv file). Comma-delimited files can be created, modified, and opened with tools as familiar as Notepad and Microsoft Office Excel. If you have user information in existing Excel or Microsoft Office Access databases, you will find that *CSVDE* is a powerful way to take advantage of that information to automate user account creation.

The basic syntax of the *CSVDE* command is:

```
csvde [-i] [-f Filename] [-k]
```

The *i* parameter specifies import mode; without it, the default mode of *CSVDE* is export. The *f* parameter identifies the file name to import from or export to. The *-k* parameter is useful during import operations because it instructs *CSVDE* to ignore errors including Object Already Exists, Constraint Violation, and Attribute Or Value Already Exists.

The import file itself is a comma-delimited text file (.csv or .txt) in which the first line defines the imported attributes by their Lightweight Directory Access Protocol (LDAP) attribute names. Each object follows, one per line, and must contain exactly the attributes listed on the first line. Here's a sample file:

```
DN,objectClass,sAMAccountName,sn,givenName,userPrincipalName  
"cn=Lisa Andrews,ou=People,dc=contoso,dc=com",user,lisa.andrews,  
Lisa,Andrews,lisa.andrews@contoso.com
```

This file, when imported by the *CSVDE* command, will create a user object for Lisa Andrews in the People OU. The user logon names, last name and first name, are configured by the file. You cannot use the *CSVDE* to import passwords, and without a password, the user account will be disabled initially. After you have reset the password, you can enable the object.

In Chapter 4, "Groups," and Chapter 5, "Computers," you will use *CSVDE* to import computers and groups. For more information about *CSVDE*, including details regarding its parameters and usage to export directory objects, type `csvde /?` or search the Windows Server 2008 Help and Support Center.

Importing Users with *LDIFDE*

You can also use *Ldifde.exe* to import or export Active Directory objects, including users. The Lightweight Directory Access Protocol Data Interchange Format (LDIF) is a draft Internet standard for file format that can be used to perform batch operations against directories that conform to the LDAP standards. LDIF supports both import and export operations as well as batch operations that modify objects in the directory. The *LDIFDE* command implements these batch operations by using LDIF files.

The LDIF file format consists of a block of lines that, together, constitute a single operation. Multiple operations in a single file are separated by a blank line. Each line comprising an operation consists of an attribute name followed by a colon and the value of the attribute. For example, suppose you wanted to import user objects for two sales representatives, named April Stewart and Tony Krijnen. The contents of the LDIF file would look similar to the following example:

```
DN: CN=April Stewart,OU=People,DC=contoso,DC=com
changeType: add
CN: April Stewart
objectClass: user
sAMAccountName: april.stewart
userPrincipalName: april.stewart@contoso.com
givenName: April
sn: Stewart
displayName: Stewart, April
mail: april.stewart@contoso.com
description: Sales Representative in the USA
title: Sales Representative
department: Sales
company: Contoso, Ltd.
```

```
DN: CN=Tony Krijnen,OU=People,DC=contoso,DC=com
changeType: add
CN: Tony Krijnen
objectClass: user
sAMAccountName: tony.krijnen
userPrincipalName: tony.krijnen@contoso.com
givenName: Tony
sn: Krijnen
displayName: Krijnen, Tony
mail: tony.krijnen@contoso.com
description: Sales Representative in The Netherlands
title: Sales Representative
department: Sales
company: Contoso, Ltd.
```

Each operation begins with the *DN* attribute of the object that is the target of the operation. The next line, *changeType*, specifies the type of operation: *add*, *modify*, or *delete*.

As you can see, the LDIF file format is not as intuitive or familiar as the comma-separated text format. However, because the LDIF format is also a standard, many directory services and databases can export LDIF files.

After creating or obtaining an LDIF file, you can perform the operations specified by the file by using the *LDIFDE* command. From a command prompt, type **ldifde /?** for usage information. The two most important switches for the *LDIFDE* command are:

- *-i* Turn on Import mode. Without this parameter, *LDIFDE* exports information.
- *-f Filename* The file from which to import, or to which to export.

For example, the following command will import objects from the file named `Newusers.ldf`:

```
ldifde Di Df newusers.ldf
```

The command accepts a variety of modifications using parameters. The most useful parameters are summarized in Table 3-1.

Table 3-1 LDIFDE Parameters

Command	Usage
General parameters	
<code>-i</code>	Import mode. (The default is Export mode.)
<code>-f filename</code>	Import or export file name.
<code>-s servername</code>	The domain controller to bind to for the query.
<code>-c FromDN ToDN</code>	Convert occurrences of <i>FromDN</i> to <i>ToDN</i> . This is useful when importing objects from another domain, for example.
<code>-v</code>	Turn on Verbose mode.
<code>-j path</code>	Log file location.
<code>-?</code>	Help.
Export-specific parameters	
<code>-d RootDN</code>	The root of the LDAP search. The default is the root of the domain.
<code>-r Filter</code>	LDAP search filter. The default is <code>(objectClass=*)</code> , meaning all objects.
<code>-p SearchScope</code>	The scope, or depth, of the search. Can be <i>subtree</i> (the container and all child containers), <i>base</i> (the immediate child objects of the container only), or <i>onelevel</i> (the container and its immediate child containers).
<code>-l list</code>	Comma-separated list of attributes to include in export for resulting objects. Useful if you want to export a limited number of attributes.
<code>-o list</code>	List of attributes (comma-separated) to omit from export for resulting objects. Useful if you want to export all but a few attributes.
Import-specific parameters	
<code>-k</code>	Ignore errors and continue processing if Constraint Violation or Object Already Exists errors appear.

Exam Tip For the 70-640 certification exam, you should understand that both *CSVDE* and *LDIFDE* are able to import and export objects by using their respective file formats. Both commands are in the export mode by default and require the `-i` parameter to specify import mode. Only *LDIFDE* is capable of modifying existing objects or removing objects. Neither command enables you to import a user's password. Only *Dsadd* supports specifying the password. If you import users with *CSVDE* or *LDIFDE*, the accounts will be disabled until you reset their passwords and enable the accounts.

Lesson 2: Creating Users with Windows PowerShell and VBScript

In Lesson 1, you learned how to use command-line tools to add or import user accounts. In this lesson, you will discover two of the most powerful tools for performing and automating administrative tasks: Windows PowerShell and VBScript. Both of these tools enable you to create scripts that can automate the creation of user accounts. Windows PowerShell also enables you to create users from a twenty-first century command shell that lives up to its middle name, *Power*.

After this lesson, you will be able to:

- Install the Windows PowerShell feature on Windows Server 2008.
- Identify key elements of the Windows PowerShell syntax, including cmdlets, variables, aliases, namespaces, and providers.
- Create a user in Windows PowerShell.
- Create a user in VBScript.

Estimated lesson time: 75 minutes

Introducing Windows PowerShell

Windows PowerShell is a powerful tool for performing and automating administrative tasks in Windows Server 2008.

Exam Tip This section introduces you to Windows PowerShell so that you can become familiar with this important administrative tool. You are not expected to create Windows PowerShell scripts on the 70-640 exam; however, you should be able to recognize cmdlets used for basic Active Directory tasks such as those described in this training kit. If you want to learn to administer using Windows PowerShell, refer to *Windows PowerShell Scripting Guide* by Ed Wilson (Microsoft Press, 2008).

Windows PowerShell is both a command-line shell and a scripting language including more than 130 command-line tools called *cmdlets* (pronounced, “command-lets”) that follow extremely consistent syntax and naming conventions and can be extended with custom cmdlets. Unlike traditional command shells such as *Cmd.exe* in Windows or BASH in Unix that operate by sending a text command a separate process or utility and then returning the results of that command as text, Windows PowerShell performs direct manipulation of Microsoft .NET Framework objects at the command line.

Windows PowerShell is installed as a feature of Windows Server 2008. Open Server Manager and click the Add Features link to install Windows PowerShell. After you have installed Windows PowerShell, you can open it from the Start menu. It is likely that you will use Windows

PowerShell often enough to warrant creating a shortcut in a more accessible location. Right-click Windows PowerShell in the Windows PowerShell program group and choose Pin To Start Menu. The Windows PowerShell command shell looks very similar to the command prompt of *Cmd.exe* except that the default background color is dark blue, and the prompt includes PS. Figure 3-1 shows the Windows PowerShell.

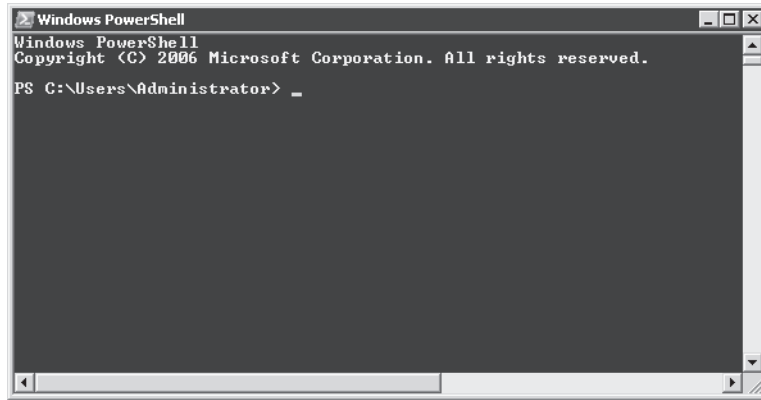


Figure 3-1 The Windows PowerShell console

NOTE One Windows, one shell

Windows PowerShell enables you to use launch programs and execute commands that are identical to those in the command shell. Therefore, Windows PowerShell is backward compatible for administrators. If you use Windows PowerShell, you can perform administrative tasks either with familiar *Cmd.exe* commands or with Windows PowerShell directives.

Understanding Windows PowerShell Syntax, Cmdlets, and Objects

In traditional shells such as *Cmd.exe*, you issue commands such as *dir* or *copy* that access utilities built into the shell, or you call executable programs such as *attrib.exe* or *xcopy.exe*, many of which accept parameters from the command line and return feedback in the form of output, errors, and error codes.

In Windows PowerShell, you issue directives by using cmdlets. A cmdlet is a single-feature command that manipulates an object. Cmdlets use a Verb-Noun syntax—a verb and a noun separated by a hyphen. Examples include *Get-Service* and *Start-Service*.

NOTE Cmdlets support direct entry and scripting

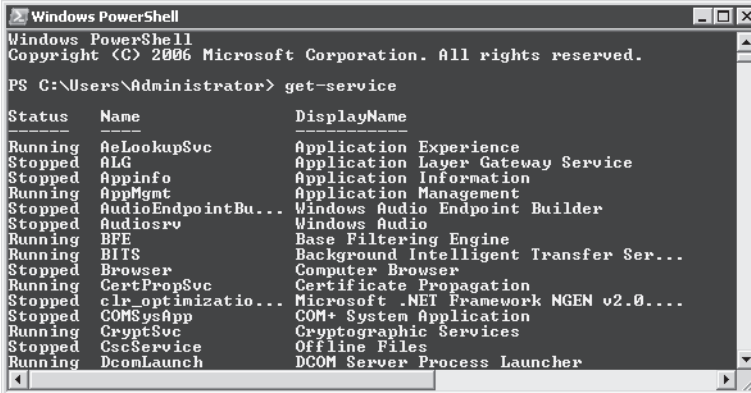
Cmdlets can be typed into the Windows PowerShell interactively or saved in script files (*.PS1) that are then executed by Windows PowerShell.

What Is an Object?

An object is a programming construct. From a technical perspective, a .NET object is an instance of a .NET class that consists of data and the operations associated with that data. Think of an object as a virtual representation of a resource of some kind. For example, when you use the `Get-Service cmdlet` in Windows PowerShell, the cmdlet returns one or more objects representing services. Objects can have *properties* that represent data, or attributes, maintained by the resource. An object representing a service, for example, has properties for the service name and its startup state. When you get a property, you are retrieving the data of the resource. When you set a property, you are writing that data to the resource.

Objects also have *methods*, which are actions that you can perform on the object. The service object has *start* and *stop* methods, for example. When you perform a method on the object that represents the resource, you perform the action on the resource itself.

These cmdlets do not pass commands or parameters to other utilities or programs but, rather, operate on .NET objects directly. If you type the cmdlet `Get-Service`, Windows PowerShell returns a *collection* of objects for all services. It presents the results of the cmdlet as a table showing the service, its name, and its display name, as shown in Figure 3-2.



```

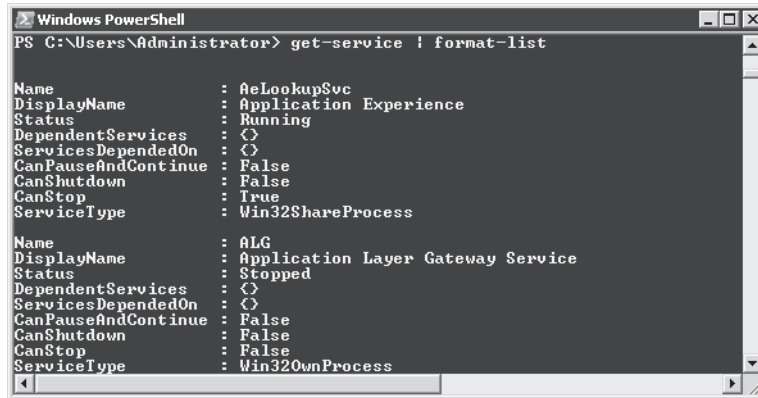
Windows PowerShell
Copyright (C) 2006 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> get-service

Status Name                DisplayName
-----
Running AeLookupSvc          Application Experience
Stopped ALG                  Application Layer Gateway Service
Stopped Appinfo          Application Information
Running AppMgmt          Application Management
Stopped AudioEndpointBu... Windows Audio Endpoint Builder
Stopped Audiosrv         Windows Audio
Running BFE              Base Filtering Engine
Running BITS             Background Intelligent Transfer Ser...
Stopped Browser          Computer Browser
Running CertPropSvc      Certificate Propagation
Stopped clr_optimizatio... Microsoft .NET Framework NGEN v2.0...
Stopped COMSysApp        COM+ System Application
Running CryptSvc          Cryptographic Services
Stopped CscService        Offline Files
Running DcomLaunch        DCOM Server Process Launcher
  
```

Figure 3-2 The `Get-Service` cmdlet

These simple commands can be used together by combining or *pipelining* to create more complex directives. For example, pipelining the `Get-Service` cmdlet to the `Format-List` cmdlet produces a different result, as Figure 3-3 shows.



```
Windows PowerShell
PS C:\Users\Administrator> get-service | format-list

Name                : AeLookupSvc
DisplayName          : Application Experience
Status              : Running
DependentServices   : {}
ServicesDependedOn  : {}
CanPauseAndContinue : False
CanShutdown         : False
CanStop             : True
ServiceType         : Win32ShareProcess

Name                : ALG
DisplayName          : Application Layer Gateway Service
Status              : Stopped
DependentServices   : {}
ServicesDependedOn  : {}
CanPauseAndContinue : False
CanShutdown         : False
CanStop             : False
ServiceType         : Win32OwnProcess
```

Figure 3-3 The Format-List cmdlet operating on the collection generated by Get-Service

Notice that the Format-List cmdlet produces far more detail than the default output of the Get-Service cmdlet. This reveals an important point. The Get-Service cmdlet is not just returning a static list of three attributes of services; it is returning objects representing the services. When those objects are pipelined, or passed, to the Format-List cmdlet, Format-List is able to work directly with those objects and display all the attributes of the services.

NOTE Subtle but important difference

This is quite different from the standard Windows command shell, in which the output of one command piped to another command can be only text. If this were *Cmd.exe*, a “format list” command could reformat only the three pieces of information provided by a “get-service” command.

The Format-List cmdlet makes decisions about which attributes to display. You can direct it to show all properties by adding a parameter, *property*, with a value of all represented by an asterisk (*). The following command will list all available properties of all services:

```
get-service | format-list *property *
```

Getting Help

The Windows PowerShell Get-Help cmdlet is the best place to start looking for information, especially when you are just getting started with Windows PowerShell. The simplest form of help is provided by typing the Get-Help cmdlet followed by the cmdlet name you want help with, for example:

```
get-help get-service
```

You can get more detailed help by adding the *detailed* or *full* parameters, for example, *get-help get-command -detailed* or *get-help get-command -full*.

Using Variables

If you are repeatedly issuing a path or object definition, you can assign it to a variable to reduce your level of effort. Variables in Windows PowerShell always begin with a dollar sign (\$). For example, you can assign the variable `$DNS` to represent the object retrieved by the `Get-Service DNS` cmdlet:

```
$DNS=get-service DNS
```

When you assign an object to a variable, you create an *object reference*. You can retrieve properties of the object by using dot (.) properties. For example, to return the status of the DNS service, type the following:

```
$DNS.status
```

A special *pipeline variable* can be used as a placeholder for the current object within the current pipeline. The pipeline variable is `$_`. For example, to get a list of all running services, type the following:

```
get-service | where-object { $_.status -eq "Running" }
```

This directive retrieves all services and pipes the objects to the `Where-Object` cmdlet, which evaluates each object in the pipeline to determine whether the object represented by the pipeline variable `$_` has a status property equal to *Running*.

Using Aliases

An *alias* is an alternative way to refer to a cmdlet. For example, the `Where-Object` cmdlet previously shown has an alias of, simply, `Where`, so the code shown previously could be shortened to the following:

```
get-service | where { $_.status -eq "Running" }
```

Many of the Windows PowerShell cmdlets have already been assigned aliases. For example, the cmdlet that displays the contents of a folder on a disk is `Get-ChildItem`. This cmdlet has been given the alias `Dir`, equivalent to the Windows command shell command, and the alias `Ls`, for users more accustomed to a UNIX shell.

How do you determine which cmdlet is behind an alias? Type **alias**, as in the following example:

```
alias dir
```

The output will reveal that `Dir` is an alias for `Get-ChildItem`.

Whereas Windows PowerShell provides aliases for command-shell commands, Windows PowerShell cmdlets do not take the same parameters as *Cmd.exe* commands. For example, to retrieve a directory of folders and all subfolders at the command prompt, type **dir /s**. In Windows PowerShell, type **dir -recurse**.

Namespaces, Providers, and PSDrives

Cmdlets operate against objects in a namespace. A folder on a disk is an example of a namespace—a hierarchy that can be navigated. Namespaces are created by providers, which you can think of as drivers. For example, the file system has a Windows PowerShell provider, as does the registry, so Windows PowerShell can directly access and manipulate objects in the namespaces of those providers.

You are certainly familiar with the concept of representing the namespace of a disk volume with a letter or representing a shared network folder's namespace as a mapped drive letter. In Windows PowerShell, namespaces from any provider can be represented as *PSDrives*. Windows PowerShell automatically creates a PSDrive for each drive letter already defined by Windows.

Windows PowerShell takes this concept to the next level by creating additional PSDrives for commonly required resources. For example, it creates two drives, *HKCU* and *HKLM*, for the `HKEY_CURRENT_USER` and `HKEY_LOCAL_MACHINE` registry hives. Now you can navigate and manipulate the registry as easily as you can a file system. Type the following in the Windows PowerShell:

```
cd hk1m:\software
dir
```

Drives are also created for aliases, environment, certificates, functions, and variables. To list the PSDrives that have been created, type **get-psdrive**.

Creating a User with Windows PowerShell

You are now ready to learn how to apply Windows PowerShell to create a user in Active Directory. The most basic Windows PowerShell script to create a user will look similar to the following:

```
$objOU=[ADSI]"LDAP://OU=People,DC=contoso,DC=com"
$objUser=$objOU.Create("user","CN=Mary North")
$objUser.Put("sAMAccountName","mary.north")
$objUser.SetInfo()
```

This code exemplifies the four basic steps to creating an object in Active Directory with Windows PowerShell:

1. Connect to the container—for example, the OU—in which the object will be created.
2. Invoke the *Create* method of the container with the object class and relative distinguished name (RDN) of the new object.
3. Populate attributes of the object with its *Put* method.
4. Commit changes to Active Directory with the object's *SetInfo* method.

Each of these steps is examined in detail in the following sections.

Connecting to an Active Directory Container

To create an object such as a user, you ask the object's container to create the object. So you begin by performing an action—a method—on the container. The first step, then, is to connect to the container. Windows PowerShell uses the Active Directory Services Interface (ADSI) type adapter to tap into Active Directory objects. A *type adapter* is a translator between the complex and sometimes quirky nature of a .NET Framework object and the simplified and consistent structure of Windows PowerShell. To connect to an Active Directory object, you submit an LDAP query string, which is simply the LDAP:// protocol moniker followed by the DN of the object. So the first line of code is as follows:

```
$objOU=[ADSI]"LDAP://OU=People,DC=contoso,DC=com"
```

Windows PowerShell uses the ADSI type adapter to create an object reference to the People OU and assigns it to a variable. The variable name *objOU* reflects programming standards that suggest a three-letter prefix to identify the type of variable, but variable names can be anything you'd like as long as they start with a dollar sign.

Invoking the *Create* Method

At this point, the variable *\$objOU* is a reference to the People OU. You can now ask the container to create the object, using the container's *Create* method. The *Create* method requires two parameters, passed as arguments: the object class and the RDN of the object. An object's RDN is the portion of its name beneath its parent container. Most object classes use the format *CN=object name* as their RDNs. The RDN of an OU, however, is *OU=organizational unit name*, and the RDN of a domain is *DC=domain name*. The following line, then, creates a user object with the RDN specified as *CN=Mary North*.

```
$objUser=$objOU.Create("user", "CN=Mary North")
```

The resulting object is assigned to the variable *\$objUser*, which will represent the object and enable you to manipulate it.

Populating User Attributes

It's important to remember that the new object and the changes you make are not saved until you commit the changes, and you cannot commit the changes successfully until all required attributes are populated. The required attribute for user objects is the pre-Windows 2000 logon name. The LDAP name for this attribute is *sAMAccountName*. Therefore, the next line of code assigns the *sAMAccountName* to the object, using the *Put* method. *Put* is a standard method for writing a property of an object. *Get* is a standard method for retrieving a property. The resulting code is:

```
$objUser.Put("sAMAccountName", "mary.north")
```

There are other mandatory attributes for a user object, including its security identifier (SID), but those attributes are created automatically by Active Directory when you commit a new user to the directory.

Committing Changes with the *SetInfo* Method

To commit the changes, use the Active Directory object's *SetInfo* method, as in the following line of code:

```
$objUser.SetInfo()
```

Populating Additional User Attributes

The preceding commands create a user with only the mandatory *sAMAccountName* attribute configured. You should populate other user attributes when creating a user object. You just learned to use the *Put* method of a user object to write a property. All you have to do is use the same method repeatedly, specifying each attribute you want to add. Examine the following code:

```
$objUser.put("sAMAccountName", $samAccountName)
$objUser.put("userPrincipalName", $userPrincipalName)
$objUser.put("displayName", $displayName)
$objUser.put("givenName", $givenName)
$objUser.put("sn", $sn)
$objUser.put("description", $description)
$objUser.put("company", $company)
$objUser.put("department", $department)
$objUser.put("title", $title)
$objUser.put("mail", $mail)
$objUser.SetInfo()
```

Each of these commands populates an attribute of a user with the value stored in a variable. Don't forget to use the *SetInfo()* method of the user object to commit the changes to Active Directory! Until you use *SetInfo()*, the changes you make are occurring only in your local copy of the object. The *SetInfo()* method evaluates your object's properties for validity. If you configured an invalid value for an attribute, you will receive an error on the *SetInfo()* line. Using the *GetInfo()* method of the user object reloads the original object, effectively undoing all your changes.

If you're not sure what the LDAP name for an attribute is, click the Attribute Editor tab of a user account in the Active Directory Users And Computers snap-in. The tab is visible when you select Advanced Features from the View menu. The Attribute Editor shows all attributes of an object, including their LDAP names and values. You can also use either of these commands to show properties that are populated for a user object:

```
$objUser.psbase.properties
$objUser | get-member
```

NOTE Multivalued attributes are different

Although most user attributes are single valued, some are multivalued. If an attribute takes multiple values, use the *PutEx()* method of the user object. Perform a search on the Internet with the following keywords: *PowerShell user array PutEx*, and you will find numerous community resources that will help you learn the nuances of working with multivalued attributes.

And what about the user's password? You do not use *Put* to set a user's password. Instead, you use the *SetPassword* method, as in the following command:

```
$objUser.SetPassword("C0mp!eXP@ssw0rd")
```

Unfortunately, *SetPassword* can be used only *after* you've created the user and invoked the *Set-Info()* method. That means, in fact, you are creating the account before assigning it a password. That's not a bug or a limitation of Windows PowerShell—it's a reality of Kerberos and LDAP. However, it's secure because the account is created in the disabled state.

You must then enable the account. The status of an account is a flag that is also not manipulated with the *Put* command. Instead, you use the following command:

```
$objUser.psbase.InvokeSet("AccountDisabled", $false)  
$objUser.SetInfo()
```

Importing Users from a Database with Windows PowerShell

Although you will not be expected to understand database imports with Windows PowerShell for the 70-640 examination, learning how to do so can be a tremendous benefit to your efforts to automate the creation of users. As you'll see, it takes only a few lines of additional code with the powerful cmdlets of Windows PowerShell.

Assume that you receive an Excel worksheet from the human resources department with information about newly hired employees. Excel can save the file as a comma-delimited text file (.csv), which can be imported by Windows PowerShell. The first line of the .csv file must have field names followed by the information about each user. As a simple example, consider the following .csv file saved as *Newusers.csv*:

Newusers.csv

```
cn, sAMAccountName, FirstName, LastName  
John Woods, john.woods, Johnathan, Woods  
Kim Akers, kim.akers, Kimberly, Akers
```

Notice that the field names do not have to match the LDAP attribute names. They will be mapped to attribute names by the script.

Windows PowerShell can import this data source with one command:

```
$dataSource=import-csv "newusers.csv"
```

After you import the data source, you must loop through each record in the data source. This is performed with a *foreach* block, which takes the following format:

```
foreach($dataRecord in $datasource)
{
    # do whatever you want to do
}
```

The `ForEach` cmdlet loops through each object or record in the data source and assigns the current object to the `$dataRecord` variable, so the `$dataRecord` variable represents the current record. You can now look at the actual fields in each record, which become properties of the `$dataRecord` variable. For example, the first name of the first user is:

```
$dataRecord.FirstName
```

You can assign it to a variable:

```
$givenName = $dataRecord.FirstName
```

Again, it is not necessary for the variable or the field name to match the LDAP attribute name. The mapping is performed when you write the variable containing the value to the attribute itself:

```
$objUser.Put("givenName", $givenName)
```

The LDAP attribute, *givenName*, is in quotes. Only when you refer to the actual attribute of an object must you use the correct name. It certainly makes it easier to follow the code, however, if data source field names and variable names reflect the attribute names.

Putting it together, you can create a user import script:

Userimport.ps1

```
$objOU=[ADSI]"LDAP://OU=People,DC=contoso,DC=com"
$dataSource=import-csv "NewUsers.csv"
foreach($dataRecord in $datasource) {
    #map variables to data source
    $cn=$dataRecord.cn
    $sAMAccountName=$dataRecord.sAMAccountName
    $givenName=$dataRecord.FirstName
    $sn=$dataRecord.LastName
    $displayName=$sn + ", " + $givenName
    $userPrincipalName=$givenName + "." + $sn + "@contoso.com"

    #create the user object
    $objUser=$objOU.Create("user", "CN="+$cn)
    $objUser.Put("sAMAccountName", $sAMAccountName)
    $objUser.Put("userPrincipalName", $userPrincipalName)
    $objUser.Put("displayName", $displayName)
    $objUser.Put("givenName", $givenName)
    $objUser.Put("sn", $sn)
    $objUser.SetInfo()
```



```
$objUser.SetPassword("C0mp!exP@ssw0rd")
$objUser.psbase.InvokeSet("AccountDisabled", $false)
$objUser.SetInfo()
}
```

The first line of the script connects to the container, the OU in which all new users will be created. The next two lines connect to the data source and loop through each record, assigning each record to a variable, *\$dataRecord*. The *foreach* block does two things. First, it maps fields in the data source to variables. Then it creates a user.

Notice that some variables are constructed by concatenating (appending) two fields. The *\$displayName* variable takes the *LastName, FirstName* format, and the *\$userPrincipalName* variable takes the *FirstName.LastName@contoso.com* format.

The user is created by invoking the *Create* method of the OU. Attributes of the user are populated and committed, and then the password is set and the account is enabled. Voilà!

Executing a Windows PowerShell Script

By default, Windows PowerShell prevents the execution of scripts as a security measure. To run a script that you have created, you must change the execution policy of Windows PowerShell with the following command:

```
set-executionpolicy remotesigned
```

The execution policy specifies which scripts can be run. The command just shown configures Windows PowerShell so that it will run local scripts but will require scripts from remote sources to be signed. Changing the execution policy has security implications, so you should read the information about running Windows PowerShell scripts at <http://www.microsoft.com/technet/scriptcenter/topics/winsh/manual/run.msp#EXC>.

After you've set the execution policy, you can run your script, but do not run it by name alone—you will receive an error. You must specify the path to the script! A shortcut is to use the *.scriptname* notation, which indicates the current directory, so the following command will execute the user import script:

```
.\UserImport.ps1
```

Introducing VBScript

VBScript is a scripting language that supports the automation of administrative tasks on all current versions of Windows. VBScript files are text files typically edited with Notepad or a script editor and saved with a *.vbs* extension. To execute a script, you can double-click it, which opens the script, using *Wscript.exe*. Alternatively, from the command line, you can run the script with *Cscript.exe*, using the following syntax:

```
cscript.exe scriptname
```

Both *Wscript.exe* and *Cscript.exe* are components of the Windows Scripting Host (WSH), which is the automation framework installed on all current versions of Windows that supports several scripting languages, including VBScript.

Creating a User with VBScript

Because VBScript also uses the ADSI interface to manipulate Active Directory, the process for creating a user in VBScript is identical to the process in Windows PowerShell. A simple script for creating a user follows:

```
Set objOU=GetObject("LDAP://OU=People,DC=contoso,DC=com")
Set objUser=objOU.Create("user", "CN=Mary North")
objUser.Put "sAMAccountName", "mary.north"
objUser.SetInfo()
```

The script first connects to the container, the OU in which the user will be created. VBScript uses the `GetObject` statement to connect to an ADSI object by its distinguished name. When you assign an object to a variable in VBScript, you use the `Set` statement to create the object reference.

The second line of code invokes the *Create* method of the OU to create an object of a specific class and with a specific relative distinguished name, just as in the Windows PowerShell example. Because the result of the method is an object, you again have to use the `Set` statement to assign the object reference to a variable.

The third line uses the *Put* method of the user object, but VBScript does not use parentheses to pass the parameters to the argument. The fourth line is identical to Windows PowerShell; it commits the changes. Save the script as `Newuser.vbs` and execute it from the command shell, or from Windows PowerShell, with this command:

```
cscript.exe newusers.vbs
```

VBScript vs. Windows PowerShell

VBScript has two major advantages over Windows PowerShell. The first is the fact that VBScript scripts can be run on all current versions of Windows using the WSH, whereas Windows PowerShell must be downloaded and installed on versions of Windows prior to Windows Server 2008 and requires .NET Framework 2.0 or greater. The second advantage of VBScript is that it has been around for many years, so there is an extraordinary amount of experience, knowledge, and community-posted information on the Internet.

However, WSH does not provide a shell for directly executing commands. Additionally, VBScript as a language is not a particularly rich scripting language and does not fully use the .NET Framework. Although the WSH exists on Windows Server 2008 and VBScript is still supported, the way of the future is Windows PowerShell. That is why it was presented first in this lesson.

The disadvantages of Windows PowerShell are the inverse of the VBScript advantages. The very fact that Windows PowerShell is new means that it is a product still in development. In the previous sections, you learned to create user accounts with Windows PowerShell. The techniques and code you learned are fairly complex in the bigger picture of Windows PowerShell. In fact, they are almost identical to VBScript.

That's because in the current version of Windows PowerShell, there is very limited support for Active Directory administration. Unlike Windows Management Interface (WMI) and Microsoft Exchange Server, which have very rich Windows PowerShell providers, Active Directory support is limited to the ADSI type adapter, which is quirky and awkward and, ultimately, relies on ADSI just as VBScript does. In future versions of Windows PowerShell, an Active Directory provider will be introduced that will make working with Active Directory objects as easy as working with files in a file system.

Remember that on the 70-640 exam, you are not expected to create scripts in either Windows PowerShell or VBScript. Be able to recognize a script that follows the correct process to create a user: Connect to the OU, create the object, populate its properties, and then commit the changes.

Lesson 3: Supporting User Objects and Accounts

The first two lessons of this chapter detailed the methods with which to create user accounts. That is only the first step in the life cycle of a user in a domain. After creating the user, you must configure attributes that define both the properties of the security principal (the account) and properties that manage the user. You must also know how and when to administer the account—to perform password resets and to unlock the account. Finally, you must be able to move the user between OUs and, eventually, deprovision the account by disabling or deleting it. This lesson will cover the procedures used to support a user object through its life cycle—procedures you can perform using both the Windows interface and the command line or automation tools.

After this lesson, you will be able to:

- Identify the purpose and requirements of user account attributes and user name properties.
- View and modify hidden attributes of user objects.
- Modify attributes of multiple users simultaneously.
- Manage users with the Active Directory Users And Computers snap-in, DS commands, Windows PowerShell, and VBScript.
- Perform common administrative tasks to support user accounts.

Estimated lesson time: 90 minutes

Managing User Attributes with Active Directory Users and Computers

When you create a user with the Active Directory Users and Computers snapin New Object–User Wizard, you are prompted for some common properties, including logon names, password, and user first and last names. A user object in Active Directory, however, supports dozens of additional properties that you can configure at any time with the Active Directory Users and Computers snap-in.

To read and modify the attributes of a user object, right-click the user and choose Properties. The user’s Properties dialog box appears, as shown in Figure 3-4. Attributes of a user object fall into several broad categories that appear on tabs of the dialog box:

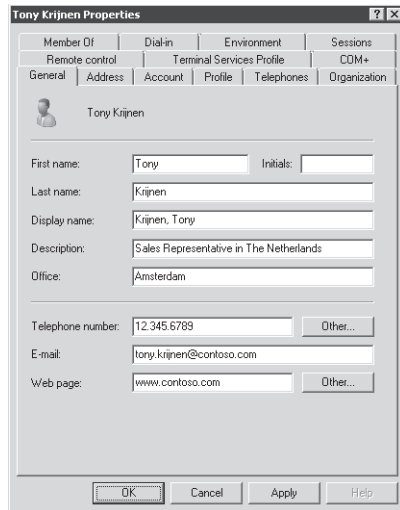


Figure 3-4 The Properties dialog box for a user

- **Account attributes: the Account tab** These properties include logon names, password, and account flags. Many of these attributes can be configured when you create a new user with the Active Directory Users and Computers snap-in. The “Account Properties” section details account attributes.
- **Personal information: the General, Address, Telephones, and Organization tabs** The General tab exposes the name properties that are configured when you create a user object, as well as basic description and contact information. The Address and Telephones tabs provide detailed contact information. The Telephones tab is also where Microsoft chose to put the *Notes* field, which maps to the *info* attribute and is a very useful general-purpose text field that is underused by many enterprises. The Organization tab shows job title, department, company, and organizational relationships.
- **User configuration management: the Profile tab** Here you can configure the user’s profile path, logon script, and home folder.
- **Group membership: the Member Of tab** You can add the user to and remove the user from groups and change the user’s primary group. Group memberships and the primary group will be discussed in Chapter 5, “Computers.”
- **Terminal services: the Terminal Services Profile, Environment, Remote Control, and Sessions tabs** These four tabs enable you to configure and manage the user’s experience when the user is connected to a Terminal Services session.

MORE INFO Terminal Services settings

For more information about configuring Terminal Services settings, see *MCTS: Configuring Windows Server 2008 Applications Infrastructure*, by J.C. Mackin and Anil Desai (Microsoft Press, 2008).

- **Remote access: the Dial-in tab** You can enable and configure remote access permission for a user on the Dial-in tab.
- **Applications: the COM+ tab** This tab enables you to assign the users to an Active Directory COM+ partition set. This feature facilitates the management of distributed applications and is beyond the scope of the 70-640 exam.

Viewing All Attributes

A user object has even more properties than are visible in its Properties dialog box. Some of the so-called hidden properties can be quite useful to your enterprise. To uncover hidden user attributes, you must turn on the Attribute Editor, a new feature in Windows Server 2008. Click the View menu and select the Advanced Features option. Then open the Properties dialog box of the user, and the Attribute Editor tab will be visible, as shown in Figure 3-5.

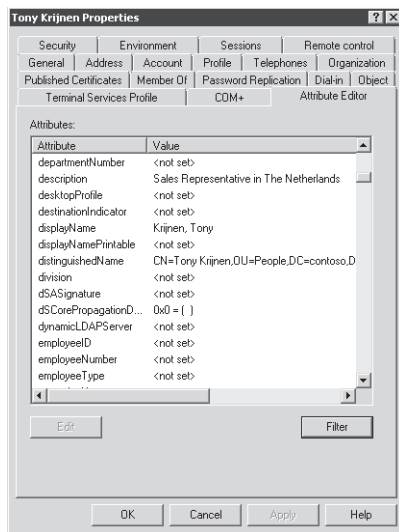


Figure 3-5 The Attribute Editor tab

The Attribute Editor displays all the system attributes of the selected object. The Filter button enables you to choose to see even more attributes, including backlinks and constructed attributes. Backlinks are attributes that result from references to the object from other objects. The easiest way to understand backlinks is to look at an example: the *memberOf* attribute. When a user is added to a group, it is the group's *member* attribute that is changed—the distinguished name of the user is added to this multivalued attribute. Therefore, the *member* attribute of a group is called a *forward link* attribute. A user's *memberOf* attribute is updated automatically by Active Directory when the user is referred to by a group's *member* attribute. You do not ever write directly to the user's *memberOf* attribute; it is dynamically maintained by Active Directory.

A *constructed* attribute is one of the results from a calculation performed by Active Directory. An example is the *tokenGroups* attribute. This attribute is a list of the security identifiers (SIDs) of all the groups to which the user belongs, including nested groups. To determine the value of *tokenGroups*, Active Directory must calculate the effective membership of the user, which takes a few processor cycles. Therefore, the attribute is not stored as part of the user object or dynamically maintained. Instead, it is calculated when needed. Because of the processing required to produce constructed attributes, the Attribute Editor does not display them by default. They also cannot be used in LDAP queries.

As you can see in Figure 3-5, some attributes of a user object could be quite useful, including *division*, *employeeID*, *employeeNumber*, and *employeeType*. Although the attributes are not shown on the standard tabs of a user object, they are now available through the Attribute Editor, and they can be accessed programmatically with Windows PowerShell or VBScript.

MORE INFO Hidden attributes of objects

For more information on using hidden attributes of objects and extending the schema with custom attributes, see *Windows Administration Resource Kit: Productivity Solutions for IT Professionals* by Dan Holme (Microsoft Press, 2008).

Managing Attributes of Multiple Users

The Active Directory Users and Computers snap-in enables you to modify the properties of multiple user objects simultaneously. Select several user objects by holding the Ctrl key as you click each user or using any other multiselection technique. Be certain that you select only objects of one class, such as users. After you have multiselected the objects, right-click any one of them and choose Properties.

When you have multiselected the user objects, a subset of properties is available for modification.

- **General** Description, Office, Telephone Number, Fax, Web Page, E-mail
- **Account** UPN Suffix, Logon Hours, Computer Restrictions (logon workstations), all Account Options, Account Expires
- **Address** Street, P.O. Box, City, State/Province, ZIP/Postal Code, Country/Region
- **Profile** Profile Path, Logon Script, and Home Folder
- **Organization** Title, Department, Company, Manager

Exam Tip Be sure to know which properties can be modified for multiple users simultaneously. Exam scenarios and simulations that suggest a need to change many user object properties as quickly as possible are often testing your understanding of multiselecting. In the real world, remember that you can and should use automation tools such as Dsmod, Windows PowerShell, and VBScript.

Understanding Name and Account Attributes

Two sets of attributes tend to appear on the certification exams and to present challenges to Windows administrators: name attributes and account attributes.

User Object Names

Several attributes are related to the name of a user object and an account. It is important to understand the distinctions between them.

- A user's *sAMAccountName* attribute (the pre-Windows 2000 logon name) must be unique for the entire domain. Many organizations use initials or some combination of first and last name to generate the *sAMAccountName*. That approach can be problematic because an organization of any size is likely to have users with names similar enough that the rules for generating the *sAMAccountName* would generate a duplicate name, so exceptions have to be built into the system and, eventually, the rules are riddled with exceptions. This problem is solved if the employee number or some other unique attribute of the users is used for the *sAMAccountName*. If you have the ability to direct the naming conventions at your organization, a unique, name-independent logon name is recommended.
- The *userPrincipalName* (UPN) attribute consists of a logon name and a UPN suffix which is, by default, the DNS name of the domain in which you create the object. The UPN must be unique for the entire forest. E-mail addresses, which must be unique for the whole world, certainly meet that requirement. Consider using e-mail addresses as UPNs. If your Active Directory domain name is not the same as your e-mail domain name, you must add the e-mail domain name as an available UPN suffix. To do this, open the Active Directory Domains And Trusts snap-in, right-click the root of the snap-in, and choose Properties.
- The RDN must be unique within an OU. For users, this means the *cn* attribute must be unique within the OU. This can be a tricky one. If you have a single, flat OU for users that already contains a user named Scott Miller, and you hire a second Scott Miller, his user object cannot have the same common name as the first. Unfortunately, there's no perfect answer to this problem for all organizations. Design a naming standard that applies a single rule for all CNs. Perhaps the CN should include an employee's number—for example, *Scott Miller (645928)*. If your OU structure for user accounts is flat, be prepared to address this challenge.

Additionally, many organizations choose to configure the *cn* attribute as *LastName*, *FirstName* because by doing so, you can sort users by last name in the Active Directory Users and Computers snap-in. This is not a recommended method to achieve the goal. Instead of using a last-name-first format for *cn*, add the Last Name column to your view in the Active Directory Users And Computers snap-in by clicking the View menu and choosing Add/Remove Columns. Then click the Last Name column header to sort by last name.

- The *displayName* attribute appears in the Exchange global address list (GAL). It can be easier to locate users in the GAL if they are sorted by last name, so you can create a naming convention for your organization that specifies that the *displayName* attribute takes the *LastName, FirstName* syntax.

Account Properties

On the Account tab of a user's Properties dialog box, shown in Figure 3-6, are the attributes directly related to the fact that a user is a security principal, meaning that it is an identity to which permissions and rights can be assigned. Other security principals include computers, groups, and the *inetOrgPerson* object class.

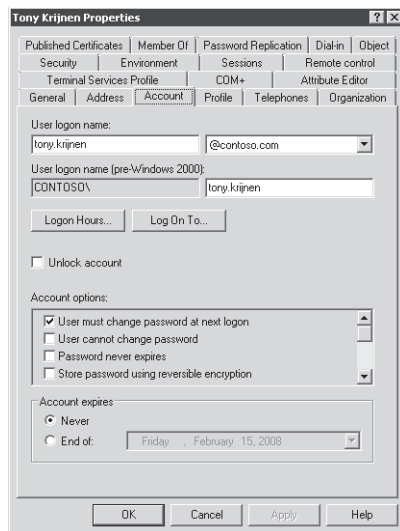


Figure 3-6 Account properties of a user object

Several of the account properties are worth highlighting because they are potentially quite useful and are not self-explanatory. Table 3-2 describes these properties.

Table 3-2 User Account Properties

Property	Description
Logon Hours	Click Logon Hours to configure the hours during which a user is allowed to log on to the network.
Log On To	Click Log On To if you want to limit the workstations to which the user can log on. This is called Computer Restrictions in other parts of the user interface and maps to the <i>userWorkstations</i> attribute. You must have NetBIOS over TCP/IP enabled for this feature to restrict users, because it uses the computer name rather than the Media Access Control (MAC) address of its network card to restrict logon.

Table 3-2 User Account Properties

Property	Description
User Must Change Password At Next Logon	Select this check box if you want the user to change the password you have entered the first time he or she logs on. You cannot select this option if you have selected Password Never Expires. Selecting this option will automatically clear the mutually exclusive option User Cannot Change Password.
User Cannot Change Password	Select this check box if you have more than one person using the same domain user account (such as Guest) or to maintain control over user account passwords. This option is commonly used to manage service account passwords. You cannot select this option if you have selected User Must Change Password At Next Logon.
Password Never Expires	Select this check box if you never want the password to expire. This option will automatically clear the User Must Change Password At Next Logon setting because they are mutually exclusive. This option is commonly used to manage service account passwords.
Account Is Disabled	Select this check box to disable the user account, for example, when creating an object for a newly hired employee who does not yet need access to the network.
Store Password Using Reversible Encryption	This option, which stores the password in Active Directory without using Active Directory's powerful, nonreversible encryption hashing algorithm, exists to support applications that require knowledge of the user password. If it is not absolutely required, do not enable this option because it weakens password security significantly. Passwords stored using reversible encryption are similar to those stored as plaintext. Macintosh clients using the AppleTalk protocol require knowledge of the user password. If a user logs on using a Macintosh client, you will need to select this option.
Smart Card Is Required For Interactive Logon	Smart cards are portable, tamper-resistant hardware devices that store unique identification information for a user. They are attached to, or inserted into, a system and provide an additional, physical identification component to the authentication process.
Account Is Trusted For Delegation	This option enables a service account to impersonate a user to access network resources on behalf of a user. This option is not typically selected, certainly not for a user object representing a human being. It is used more often for service accounts in three-tier (or multitier) application infrastructures.
Account Expires	Use the Account Expires controls to specify when an account expires.

NOTE Configure highly complex passwords for service accounts

Services require credentials with which to access system resources. Many services require a domain user account with which to authenticate, and it is common to specify that the account password never expires. In such situations, be sure you use a long, complex password. If the service account is used by services on a limited number of systems, you can increase the security of the account by configuring the Log On To property with the list of systems using the service account.

Managing User Attributes with *Dsmod* and *Dsget*

The *Dsmod* and *Dsget* commands are two Active Directory command-line tools, called DS commands. You encountered *Dsquery* in Chapter 2 and *Dsadd* in Lesson 1 of this chapter.

Dsmod

Dsmod modifies the attributes of one or more existing objects. DS commands were introduced in Lesson 1. Like other DS commands, the *Dsmod* basic syntax is:

```
dsmod user UserDN ... parameters
```

The *UserDN* parameter specifies the distinguished name of the user to modify. The remaining parameters indicate the attribute to change and the new value. For example, the following command changes the *Office* attribute of Tony Krijnen:

```
dsmod "cn=Tony Krijnen,ou=People,dc=contoso,dc=com" Office "Amsterdam"
```

The attribute parameters do not map directly to the names of LDAP attributes of a user object. For example, the *dept* parameter of the *DSMOD USER* command modifies the *department* attribute of a user object. Additionally, *DSMOD USER* can modify only a subset of user attributes. Type **DSMOD USER /?** for usage information and a list of supported parameters.

Piping Multiple DNs to *Dsmod*

The *UserDN* parameter of the *Dsmod* command does not have to be entered directly into the command line. There are two other ways to pipe DNs to it. The first is to enter the DNs into the console. Let's assume you need to change the *office* attribute of two users, Linda Mitchell and Scott Mitchell, to reflect their relocation to the Sydney office. At the command prompt, type the following command:

```
dsmod user Office "Sydney"
```

The *UserDN* parameter is missing. The console (the command prompt) waits for you to enter DNs of users. Enter one per line, surrounded with quotes, pressing Enter at the end of each DN. After entering the last DN and pressing Enter, press Ctrl+Z at the beginning of the next

line and press Enter to indicate that you are finished. The command will then execute against each of the DNs you have entered.

A more sophisticated way to send DNs to the *Dsmod* command is by piping the results of a *Dsquery* command. *Dsquery* was covered in Chapter 2; it searches Active Directory for specified criteria and returns the DNs of matching objects. For example, to change the *office* attribute of Linda and Scott Mitchell's accounts to Sydney, use the following command:

```
dsquery user &name "* Mitchell" | dsmod user &office "Sydney"
```

The *DSMOD USER* command searches Active Directory for users whose names end with *Mitchell*. The resulting objects' DNs are then piped to *DSMOD USER*, which changes the *office* attribute to *Sydney*.

As another example, assume you want to assign all users a home folder on SERVER01. The following command changes the *homeDirectory* and *homeDrive* attributes of user objects in the People OU:

```
dsquery user "ou=People,dc=contoso,dc=com" | dsmod user  
-hmdir "\\server01\users%\username%\documents" &hmdirv "U:"
```

As mentioned in Lesson 1, the special *%username%* token can be used to represent the *sAMAccountName* of user objects when using DS commands to configure the value of the *-email*, *-hmdir*, *-profile*, and *-webpg* parameters.

Dsget

The *Dsget* command gets and outputs selected attributes of one or more objects. Its syntax, like that of *Dsmod*, is:

```
dsget user UserDN... parameters
```

You can supply the DNs of one or more user objects by specifying them on the command line, separated by spaces; by entering them in the console; or by piping the results of a *DSQUERY USER* command. Unlike *Dsadd* and *Dsmod*, *Dsget* takes only a parameter and not an associated value. For example, *Dsget* takes the *samid* parameter like *Dsadd* does, but it does not take a value. Instead, it reports the current value of the attribute. For example, to display the pre-Windows 2000 logon name of Jeff Ford in the People OU, use the following command:

```
dsget user "cn=Jeff Ford,ou=People,dc=contoso,dc=com" &samid
```

To display the pre-Windows 2000 logon names of all users in the Sydney office, use this command:

```
dsquery user &office "Sydney" | dsget user &samid
```

Managing User Attributes with Windows PowerShell and VBScript

To read an attribute of a user object with Windows PowerShell or VBScript, you use the ADSI to connect to the user object, a process called *binding*. In Lesson 2, you connected to an OU to create an object. After the object exists, you connect directly to the object. One way to do so is with the Active Directory services path (*adSPath*) of the object, which is the “LDAP://” protocol moniker followed by the distinguished name of the object.

The Windows PowerShell command for connecting to the user account of Jeff Ford in the People OU is:

```
$objUser=[ADSI]"LDAP://cn=Jeff Ford,ou=People,dc=contoso,dc=com"
```

The VBScript equivalent is:

```
Set objUser=GetObject("LDAP://cn=Jeff Ford,ou=People,dc=contoso,dc=com")
```

Remember that Windows PowerShell specifies the ADSI type adapter, and VBScript uses *GetObject*. VBScript uses the Set statement to assign an object reference to a variable. Windows PowerShell does not use the Set statement and prefixes all variables with a dollar sign.

After you have a variable that references the object, you can get its properties. For example, in Windows PowerShell, type the following to report the user’s *sAMAccountName* attribute:

```
$objUser.Get("sAMAccountName")
```

In VBScript, you must indicate that you want to output the attribute. A common way to do that is with the WScript.Echo statement, as follows:

```
WScript.Echo objUser.Get("sAMAccountName")
```

You will often see a shorthand form called the *.property* (pronounced “dot-property”) format, such as *\$objUser.sAMAccountName* in Windows PowerShell and *objUser.sAMAccountName* in VBScript. Although this method works most of the time, it is recommended to specify the *Get* method, particularly when working with Active Directory objects in Windows PowerShell.

If you want to modify an attribute, you need to perform three steps:

1. Connect to the user object.
2. Modify an attribute.
3. Commit the change.

You’ve already seen how to connect to the object. The second step is to change the attribute. Most attributes are simple, single-valued attributes and can be changed with the *Put* method of the object. For example, in Windows PowerShell:

```
$objUser.put("company","Contoso, Ltd.")
```

and in VBScript:

```
objUser.put "company","Contoso, Ltd."
```

The only difference here is that VBScript does not use parentheses to pass the parameters to the *Put* method.

You can set multiple attributes during the second step. After all attributes have been specified, you must commit the changes to the directory with *SetInfo*. The Windows PowerShell version is:

```
$objUser.SetInfo()
```

The VBScript version is identical, except for the variable name:

```
objUser.SetInfo()
```

Putting the three steps together, you have a Windows PowerShell script:

```
$objUser=[ADSI]"LDAP://cn=Jeff Ford,ou=People,dc=contoso,dc=com"
$objUser.put("company","Contoso, Ltd.")
$objUser.SetInfo()
```

In VBScript, the code is as follows:

```
Set objUser=GetObject("LDAP://cn=Jeff Ford,ou=People,dc=contoso,dc=com")
objUser.put "company","Contoso, Ltd."
objUser.SetInfo()
```

What if you want to delete an attribute entirely? You must first connect to the object. Then you can set an attribute to a blank string, "" if it is a string attribute or to 0 if it is a numeric attribute and 0 if 0 is an appropriate representation of "empty." However, you can also delete the attribute entirely, assuming it is not a mandatory attribute. To do so, you must use the *PutEx* method of the user object. To delete the *office* attribute, for example, you would use the following code in Windows PowerShell:

```
$objUser.PutEx(1, "office", 0)
$objUser.SetInfo()
```

In VBScript, you would use the following lines to delete an attribute:

```
objUser.PutEx 1, "office", 0
objUser.SetInfo()
```

Administering User Accounts

The primary purpose of user objects in Active Directory is to support authentication of a human being or of a service. Accounts are provisioned, administered, and, eventually, deprovisioned. The most common administrative tasks related to user accounts are resetting a password, unlocking an account, disabling, enabling, deleting, moving, and renaming user objects.

The following sections will examine each of these tasks and how they can be performed using the Windows interface, Windows PowerShell, VBScript, or the command prompt. Each of these tasks requires you to have appropriate permissions to the user objects. Delegating administrative permissions was discussed in Chapter 2.

Resetting a User's Password

If the user forgets his or her password and attempts to log on, he or she will receive a logon message, as shown in Figure 3-7.

Before the user can log on successfully, you will have to reset that password. You do not need to know the user's old password to do so. Simply right-click the user's object in Active Directory and choose Reset Password. The Reset Password dialog box, shown in Figure 3-8, appears. Enter the new password in both the New Password and Confirm Password boxes. It is a best practice to select the User Must Change Password At Next Logon option so that the user's password is known only to the user.



Figure 3-7 A logon message notifying a user that the user name or password is invalid

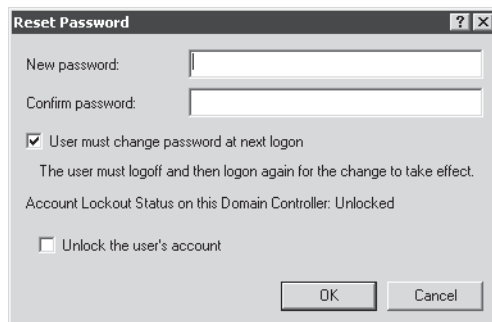


Figure 3-8 The Reset Password dialog box

You can also use a DS command to reset a user's password and, optionally, to force the user to change that password at the next logon. Type the following command:

```
dsmod user UserDN &pwd NewPassword -mustchpwd yes
```

Using Windows PowerShell, type the following commands:

```
$objUser=[ADSI]"LDAP://UserDN"  
$objUser.SetPassword("NewPassword")
```

Note that, unlike other attributes, you do not use *SetInfo* after using *SetPassword* to configure the user's password. However, if you want to force the user to change passwords at the next logon, you do as follows:

```
$objUser.Put ("pwdLastSet",0)  
$objUser.SetInfo()
```

In VBScript, the code is very similar:

```
Set objUser=GetObject("LDAP://UserDN")  
objUser.SetPassword "NewPassword"  
objUser.Put "pwdLastSet",0  
objUser.SetInfo
```

It is even possible to import passwords, using *LDIFDE*, a command introduced in Lesson 1. See Knowledge Base article 263991 at <http://support.microsoft.com/default.aspx?scid=kb;en-us;263991> for information.

Unlocking a User Account

In Chapter 8, "Authentication," you will learn to configure password and account lockout policies. A lockout policy is designed to prevent an intruder from attempting to penetrate the enterprise network by logging on repeatedly with various passwords until he or she finds a correct password. When a user attempts to log on with an incorrect password, a logon failure is generated. When too many logon failures occur within a specified period of time defined by the lockout policy, the account is locked out. The next time the user attempts to log on, a notification clearly states the account lockout.

NOTE Watch for drives mapped with alternate credentials

A common cause of account lockout is a drive mapped with alternate credentials. If the alternate credentials' password is changed, and the Windows client attempts repeatedly to connect to the drive, that account will be locked out.

Your lockout policy can define a period of time after which a lockout account is automatically unlocked. But when a user is trying to log on and discovers he or she is locked out, it is likely he or she will contact the help desk for support. You can unlock a user account by right-clicking the account, choosing Properties, clicking the Account tab, and selecting the Unlock Account check box.

Windows Server 2008 also adds the option to unlock a user's account when you choose the *Reset Password* command. Select the *Unlock The User's Account* check box, shown in Figure 3-8. This method is particularly handy when a user's account has become locked out because the user did, in fact, forget the password. You can now assign a new password, specify that the user must change the password at next logon, and unlock the user's account in one dialog box.

Unfortunately, neither the command line nor Windows PowerShell provides a native tool for unlocking accounts. To unlock a user with VBScript, use the following code:

```
Set objUser = GetObject("LDAP://UserDN")
objUser.IsAccountLocked = False
objUser.SetInfo
```

Disabling and Enabling a User Account

User accounts are security principals—identities that can be given access to network resources. Because each user is a member of Domain Users and of the Authenticated Users special identity, each user account has at least read access to a vast amount of information in Active Directory and on your file systems unless you have been severe and unusually successful at locking down access control lists (ACLs).

Therefore, it is important not to leave user accounts open. That means you should configure password policies and auditing—both discussed in Chapter 8—and procedures to ensure that accounts are being used appropriately. If a user account is provisioned before it is needed, or if an employee will be absent for an extended period of time, disable the account.

To disable an account in the Active Directory Users And Computers snap-in, right-click a user and choose *Disable*. From the command line, you can use *Dsmod.exe*, as in the following example:

```
dsmod user UserDN /d:disabled yes
```

With Windows PowerShell, as you learned in Lesson 2, you must use a roundabout method to set the flag:

```
$objUser=[ADSI]"LDAP://UserDN"
$objUser.psbase.InvokeSet('Account Disabled', $true)
$objUser.SetInfo()
```

VBScript is more straightforward:

```
Set objUser = GetObject("LDAP://UserDN")
objUser.AccountDisabled=TRUE
```

Enabling an account is just a matter of *yes* to *no* for the *Dsmod.exe* command:

```
dsmod user UserDN /d:disabled no
```

In the Windows PowerShell commands shown earlier, change *\$true* to *\$false* and, in VBScript, change *TRUE* to *FALSE*.

Deleting a User Account

When an account is no longer necessary, you can delete it from your directory. However, it is critical to consider that after the account has been deleted, it is eventually purged entirely from the directory. You cannot simply re-create a new account with the same name as a deleted account and hope it has the same group memberships and access to resources; it will not. The loss of the user's SID and of its group memberships can cause significant problems if, later, you realize you need the account.

Therefore, many organizations choose to deprovision a user account in stages. First, the account is disabled. After a period of time, it is deleted. Active Directory actually maintains a subset of the account's properties—most notably its SID—for a period of time called the *tombstone lifetime*, 60 days by default. After that time, the account's record is removed from the directory.

You can also consider recycling a user account. If a user leaves your organization, it's possible you will eventually hire a replacement who will need very similar resource access, group memberships, and user rights as the previous user. You can disable the account until a replacement is found and then rename the account to match the new user's name. The previous user's SID, group memberships, and resource access are thereby transferred to the replacement.

To delete a user account in Active Directory, select the user and press Delete or right-click the user and choose Delete. You will be prompted to confirm your choice because of the significant implications of deleting a security principal.

You can delete objects from Active Directory by using the *Dsrm* command, another of the DS commands. *Dsrm* uses a simple syntax:

```
dsrm UserDN
```

Notice that *Dsrm* is not followed by the *user* object class as are the other DS commands.

To delete a user from Active Directory, using Windows PowerShell, you connect to the parent container—the OU—and use the container's *Delete* method. This might seem slightly strange, but it parallels the fact that you use the container's *Create* method to create a user. The following two Windows PowerShell commands will delete a user:

```
$objOU = [ADSI]"LDAP://organizational unit's DN"  
$objOU.Delete("user", "CN=UserCN")
```

VBScript uses the same approach, with its unique syntax:

```
Set objOU = GetObject(LDAP://organizational unit's DN")  
objOU.Delete "user", "CN=UserCN"
```

Moving a User Account

If you need to move a user object in Active Directory, you can drag and drop it in the Active Directory Users and Computers snap-in. However, it is more accurate to right-click the user and choose the *Move* command. Keep in mind that when you move a user, you might change the Group Policy objects (GPOs) that apply to that user. GPOs are discussed in Chapter 6, “Group Policy Infrastructure.”

To move a user with a command-line tool, use *Dsmove*. *Dsmove* uses the following syntax:

```
dsmove UserDN Dnewparent TargetOUDN
```

Dsmove does not specify a *user* object class. Instead, it simply indicates the DN of the user to move and, in the *TargetOUDN* placeholder, the distinguished name of the OU to which the user will be moved.

To move a user in Windows PowerShell, you must use the *psbase.MoveTo* method. The following two lines of code will move a user:

```
$objUser=[ADSI]"LDAP://UserDN"  
$objUser.psbase.MoveTo("LDAP://TargetOUDN")
```

This is another example of a workaround required because this version of Windows PowerShell does not deliver an Active Directory provider. Some day in the future, you will be able to use the *Move-Item* cmdlet as you can with the file system and registry providers, but not yet.

In VBScript, you use an approach that seems a bit backward. You connect to the target container and then you grab the user object and move it to the container. The following two lines of code do the trick:

```
Set objOU = GetObject("LDAP://TargetOUDN")  
objOU.MoveHere "LDAP://UserDN", vbNullString
```

The intrinsic constant *vbNullString* passes *Null* to the *MoveHere* method, instructing it that you want the object to keep its current CN.

Renaming a User Account

In the “User Object Names” section, you learned about many of the names associated with a user account. When a user account needs to be renamed, there can be one or more attributes you must change. To rename a user in Active Directory, right-click the user and choose *Rename*. Type the new common name (CN) for the user and press Enter. The *Rename User* dialog box appears and prompts you to enter the Full Name (which maps to the *cn* and *name* attributes), First Name, Last Name, Display Name, User Logon Name, and User Logon Name (Pre-Windows 2000).

From a command prompt, you can use *Dsmod.exe* with the following syntax:

```
dsmod user UserDN [-upn UPN][-fn FirstName][-mi Initial][-ln LastName]  
[-dn DisplayName][-email EmailAddress]
```

Chapter 4

Groups

Although users and computers, and even services, change over time, business roles and rules tend to remain more stable. Your business probably has a finance role, which requires certain capabilities in the enterprise. The user or users who perform that role will change, but the role will remain. For that reason, it is not practical to manage an enterprise by assigning rights and permissions to individual user, computer, or service identities. Management tasks should be associated with groups. In this training kit, you will use groups to identify administrative and user roles, to filter Group Policy, to assign unique password policies, to assign rights and permissions, and more. To prepare for those tasks, in this lesson you will learn how to create, modify, delete, and support group objects in an Active Directory Domain Services (AD DS) domain.

Exam objectives in this chapter:

- Creating and Maintaining Active Directory Objects
 - Automate creation of Active Directory accounts.
 - Maintain Active Directory accounts.

Before You Begin

This chapter applies Microsoft Windows PowerShell, Microsoft VBScript, Comma-Separated Values Data Exchange (CSVDE), and LDAP Data Interchange Format Data Exchange (LDIFDE) to the task of automating computer account creation. Read Lesson 1, “Automating the Creation of User Accounts,” and Lesson 2, “Creating Users with Windows PowerShell and VBScript,” of Chapter 3, “Users,” prior to reading this chapter.

In addition, to perform exercises in this chapter, you must have created a domain controller named SERVER01 in a domain named *contoso.com*. See Chapter 1, “Installation,” for detailed steps for this task.

Real World

Dan Holme

Efficient and effective group management is a tremendous enabler for security, consistency, and productivity in an IT environment. As a consultant, I spend a lot of time with clients, aligning technology with their business needs. In the case of Microsoft Windows technologies, that entails defining and implementing business roles and rules so that administration can be defined, documented, and automated. And that process often requires improving clients' group management knowledge, technologies, and processes. Many IT professionals have come into Windows Server 2008 Active Directory with former practices that do not take advantage of groups as fully as possible. In fact, I've seen so much wasted productivity and decreased security due to poor group management that I dedicated two chapters of my book, *Windows Administration Resource Kit: Productivity Solutions for IT Professionals* (Microsoft Press, 2008), to improving and automating group management. In this lesson, you will learn what you need to know for the certification exam, and I share with you a few of the tips and best practices you'll need to make the most of groups in a production environment. I highly recommend reading the resource kit for more information, guidance, and fantastic tools related to group management.

Lesson 1: Creating and Managing Groups

You are certainly familiar with the purpose of groups: to collect items and manage them as a single entity. The implementation of group management in Active Directory is not intuitive because Active Directory is designed to support large, distributed environments, so it includes seven types of groups: two types of domain groups with three scopes each and local security groups. In this lesson, you will learn the purpose each of these groups serves, and you'll learn to align your business requirements with the potentially complex options that Active Directory provides.

After this lesson, you will be able to:

- Create groups by using the Active Directory Users and Computers snap-in.
- Manage and convert group type and scope.
- Identify the types of objects that can be members of groups of various scopes.
- Manage group membership.
- Develop a group management strategy.

Estimated lesson time: 45 minutes

Managing an Enterprise with Groups

Groups are security principals with a security identifier (SID) that, through their *member* attribute, collect other security principals (users, computers, contacts, and other groups) to facilitate management.

Imagine that all 100 users in the sales department require read-level access to a shared folder on a server: It is not manageable to assign permissions to each user individually. When new salespeople are hired, you will have to add the new accounts to the access control list (ACL) of the folder. When accounts are deleted, you will have to remove the permissions from the ACL, or else you will be left with the missing account entry on the ACL, shown in 4-1, which results from a SID on the ACL that refers to an account that cannot be resolved. Imagine now that all 100 users in the sales department require access to 10 shared folders on three servers. The management challenges just increased significantly.

You have, no doubt, learned that although assigning permissions to a resource to an individual identity—user or computer—is possible, the best practice is to assign a single permission to a group and then to manage access to the resource simply by changing membership of the group.

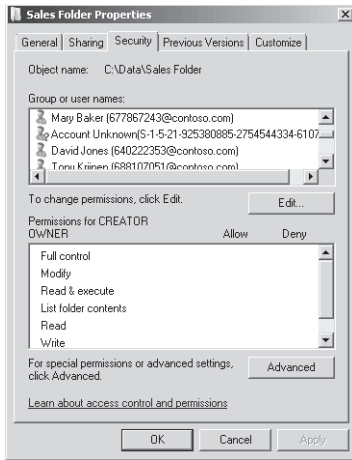


Figure 4-1 An ACL with a SID that refers to an account that can no longer be resolved

So, to continue the example, you could create a group called Sales and assign the group the Allow Read permission on the 10 shared folders on the three servers. Now, you have a *single point of management*. The Sales group effectively manages access to the shared folder. You can add new sales users to the group, and they will gain access to the 10 shared folders. When you delete an account, it is automatically deleted from the group, so you will not have irresolvable SIDs on your ACLs. There’s an extra benefit also: because your ACL will remain stable with the Sales group having Allow Read permission, your backups will be easier. When you change the ACL of a folder, the ACL propagates to all child files and folders, setting the Archive flag and thereby requiring a backup of all files, even if the contents of the files have not changed.

Imagine, now, that it is not only salespeople who require read access to the folders. Marketing department employees and the sales consultants hired by your organization also require Read permission to the same folders. You could add those groups to the ACL of the folders, but soon you will end up with an ACL with multiple permissions, this time assigning the Allow Read permission to multiple groups instead of multiple users. To give the three groups permission to the 10 folders on the three servers, you will have to add 30 permissions! The next group that requires access will require 10 more changes to grant permissions to the ACLs of the 10 shared folders. What if eight users, who are not salespeople, marketing employees, or consultants, have business need for Read access to the 10 folders? Do you add their individual user accounts to the ACLs?

You can see quickly that using only one type of group—a group that defines the business roles of users—is not enabling effective management of access to the 10 folders. The solution is to recognize that two types of management must exist to manage this scenario effectively. You must manage the users as collections, based upon their business roles, and you must manage access to the 10 folders. The 10 folders are also a collection of items: They are a single resource

that just happens to be distributed across 10 folders on three servers. You are trying to manage Read access to that resource collection. You need a single point of management with which to manage access to the resource collection.

This requires another group—a group that represents read access to the 10 folders on the three servers. Imagine that a group is created called ACL_Sales Folders_Read. This group will be assigned the Allow Read permission on the 10 folders. The sales, marketing, and consultants groups, along with the eight individual users, will all be members of the ACL_Sales Folders_Read group. As additional groups or users require access to the folders, they will be added to that group. It also becomes much easier to report who has access to the folders. Instead of having to examine the ACLs on each of the 10 folders, you simply examine the membership of the ACL_Sales Folders Read group.

This approach to managing the enterprise with groups is called *role-based management*. You define roles of users based on business characteristics—for example, department or division affiliation such as sales, marketing, and consultants—and you reflect your business rules such as which roles and individuals can access the 10 folders.

You can achieve both management tasks, using groups in a directory. Roles are represented by groups that contain users, computers, and other roles. That's right—roles can include other roles. For example, a Managers role might include the Sales Managers, Finance Managers, and Production Managers roles. Rules, such as the rule that defines Read access to the 10 folders, are represented by groups as well. Rule groups contain role groups and, occasionally, individual users or computers such as the eight users in the example.

To achieve manageability of an enterprise of any size or complexity, you will need to manage groups effectively and have an infrastructure of groups that provide single points of management for roles and rules. That means, technically, that you will need groups that can include as members users, computers, other groups, and, possibly, security principals from other domains.

For more information about role-based management, see *Windows Administration Resource Kit: Productivity Solutions for IT Professionals*.

Defining Group Naming Conventions

To create a group by using the Active Directory Users And Computers snap-in, simply right-click the OU in which you want to create a group, choose New, and select Group. The New Object – Group dialog box, shown in Figure 4-2, enables you to specify fundamental properties of the new group.

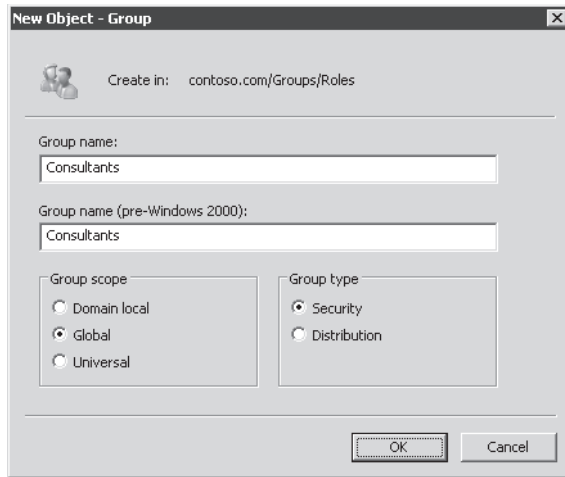


Figure 4-2 The New Object – Group dialog box

The first properties you must configure are the group's names. A group, like a user or computer, has several names. The first, shown in the Group Name box in Figure 4-2, is used by Windows 2000 and later systems to identify the object—it becomes the *cn* and *name* attributes of the object. The second, the pre-Windows 2000 name, is the *sAMAccountName* attribute, used to identify the group to computers running Microsoft Windows NT 4.0 and to some devices such as network attached storage (NAS) devices running non-Microsoft operating systems. The *cn* and *name* attributes must be unique only within the container—the OU—in which the group exists. The *sAMAccountName* must be unique in the entire domain. Technically, the *sAMAccountName* could be a different value than the *cn* and *name*, but it is highly discouraged to do so. Pick a name that is unique in the domain and use it in both name fields in the New Object – Group dialog box.

The name you choose should help you manage the group and manage your enterprise on a day-to-day basis. It is recommended to follow a naming convention that identifies the type of group and the purpose of the group. The example in the previous section used a group name, *ACL_Sales Folder_Read*. The prefix indicates that the group is used to assign permissions to a folder: It is used on access control lists. The main part of the name describes the resource that is being managed with the group: the sales folder. The suffix further defines what is being managed by the group: read access. A delimiter—in this case, an underscore—is used to separate parts of the name. Note that the delimiter is not used between the words *Sales* and *Folder*. Spaces are acceptable in group names—you will just need to enclose group names in quotes when you refer to them on command lines. You can create scripts that use the delimiter to deconstruct group names to facilitate auditing and reporting. Keep in mind that role groups that define user roles will often be used by nontechnical users. For example, you might e-mail enable the Sales group so that it can be used as an e-mail distribution list. Therefore, it is recommended that you do not use prefixes on role group names—keep the names user-friendly and descriptive.

For more information about managing groups effectively, see *Windows Administration Resource Kit: Productivity Solutions for IT Professionals*.

Understanding Group Types

There are two types of groups: security and distribution. When you create a group, you make the selection of the group type in the New Object – Group dialog box.

Distribution groups are used primarily by e-mail applications. These groups are not security enabled; they do not have SIDs, so they cannot be given permissions to resources. Sending a message to a distribution group sends the message to all members of the group.

Security groups are security principals with SIDs. These groups can, therefore, be used as permission entries in ACLs to control security for resource access. Security groups can also be used as distribution groups by e-mail applications. If a group will be used to manage security, it must be a security group.

Because security groups can be used for both resource access and e-mail distribution, many organizations use only security groups. However, it is recommended that if a group will be used only for e-mail distribution, you should create the group as a distribution group. Otherwise, the group is assigned a SID, and the SID is added to the user's security access token, which can lead to unnecessary token bloat.

Understanding Group Scope

Groups have members: users, computers, and other groups. Groups can be members of other groups, and groups can be referred to by ACLs, Group Policy object (GPO) filters, and other management components. *Group scope* affects each of these characteristics of a group: what it can contain, what it can belong to, and where it can be used. There are four group scopes: global, domain local, local, and universal.

The characteristics that define each scope fall into these categories:

- **Replication** Where is the group defined and to what systems is the group replicated?
- **Membership** What types of security principals can the group contain as members? Can the group include security principals from trusted domains?

In Chapter 12, “Domains and Forests,” you will learn about trust relationships, or *trusts*. A trust enables a domain to refer to another domain for user authentication, to include security principals from the other domain as group members, and to assign permissions to security principals in the other domain. The terminology used can be confusing. If Domain A trusts Domain B, then Domain A is the *trusting* domain and Domain B is the *trusted* domain. Domain A accepts the credentials of users in Domain B. It forwards requests by Domain B users to authenticate to a domain controller in Domain B because it *trusts* the identity store and authentication service of Domain B. Domain A can add

Domain B's security principals to groups and ACLs in Domain A. See Chapter 12 for more detail.

Exam Tip In the context of group membership, remember that if Domain A trusts Domain B, Domain B is *trusted*, and its users and global groups can be members of domain local groups in Domain A. Additionally, Domain B's users and global groups can be assigned permissions to resources in Domain A.

- **Availability** Where can the group be used? Is the group available to add to another group? Is the group available to add to an ACL?

Keep these broad characteristics in mind as you explore the details of each group scope.

Local Groups

Local groups are truly local—defined on and available to a single computer. Local groups are created in the security accounts manager (SAM) database of a domain member computer. Both workstations and servers have local groups. In a workgroup, you use local groups to manage security of resources on a system. In a domain, however, managing the local groups of individual computers becomes unwieldy and is, for the most part, unnecessary. It is not recommended to create custom local groups on domain members. In fact, the Users and Administrators local groups are the only local groups that you should be concerned with managing in a domain environment. To summarize:

- **Replication** A local group is defined only in the local SAM database of a domain member server. The group and its membership are not replicated to any other system.
- **Membership** A local group can include as members:
 - Any security principals from the domain: users, computers, global groups, or domain local groups.
 - Users, computers, and global groups from any domain in the forest.
 - Users, computers, and global groups from any trusted domain.
 - Universal groups defined in any domain in the forest.
- **Availability** A local group has only computer-wide scope. It can be used in ACLs on the local computer only. A local group cannot be a member of any other group.

Domain Local Groups

Domain local groups are used primarily to manage permissions to resources. For example, the ACL_Sales Folder_Read group discussed earlier in the lesson would be created as a domain local group. Domain local groups have the following characteristics:

- **Replication** A domain local group is defined in the domain naming context. The group object and its membership (the *member* attribute) are replicated to every domain controller in the domain.
- **Membership** A domain local group can include as members:
 - Any security principals from the domain: users, computers, global groups, or other domain local groups.
 - Users, computers, and global groups from any domain in the forest.
 - Users, computers, and global groups from any trusted domain.
 - Universal groups defined in any domain in the forest.
- **Availability** A domain local group can be added to ACLs on any resource on any domain member. Additionally, a domain local group can be a member of other domain local groups or even computer local groups.

The membership capabilities of a domain local group are identical to those of local groups, but the replication and availability of the domain local group makes it useful across the entire domain. The domain local group is, therefore, well suited for defining business management rules, such as access rules, because the group can be applied anywhere in the domain, and it can include members of any type within the domain and members from trusted domains as well.

Global Groups

Global groups are used primarily to define collections of domain objects based on business roles. Role groups, such as the Sales and Marketing groups mentioned earlier, as well as roles of computers such as a Sales Laptops group, will be created as global groups. Global groups have the following characteristics:

- **Replication** A global group is defined in the domain naming context. The group object, including the *member* attribute, is replicated to all domain controllers in the domain.
- **Membership** A global group can include as members users, computers, and other global groups in the same domain only.
- **Availability** A global group is available for use by all domain members as well as by all other domains in the forest and all trusting external domains. A global group can be a member of any domain local or universal group in the domain or in the forest. It can also be a member of any domain local group in a trusting domain. Finally, a global group can be added to ACLs in the domain, in the forest, or in trusting domains.

As you can see, global groups have the most limited membership (only users, computers, and global groups from the same domain) but the broadest availability across the domain, the forest, and trusting domains. That is why they are well suited to defining roles, because roles are generally collections of objects from the same directory.

Universal Groups

Universal groups are useful in multidomain forests. They enable you to define roles, or to manage resources, that span more than one domain. The best way to understand universal groups is through an example. Trey Research has a forest with three domains: Americas, Asia, and Europe. Each domain has user accounts and a global group called Regional Managers that includes the managers of that region. Remember that global groups can contain only users from the same domain. A universal group called Trey Research Regional Managers is created, and the three Regional Managers groups are added as members. The Trey Research Regional Managers group, therefore, defines a role for the entire forest. As users are added to any one of the Regional Managers groups, they will, through group nesting, be a member of the Trey Research Regional Managers.

Trey Research is planning to release a new product that requires collaboration across its regions. Resources related to the project are stored on file servers in each domain. To define who can modify files related to the new product, a universal group is created called *ACL_New Product_Modify*. That group is assigned the Allow Modify permission to the shared folders on each of the file servers in each of the domains. The Trey Research Regional Managers group is made a member of the *ACL_New Product_Modify* group, as are various global groups and a handful of users from each of the regions.

As you can see from this example, universal groups can help you represent and consolidate groups that span domains in a forest and help you define rules that can be applied across the forest. Universal groups have the following characteristics:

- **Replication** A universal group is defined in a single domain in the forest but is replicated to the global catalog. You will learn more about the global catalog in Chapter 10, “Domain Controllers.” Objects in the global catalog will be readily accessible across the forest.
- **Membership** A universal group can include as members users, global groups, and other universal groups from any domain in the forest.
- **Availability** A universal group can be a member of a universal group or domain local group anywhere in the forest. Additionally, a universal group can be used to manage resources, for example, to assign permissions anywhere in the forest.

Summarizing Group Membership Possibilities

Both on the 70-640 examination and in day-to-day administration, it is important for you to be completely familiar with the membership characteristics of each group scope.

Table 4-1 summarizes the objects that can be members of each group scope.

Table 4-1 Group Scope and Members

Group Scope	Members from the same domain	Members from another domain in the same forest	Members from a trusted external domain
Local	Users Computers Global groups Universal groups Domain local groups Local users defined on the same computer as the local group	Users Computers Global groups Universal groups	Users Computers Global groups
Domain Local	Users Computers Global groups Domain local groups Universal groups	Users Computers Global groups Universal groups	Users Computers Global groups
Universal	Users Computers Global groups Universal groups	Users Computers Global groups Universal groups.	N/A
Global	Users Computers Global groups	N/A	N/A

Quick Check

- Which types of objects can be members of a global group in a domain?

Quick Check Answer

- Global groups can contain only users, computers, and other global groups from the same domain.

Converting Group Scope and Type

If, after creating a group, you determine that you need to modify the group's scope or type, you can do so. Open the Properties dialog box of an existing group and, on the General tab, shown in Figure 4-3, you will see the existing scope and type. At least one more scope and type are available to be selected.

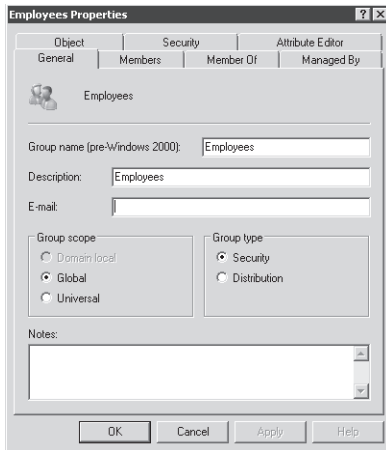


Figure 4-3 The General tab of a group's Properties dialog box

You can convert the group type at any time by changing the selection in the Group Type section of the General tab. Be cautious, however. When you convert a group from security to distribution, any resources to which the group had been assigned permission will no longer be accessible in the same way. After the group becomes a distribution group, users who log on to the domain will no longer include the group's SID in their security access tokens.

You can change the group scope in one of the following ways:

- Global to universal
- Domain local to universal
- Universal to global
- Universal to domain local

The only scope changes that you cannot make directly are from global to domain local or domain local to global. However, you can make these changes indirectly by first converting to universal scope and then converting to the desired scope, so all scope changes are possible.

Remember, however, that a group's scope determines the types of objects that can be members of the group. If a group already contains members or is a member of another group, you will be prevented from changing scope. For example, if a global group is a member of another global group, you cannot change the first group to universal scope, because a universal group cannot be a member of a global group. An explanatory error message will display such as that shown in Figure 4-4. You must correct the membership conflicts before you can change the group's scope.

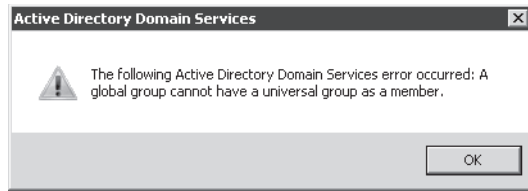


Figure 4-4 The error produced when a group's membership will not allow a change of scope

The *Dsmod* command, introduced in Chapter 3, can be used to change group type and scope by using the following syntax:

```
dsmod group GroupDN /secgrp { yes | no } /scope { l | g | u }
```

The *GroupDN* is the distinguished name of the group to modify. The following two parameters affect group scope and type:

- **-secgrp { yes | no }** specifies group type: security (*yes*) or distribution (*no*).
- **-scope { l | g | u }** determines the group scope: domain local (*l*), global (*g*), or universal (*u*).

Managing Group Membership

When you need to add or remove members of a group, you have several methods by which to do so. First, you can open the group's Properties dialog box and click the Members tab. To remove a member, simply select the member and click Remove. To add a member, click the Add button. The Select Users, Computers, Or Groups dialog box appears, as shown in Figure 4-5.

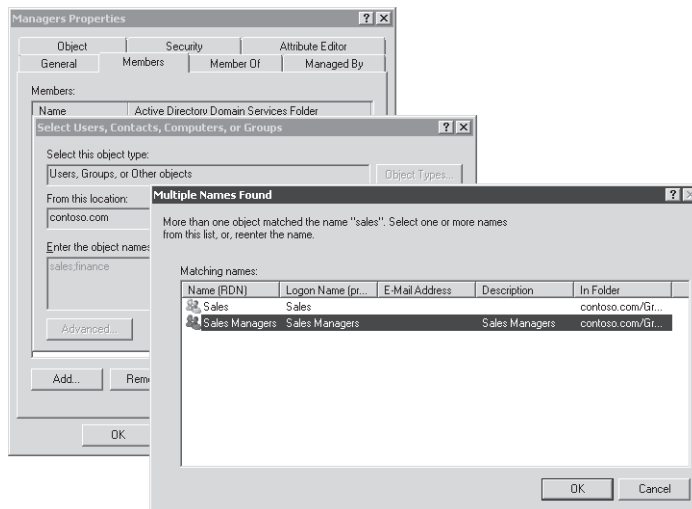


Figure 4-5 Adding a member to a group

Several tips are worth mentioning about this process:

- In the Select dialog box, in the Enter The Object Names box, you can type multiple accounts separated by semicolons. For example, in Figure 4-5, both *sales* and *finance* were entered. They are separated by a semicolon.
- You can type partial names of accounts—you do not need to type the full name. Windows searches Active Directory for accounts that begin with the name you entered. If there is only one match, Windows selects it automatically. If there are multiple accounts that match, the Multiple Names Found dialog box appears, enabling you to select the specific object you want. This shortcut—typing partial names—can save time adding members to groups and can help when you don't remember the exact name of a member.
- By default, Windows searches only for users and groups that match the names you enter in the Select dialog box. If you want to add computers to a group, you must click the Options button and select Computers.
- By default, Windows searches only domain groups. If you want to add local accounts, click the Locations button in the Select dialog box.
- If you cannot find the member you want to add, click the Advanced button in the Select dialog box. A more powerful query window will appear, giving you more options for searching Active Directory.

You can also add an object to a group in the Active Directory Users And Computers snap-in by opening the properties of the object and clicking its Member Of tab. Click the Add button and select the group. Similarly, you can right-click one or more selected objects and use the *Add To Group* command.

The *Member* and *MemberOf* Attributes

When you add a member to a group, you change the group's *member* attribute. The *member* attribute is a multivalued attribute. Each member is a value represented by the distinguished name (DN) of the member. If the member is moved or renamed, Active Directory automatically updates the *member* attributes of groups that include the member.

When you add a member to a group, the member's *memberOf* attribute is also updated, indirectly. The *memberOf* attribute is a special type of attribute called a *backlink*. It is updated by Active Directory when a forward link attribute, such as *member*, refers to the object. When you add a member to a group, you are always changing the *member* attribute. Therefore, when you use the Member Of tab of an object to add to a group, you are actually changing the group's *member* attribute. Active Directory updates the *memberOf* attribute automatically.

Helping Membership Changes Take Effect Quickly

When you add a user to a group, the membership does not take effect immediately. Group membership is evaluated at logon for a user (at startup for a computer). Therefore, a user will have to log off and log on before the membership change becomes a part of the user's token.

Additionally, there can be a delay while the group membership change replicates. Replication will be discussed in Chapter 11, "Sites and Replication." This is particularly true if your enterprise has more than one Active Directory site. You can facilitate the speed with which a change affects a user by making the change on a domain controller in the user's site. Right-click the domain in the Active Directory Users And Computers snap-in and choose Change Domain Controller.

Developing a Group Management Strategy

Adding groups to other groups—a process called *nesting*—can create a hierarchy of groups that support your business roles and rules. Now that you have learned the business purposes and technical characteristics of groups, it is time to align the two in a strategy for group management.

Earlier in this lesson, you learned which types of objects *can* be members of each group scope. Now it is time to identify which types of objects *should* be members of each group scope. This leads to the best practice for group nesting, known as AGDLA:

- Accounts (user and computer identities) are members of
- Global groups that represent business roles. Those role groups (global groups) are members of
- Domain Local groups that represent management rules—which have Read permission to a specific collection of folders, for example. These rule groups (domain local groups) are added to
- Access control lists (ACLs), which provide the level of access required by the rule.

In a multidomain forest, there are universal groups, as well, that fit in between global and domain local. Global groups from multiple domains are members of a single universal group. That universal group is a member of domain local groups in multiple domains. You can remember the nesting as AGUDLA.

This best practice for implementing group nesting translates well even in multidomain scenarios. Consider Figure 4-6, which represents a group implementation that reflects not only the technical view of group management best practices (AGDLA) but also the business view of role-based, rule-based management.

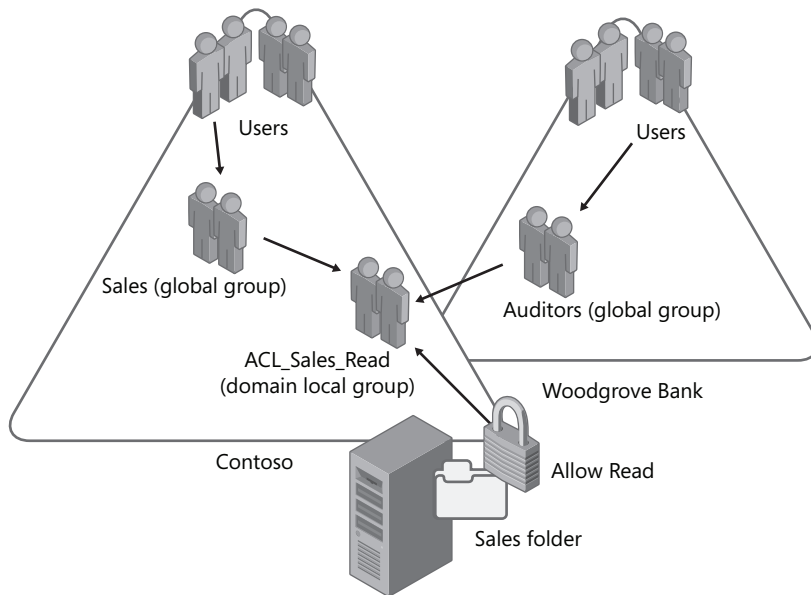


Figure 4-6 A group management implementation

Consider the following scenario. The sales force at Contoso, Ltd., has just completed their fiscal year. Sales files from the previous year are in a folder called Sales. The sales force needs read access to the Sales folder. Additionally, a team of auditors from Woodgrove Bank, a potential investor, require Read access to the Sales folder to perform the audit. The steps to implement the security required by this scenario are as follows:

1. Assign users with common job responsibilities or other business characteristics to role groups implemented as global security groups.
This happens separately in each domain. Sales people at Contoso are added to a Sales role group. Auditors at Woodgrove Bank are added to an Auditors role group.
2. Create a group to represent the business rule regarding who can access the Sales folder with Read permission.
This is implemented in the domain containing the resource to which the rule applies. In this case, it is the Contoso domain in which the Sales folder resides. The rule group is created as a domain local group.
3. Add the role groups to whom the business rule applies to the rule group.
These groups can come from any domain in the forest or from a trusted domain such as Woodgrove Bank. Global groups from trusted external domains, or from any domain in the same forest, can be a member of a domain local group.
4. Assign the permission that implements the required level of access.
In this case, grant the Allow Read permission to the domain local group.

This strategy results in single points of management, reducing the management burden. There is one point of management that defines who is in Sales or who is an Auditor. Those roles, of course, are likely to have a variety of permissions to resources beyond simply the Sales folder. There is another single point of management to determine who has Read access to the Sales folder. The Sales folder might not just be a single folder on a single server; it could be a collection of folders across multiple servers, each of which assigns Allow Read permission to the single domain local group.

Lesson 2: Automating the Creation and Management of Groups

In Lesson 1, you learned the steps for creating groups, choosing group scope and type, and configuring group membership, using the Active Directory Users and Computers snap-in. When you need to create more than one group at a time, or when you want to automate group creation, you must turn to other tools. Chapter 3 introduced you to command-line and automation tools, including *CSVDE*, *LDIFDE*, *Dsadd*, Windows PowerShell, and VBScript. These tools can also be used to automate the creation and management of group objects. In this lesson, you'll learn how to manage the life cycle of group objects, from birth to death, using command-line and automation tools.

After this lesson, you will be able to:

- Create groups with *Dsadd*, *CSVDE*, and *LDIFDE*.
- Modify groups' membership with *Dsmod*, *LDIFDE*, Windows PowerShell, and VBScript.
- Enumerate group membership with *Dsget*.
- Move and delete groups with *Dsmove* and *Dsrm*.

Estimated lesson time: 45 minutes

Creating Groups with *Dsadd*

The *Dsadd* command, introduced in Chapter 3, enables you to add objects to Active Directory. To add a group, type the command **dsadd group *GroupDN***, where *GroupDN* is the DN of the group, such as "CN=Finance Managers,OU=Groups,DC=contoso,DC=com." Be certain to surround the DN with quotes if the DN includes spaces. For example, to create a new global security group named Marketing in the Groups OU of the *contoso.com* domain, the command would be:

```
dsadd group "CN=Marketing,OU=Groups,DC=contoso,DC=com"  
    /s:amid /m:Marketing /o:secgrp /y:yes /s:scope g
```

You can also provide the *GroupDN* parameter by one of the following ways:

- By piping a list of DNs from another command such as *Dsquery*.
- By typing each DN on the command line, separated by spaces.
- By leaving the DN parameter empty, at which point you can type the DNs one at a time at the keyboard console of the command prompt. Press Enter after each DN. Press Ctrl + Z and Enter after the last DN.

Because you can include more than one DN on the command line, separated by a space, you can generate multiple groups at once with *Dsadd*. The *Dsadd* command can also configure group attributes of the groups you create with the following optional parameters:

- **-secgrp { yes | no }** specifies group type: security (yes) or distribution (no).
- **-scope { l | g | u }** determines the group scope: domain local (l), global (g), or universal (u).
- **-samid *Name*** specifies the *sAMAccountName* of the group. If not specified, the name of the group from its DN is used. It is recommended that the *sAMAccountName* and the group name be the same, so you do not need to include this parameter when using *Dsadd*.
- **-desc *Description*** configures the group's description.
- **-members *MemberDN*** adds members to the group. Members are specified by their DNs in a space-separated list.
- **-memberof *GroupDN* ...** makes the new group a member of one or more existing groups. The groups are specified by their DNs in a space-separated list.

Importing Groups with CSVDE

Chapter 3 also introduced you to *CSVDE*, which imports data from comma-separated values (.csv) files. It is also able to export data to a .csv file. The following example shows a .csv file that will create a group, Marketing, and populate the group with two initial members, Linda Mitchell and Scott Mitchell.

```
objectClass,sAMAccountName,DN,member
group,Marketing,"CN=Marketing,OU=Groups,DC=contoso,DC=com",
  "CN=Linda Mitchell,OU=People,DC=contoso,DC=com;CN=Scott Mitchell,
  OU=People,DC=contoso,DC=com"
```

The objects listed in the *member* attribute must already exist in the directory service. Their DNs are separated by semicolons within the *member* column.

You can import this file into Active Directory by using the command:

```
csvde -i -f "Filename" [-k]
```

The *-i* parameter specifies import mode. Without it, *CSVDE* uses export mode. The *-f* parameter precedes the filename, and the *-k* parameter ensures that processing continues even if errors are encountered.

Exam Tip *CSVDE* can be used to create objects, not to modify existing objects. You cannot use *CSVDE* to import members to existing groups.

Managing Groups with *LDIFDE*

LDIFDE, as you learned in Chapter 3, is a tool that imports and exports files in the Lightweight Directory Access Protocol Data Interchange Format (LDIF) format. LDIF files are text files within which operations are specified by a block of lines separated by a blank line. Each operation begins with the DN attribute of the object that is the target of the operation. The next line, *changeType*, specifies the type of operation: *add*, *modify*, or *delete*.

The following LDIF file creates two groups, Finance and Research, in the Groups OU of the *contoso.com* domain:

```
DN: CN=Finance,OU=Groups,DC=contoso,DC=com
changeType: add
CN: Finance
description: Finance Users
objectClass: group
sAMAccountName: Finance
```

```
DN: CN=Research,OU=Groups,DC=contoso,DC=com
changeType: add
CN: Research
description: Research Users
objectClass: group
sAMAccountName: Research
```

Convention would suggest saving the file with an *.ldf* extension, for example *Groups.ldf*. To import the groups into the directory, issue the *Ldifde.exe* command as shown here:

```
ldifde -i -f groups.ldf
```

Modifying Group Membership with *LDIFDE*

LDIFDE can also be used to modify existing objects in Active Directory, using LDIF operations with a *changeType* of *modify*. To add two members to the Finance group, the LDIF file would be:

```
dn: CN=Finance,OU=Groups,DC=contoso,DC=com
changetype: modify
add: member
member: CN=April Stewart,OU=People,dc=contoso,dc=com
member: CN=Mike Fitzmaurice,OU=People,dc=contoso,dc=com
-
```

The *changeType* is set to *modify*, and then the change operation is specified: *add* objects to the *member* attribute. Each new member is then listed on a separate line that begins with the *member* attribute name. The change operation is terminated with a line containing a single dash. Changing the third line to the following would remove the two specified members from the group:

```
delete: member
```

Retrieving Group Membership with *Dsget*

The *Dsmod* and *Dsget* commands discussed in Chapter 3 are particularly helpful for managing the membership of groups. There is no option in the Active Directory Users and Computers snap-in to list all the members of a group, including nested members. You can see only direct members of a group on the group's Members tab. Similarly, there is no way to list all the groups to which a user or computer belongs, including nested groups. You can see only direct membership on the user's or computer's Member Of tab.

The *Dsget* command enables you to retrieve a complete list of a group's membership, including nested members, with the following syntax:

```
dsget group "GroupDN" Dmembers [-expand]
```

The *expand* option performs the magic of expanding nested groups' members.

Similarly, the *Dsget* command can be used to retrieve a complete list of groups to which a user or computer belongs, again by using the *expand* option in the following commands:

```
dsget user "UserDN" Dmemberof [-expand]
dsget computer "ComputerDN" Dmemberof [-expand]
```

The *memberof* option returns the value of the user's or computer's *memberOf* attribute, showing the groups to which the object directly belongs. By adding the *expand* option, those groups are searched recursively, producing an exhaustive list of all groups to which object the user belongs in the domain.

Changing Group Membership with *Dsmod*

The *Dsmod* command was applied in Lesson 1 to modify the scope and type of a group. The command's basic syntax is:

```
dsmod group "GroupDN" [options]
```

You can use options such as *samid* and *desc* to modify the *sAMAccountName* and *description* attributes of the group. Most useful, however, are the options that enable you to modify a group's membership:

- **-addmbr "Member DN"** Adds members to the group
- **-rmmbr "Member DN"** Removes members from the group

As with all DS commands, *Member DN* is the distinguished name of another Active Directory object, surrounded by quotes if the DN includes spaces. Multiple *Member DN* entries can be included, separated by spaces. For example, to add Mike Danseglio to the Research group, the *Dsmod* command would be:

```
dsmod group "CN=Research,OU=Groups,DC=contoso,DC=com"
  -addmbr "CN=Mike Danseglio,OU=People,DC=contoso,DC=com"
```


You can use *Dsget* in combination with *Dsmod* to copy group membership. In the following example, the *Dsget* command is used to get information about all the members of the Sales group and then, by piping that list to *Dsmod*, to add those users to the Marketing group:

```
dsget group "CN=Sales,OU=Groups,DC=contoso,DC=com" %members |
dsmod group "CN=Marketing,OU=Groups,DC=contoso,DC=com" %addmbr
```

Moving and Renaming Groups with *Dsmove*

The *Dsmove* command, also discussed in Chapter 3, enables you to move or rename an object within a domain. You cannot use it to move objects between domains. Its basic syntax is:

```
dsmove ObjectDN [-newname NewName] [-newparent TargetOUDN]
```

The object is specified by using its distinguished name in the *ObjectDN* parameter. To rename the object, specify its new common name as the value of the *newname* parameter. To move an object to a new location, specify the distinguished name of the target container as the value of the *newparent* parameter.

For example, to change the name of the Marketing group to Public Relations, type:

```
dsmove "CN=Marketing,OU=Groups,DC=contoso,DC=com"
      %newname "Public Relations"
```

To then move that group to the Marketing OU, type:

```
dsmove "CN=Public Relations,OU=Groups,DC=contoso,DC=com"
      %newparent "OU=Marketing,DC=contoso,DC=com"
```

NOTE You're not limited to the command line

You can also move or rename a group in the Active Directory Users And Computers snap-in by right-clicking the group and choosing Move or Rename from the context menu.

Deleting Groups with *Dsrm*

Dsrm can be used to delete a group or any other Active Directory object. The basic syntax of *Dsrm* is:

```
dsrm ObjectDN ... [-subtree [-exclude]] [-noprompt] [-c]
```

The object is specified by its distinguished name in the *ObjectDN* parameter. You will be prompted to confirm the deletion of each object unless you specify the *noprompt* option. The *-c* switch puts *Dsrm* into continuous operation mode, in which errors are reported, but the command keeps processing additional objects. Without the *-c* switch, processing halts on the first error.

To delete the Public Relations group, type:

```
dsrm "CN=Public Relations,OU=Marketing,DC=contoso,DC=com"
```

You can also delete a group in the Active Directory Users And Computers snap-in by right-clicking the group and choosing the *Delete* command.

NOTE Know the impact before deleting a group

When you delete a group, you are removing a point of management in your organization. Be certain you have evaluated the environment to verify that there are no permissions or other resources that rely on the group. Deleting a group is a serious action with potentially significant consequences. It is recommended that, before you delete a group, you record its membership and remove all members for a period of time to determine whether the members lose access to any resources. If anything goes wrong, simply re-add the members. If the test succeeds, then delete the group.

Managing Group Membership with Windows PowerShell and VBScript

It is unlikely that you will need to understand the intricacies of managing group membership for the 70-640 examination, and an exhaustive discussion of scripting groups is beyond the scope of this book. See *Windows Administration Resource Kit: Productivity Solutions for IT Professionals* for detailed discussions about automating group management with VBScript.

However, it doesn't hurt to know the basics. In both VBScript and Windows PowerShell, there are several ways to manipulate group membership—a group's *member* attribute—but the most common and effective involve these steps:

1. Determine the *aDSPath* of the member. The *aDSPath* takes the form, *LDAP://<DN of member>*.
2. Connect to the group.
3. Use the *Add* or *Remove* method of the group object, specifying the *aDSPath* of the member.

A Windows PowerShell script that adds Mike Danseglio to the Research group would, therefore, be:

```
$MemberADSPath = "LDAP://CN=Mike Danseglio,OU=People,DC=contoso,DC=com"
$objGroup = [ADSI]"LDAP://CN=Research,OU=Groups,DC=contoso,DC=com"
$objGroup.Add ($MemberADSPath)
```

In VBScript, the script would be:

```
MemberADSPath = "LDAP://CN=Mike Danseglio,OU=People,DC=contoso,DC=com"
Set objGroup = GetObject("LDAP://CN=Research,OU=Groups,DC=contoso,DC=com")
objGroup.Add MemberADSPath
```

To remove members, use the *Remove* method instead of the *Add* method. The remainder of each script remains the same.

groups in the directory, thereby ensuring that permissions will be assigned to a group that is designed to manage resource access.

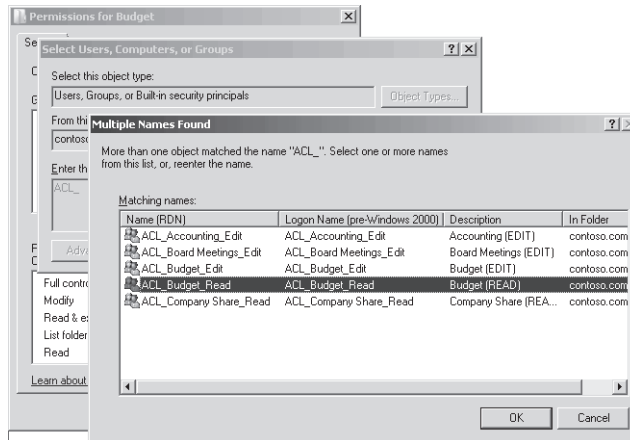


Figure 4-7 Selecting a group by using a group prefix to narrow down to the correct type of group

- **Summarize a group's purpose with its *description* attribute** Use the *description* attribute of a group to summarize the group's purpose. Because the Description column is enabled by default in the details pane of the Active Directory Users and Computers snap-in, the group's purpose can be highly visible to administrators.
- **Detail a group's purpose in its Notes** When you open a group's Properties dialog box, the *Notes* field, at the bottom of the General tab, can be used to document the group's purpose. For example, you can list the folders to which a group has been given permission, as shown in Figure 4-8.

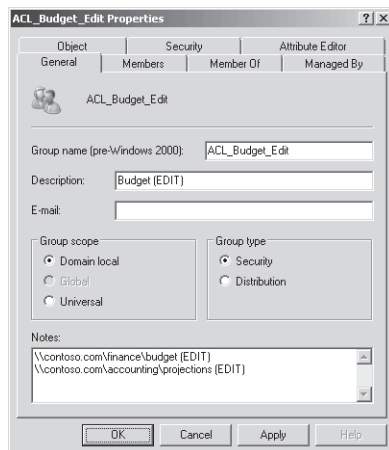


Figure 4-8 A group's Properties dialog box, showing the *Notes* field used to detail the group's purpose

Protecting Groups from Accidental Deletion

Deleting a group has a high impact on administrators and, potentially, on security. Consider a group that has been used to manage access to resources. If the group is deleted, access to that resource is changed. Either users who should be able to access the resource are suddenly prevented from access, creating a denial-of-service scenario, or if you had used the group to deny access to a resource with a Deny permission, inappropriate access to the resource becomes possible.

Additionally, if you re-create the group, the new group object will have a new SID, which will not match the SIDs on ACLs of resources. Instead, you must perform object recovery to reanimate the deleted group before the tombstone interval is reached. When a group has been deleted for the tombstone interval—60 days by default—the group and its SID are permanently deleted from Active Directory. When you reanimate a tombstoned object, you must re-create most of its attributes, including, significantly, the *member* attribute of group objects. That means you must rebuild the group membership after restoring the deleted object. Alternatively, you can perform an authoritative restore or, in Windows Server 2008, turn to your Active Directory snapshots to recover both the group and its membership. Authoritative restore and snapshots are discussed in Chapter 13, “Maintenance, Backup, and Recovery.”

MORE INFO Recovering deleted groups

You can learn more about recovering deleted groups and their memberships in Knowledge Base article 840001, which you can find at <http://support.microsoft.com/kb/840001/en-us>.

In any event, it is safe to say that recovering a deleted group is a skill you should hope to use only in disaster recovery fire drills, not in a production environment. Protect yourself from the potentially devastating results of group object deletion by protecting each group you create from deletion. Windows Server 2008 makes it easy to protect any object from accidental deletion. To protect an object, follow these steps:

1. In the Active Directory Users And Computers snap-in, click the View menu and make sure that Advanced Features is selected.
2. Open the Properties dialog box for a group.
3. On the Object tab, select the Protect Object From Accidental Deletion check box.
4. Click OK.

This is one of the few places in Windows where you actually have to click OK. Clicking Apply does not modify the ACL based on your selection.

The Protect Object From Accidental Deletion option applies an access control entry (ACE) to the ACL of the object that explicitly denies the Everyone group both the Delete permission and the Delete Subtree permission. If you really do want to delete the group, you can return to

the Object tab of the Properties dialog box and clear the Protect Object From Accidental Deletion check box.

Delegating the Management of Group Membership

After a group has been created, you might want to delegate the management of the group's membership to a team or an individual who has the business responsibility for the resource that the group manages. For example, assume that your finance manager is responsible for creating next year's budget. You create a shared folder for the budget and assign Write permission to a group named *ACL_Budget_Edit*. If someone needs access to the budget folder, he or she contacts the help desk to enter a request, the help desk contacts the finance manager for business approval, and then the help desk adds the user to the *ACL_Budget_Edit* group. You can improve the responsiveness and accountability of the process by allowing the finance manager to change the group's membership. Then, users needing access can request access directly from the finance manager, who can make the change, removing the intermediate step of the help desk. To delegate the management of a group's membership, you must assign to the finance manager the Allow Write Member permission for the group. The *member* attribute is the multivalued attribute that is the group's membership. There are several ways to delegate the Write Member permission. Two of them are covered in the following sections.

Delegating Membership Management with the Managed By Tab

The easiest way to delegate membership management of a single group is to use the Managed By tab. The Managed By tab of a group object's Properties dialog box, shown in Figure 4-9, serves two purposes. First it provides contact information related to the manager of a group. You can use this information to contact the business owner of a group to obtain approval prior to adding a user to the group.

The second purpose served by the Managed By tab is to manage the delegation of the *member* attribute. Note the check box shown in Figure 4-9. It is labeled Manager Can Update Membership List. When selected, the user or group shown in the Name box is given the WriteMember permission. If you change or clear the manager, the appropriate change is made to the group's ACL.

NOTE Click OK

This is another of the strange and rare places where you must actually click OK to implement the change. Clicking Apply does not change the ACL on the group.

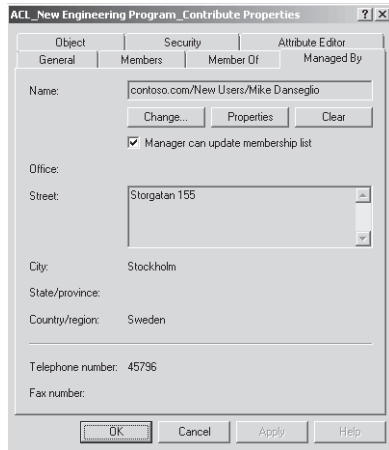


Figure 4-9 The Managed By tab of a group's Properties dialog box

It is not quite so easy to insert a group into the Managed By tab of another group. When you click the Change button, the Select User, Contact, Or Group dialog box appears, shown in Figure 4-10. If you enter the name of a group and click OK, an error occurs. That's because this dialog box is not configured to accept groups as valid object types, even though *Group* is in the name of the dialog box itself. To work around this odd limitation, click the Object Types button, and then select the check box next to Groups. Click OK to close both the Object Types and Select dialog boxes. Be sure to select the Manager Can Update Membership List check box if you want to assign the WriteMember permission to the group. When a group is used on the Managed By tab, no contact information is visible because groups do not maintain contact-related attributes.

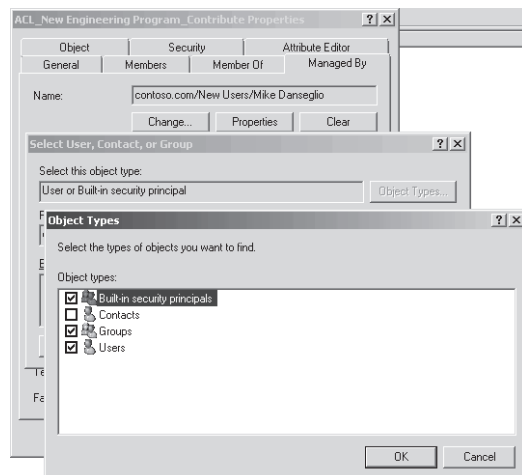


Figure 4-10 Selecting a group for the Managed By tab

Delegating Membership Management Using Advanced Security Settings

You can use the Advanced Security Settings dialog box to assign the Allow Write Member permission directly. You can assign the permission for an individual group or for all the groups in an OU.

Delegate the management of membership for an individual group

1. In the Active Directory Users And Computers snap-in, click the View menu and make sure Advanced Features is selected.
2. Right-click the groups' OU and choose Properties.
3. Click the Security tab.
4. Click the Advanced button.
5. In the Advanced Security Settings dialog box, click the Add button.

If the Add button is not visible, click the Edit button, and then click the Add button.

6. In the Select dialog box, enter the name for the group to whom you want to grant permission or click Browse to search for the group. When you are finished, click OK.

The Permission Entry dialog box appears.

7. Click the Properties tab.
8. In the Apply To drop-down list, choose This Object And All Descendant Objects.
9. In the Permissions list, select the Allow check boxes for the Read Members and Write Members permissions.

By default, all users have the Read Members permission, so that permission is not required. However, role-based access control is best implemented by assigning all the permissions required to achieve the desired capability rather than relying on permissions assigned indirectly.

Figure 4-11 shows the resulting Permission Entry dialog box.

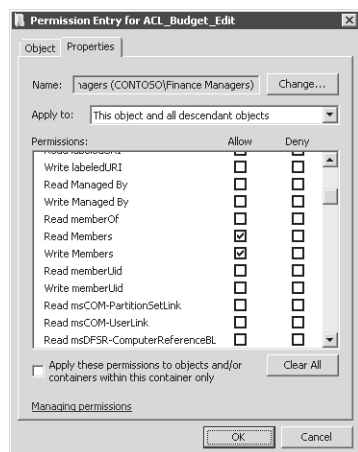


Figure 4-11 The Permission Entry dialog box showing the delegation of group membership management for a group

- Click OK to close each of the security dialog boxes.

Delegate the ability to manage membership for all groups in an OU

- In the Active Directory Users And Computers snap-in, click the View menu and make sure Advanced Features is selected.
- Right-click the groups' OU and choose Properties.
- Click the Security tab.
- Click the Advanced button.
- In the Advanced Security Settings dialog box, click the Add button.
If the Add button is not visible, click the Edit button, and then click the Add button.
- In the Select dialog box, enter the name for the group to whom you want to grant permission or click Browse to search for the group. When you are finished, click OK.
The Permission Entry dialog box appears.
- Click the Properties tab.
- In the Apply To drop-down list, choose Descendant Group Objects. If you are using earlier versions of the Active Directory Users And Computers snap-in, choose Group Objects.
- In the Permissions list, select the Allow check boxes for the Read Members and Write Members permissions.

By default, all users have the Read Members permission, so that permission is not required. However, role-based access control is best implemented by assigning all the permissions required to achieve the desired capability rather than relying on permissions assigned indirectly.

Figure 4-12 shows the resulting Permission Entry dialog box.

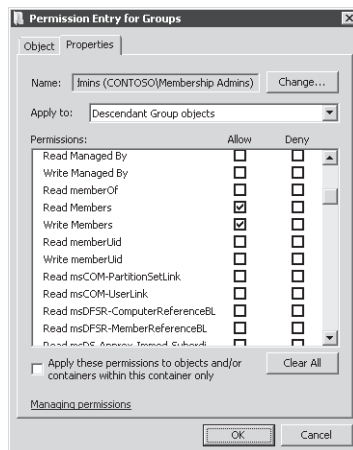


Figure 4-12 The Permission Entry dialog box showing the delegation of group membership management for all groups in the Groups OU

- Click OK to close each of the security dialog boxes.

Understanding Shadow Groups

Most management of an enterprise is implemented with groups. Groups are assigned permission to resources. Groups can be used to filter the scope of Group Policy objects. Groups are assigned fine-grained password policies. Groups can be used as collections for configuration management tools such as Microsoft System Center Configuration Manager. The list goes on. OUs, however, are not used as frequently to manage the enterprise, and in some cases, they cannot be used. For instance, OUs cannot be assigned permissions to resources, nor can they be assigned fine-grained password policies (discussed in Chapter 8, “Authentication”). Instead, the primary purpose of an OU is to provide a scope of management for the delegation of administrative permissions for the objects in that OU. In other words, an OU of users enables you to delegate to your help desk the ability to reset passwords for all users in the OU. OUs are administrative containers.

The reason for this separation of purpose between OUs and groups is that OUs do not provide the same flexibility as groups. A user or computer (or other object) can only exist within the context of a single OU whereas a security principal can belong to many groups. Therefore, groups are used for aligning identities with the capabilities required by those identities.

Sometimes, you might want to manage using an OU when it is not possible. For example, you might want to give all users in an OU access to a folder. Or you might want to assign a unique password policy to users in an OU. You cannot do so directly, but you can achieve your goal by creating what is called a *shadow group*. A shadow group is a group that contains the same users as an OU. More accurately, a shadow group contains users that meet a certain criterion.

The easiest way to create a shadow group is to create the group; then, in the OU containing the users, press Ctrl + A to select all users. Right-click any selected user and choose Add To Group. Type the name of the group and click OK.

Exam Tip On the 70-640 exam, be prepared to see the term *shadow group* in use. Know that it means a group that contains, as members, the users in an OU.

Unfortunately, Windows does not yet provide a way to maintain the membership of a shadow group dynamically. When you add or remove a user to or from the OU, you must also add or remove the user to or from the shadow group.

MORE INFO Maintaining shadow groups dynamically

See *Windows Administration Resource Kit: Productivity Solutions for IT Professionals* for scripts that will help maintain shadow groups dynamically.

Default Groups

A number of groups are created automatically on a server running Windows Server 2008. These are called *default local groups*, and they include well-known groups such as Administrators, Backup Operators, and Remote Desktop Users. Additional groups are created in a domain, in both the Builtin and Users containers, including Domain Admins, Enterprise Admins, and Schema Admins. The following list provides a summary of capabilities of the subset of default groups that have significant permissions and user rights related to the management of Active Directory:

- **Enterprise Admins (Users container of the forest root domain)** This group is a member of the Administrators group in every domain in the forest, giving it complete access to the configuration of all domain controllers. It also owns the Configuration partition of the directory and has full control of the domain naming context in all forest domains.
- **Schema Admins (Users container of the forest root domain)** This group owns and has full control of the Active Directory schema.
- **Administrators (Builtin container of each domain)** This group has complete control over all domain controllers and data in the domain naming context. It can change the membership of all other administrative groups in the domain, and the Administrators group in the forest root domain can change the membership of Enterprise Admins, Schema Admins, and Domain Admins. The Administrators group in the forest root domain is arguably the most powerful service administration group in the forest.
- **Domain Admins (Users container of each domain)** This group is added to the Administrators group of its domain. Therefore, it inherits all the capabilities of the Administrators group. It is also, by default, added to the local Administrators group of each domain member computer, giving Domain Admins ownership of all domain computers.
- **Server Operators (Builtin container of each domain)** This group can perform maintenance tasks on domain controllers. It has the right to log on locally, start and stop services, perform backup and restore operations, format disks, create or delete shares, and shut down domain controllers. By default, this group has no members.
- **Account Operators (Builtin container of each domain)** This group can create, modify, and delete accounts for users, groups, and computers located in any organizational unit in the domain (except the Domain Controllers OU) as well as in the Users and Computers container. Account Operators cannot modify accounts that are members of the Administrators or Domain Admins groups, nor can they modify those groups. Account Operators can also log on locally to domain controllers. By default, this group has no members.

- **Backup Operators (Builtin container of each domain)** This group can perform backup and restore operations on domain controllers as well as log on locally and shut down domain controllers. By default, this group has no members.
- **Print Operators (Builtin container of each domain)** This group can maintain print queues on domain controllers. It can also log on locally and shut down domain controllers.

The default groups that provide administrative privileges should be managed carefully because they typically have broader privileges than are necessary for most delegated environments and because they often apply protection to their members.

The Account Operators group is a perfect example. If you examine its capabilities in the preceding list, you will see that its rights are very broad, indeed. It can even log on locally to a domain controller. In very small enterprises, such rights will probably be appropriate for one or two individuals who might be domain administrators anyway. In enterprises of any size, the rights and permissions granted to Account Operators are usually far too broad.

Additionally, Account Operators is, like the other administrative groups listed previously, a *protected group*. Protected groups are defined by the operating system and cannot be unprotected. Members of a protected group become protected. The result of protection is that the permissions (ACLs) of members are modified so that they no longer inherit permissions from their OU but, rather, receive a copy of an ACL that is quite restrictive. For example, if Jeff Ford is added to the Account Operators group, his account becomes protected and the help desk, which can reset all other user passwords in the People OU, cannot reset Jeff Ford's password.

MORE INFO Protected accounts

For more information about protected accounts, see Knowledge Base article 817433 at <http://support.microsoft.com/?kbid=817433>. If you want to search the Internet for resources, use the keyword *adminSDHolder*.

For these reasons—overdelegation and protection—strive to avoid adding users to the groups listed previously that do not have members by default: Account Operators, Backup Operators, Server Operators, and Print Operators. Instead, create custom groups to which you assign permissions and user rights that achieve your business and administrative requirements. For example, if Scott Mitchell should be able to perform backup operations on a domain controller but should not be able to perform restore operations that could lead to database rollback or corruption and should not be able to shut down a domain controller, don't put Scott in the Backup Operators group. Instead, create a group and assign it only the Backup Files And Directories user right; then add Scott as a member.

MORE INFO Default group capabilities information

There is an exhaustive reference to the default groups in a domain and to the default local groups on Microsoft TechNet. If you are not familiar with the default groups and their capabilities, you should prepare for the examination by reading them. The default domain groups reference is at <http://technet2.microsoft.com/WindowsServer/en/library/1631acad-ef34-4f77-9c2e-94a62f8846cf1033.mspx>, and the default local groups reference is at <http://technet2.microsoft.com/WindowsServer/en/library/f6e01e51-14ea-48f4-97fc-5288a9a4a9b11033.mspx>.

Special Identities

Windows and Active Directory also support *special identities*, groups for which membership is controlled by the operating system. You cannot view the groups in any list in the Active Directory Users and Computers snap-in, for example. You cannot view or modify the membership of these special identities, and you cannot add them to other groups. You can, however, use these groups to assign rights and permissions. The most important special identities, often referred to as groups for convenience, are described in the following list:

- **Anonymous Logon** Represents connections to a computer and its resources that are made without supplying a user name and password. Prior to Microsoft Windows Server 2003, this group was a member of the Everyone group. Beginning in Windows Server 2003, this group is no longer a default member of the Everyone group.
- **Authenticated Users** Represents identities that have been authenticated. This group does not include Guest, even if the Guest account has a password.
- **Everyone** Includes Authenticated Users and Guest. On computers running versions of Windows earlier than Windows Server 2003, this group includes Anonymous Logon.
- **Interactive** Represents users accessing a resource while logged on locally to the computer hosting the resource, as opposed to accessing the resource over the network. When a user accesses any given resource on a computer to which the user is logged on locally, the user is automatically added to the Interactive group for that resource. Interactive also includes users logged on through a remote desktop connection.
- **Network** Represents users accessing a resource over the network, as opposed to users who are logged on locally at the computer hosting the resource. When a user accesses any given resource over the network, the user is automatically added to the Network group for that resource.

The importance of these special identities is that they enable you to provide access to resources based on the type of authentication or connection rather than on the user account. For example, you could create a folder on a system that allows users to view its contents when logged on locally to the system but does not allow the same users to view the contents from a

mapped drive over the network. This would be achieved by assigning permissions to the Interactive special identity.

Chapter 5

Computers

Computers in a domain are security principals, like users are. They have an account with a logon name and password that Microsoft Windows changes automatically every 30 days or so. They authenticate with the domain. They can belong to groups, have access to resources, and be configured by Group Policy. And, like users, computers sometimes lose track of their passwords, requiring a reset, or have accounts that need to be disabled or enabled.

Managing computers—both the objects in Active Directory Domain Services (AD DS) and the physical devices—is part of the day-to-day work of most IT professionals. New systems are added to the organization, computers are taken offline for repairs, computers are exchanged between users or roles, and older equipment is retired or upgraded, leading to the acquisition of replacement systems. Each of these activities requires managing the identity of the computer represented by its object, or account, and Active Directory.

Unfortunately, most enterprises do not invest the same kind of care and process in the creation and management of computer accounts as they do for user accounts, even though both are security principals. In this chapter, you will learn how to create computer objects, which include attributes required for the object to be an account. You will learn how to support computer accounts through their life cycle, including configuration, troubleshooting, repairing, and deprovisioning computer objects. You will also deepen your understanding of the process through which a computer joins a domain, so that you can identify and avoid potential points of failure.

Exam objectives in this chapter:

- Creating and Maintaining Active Directory Objects
 - Automate creation of Active Directory accounts.
 - Maintain Active Directory accounts.

Before You Begin

This chapter applies Microsoft Windows PowerShell, Microsoft VBScript, *CSVDE*, and *LDIFDE* to the task of automating computer account creation. Please read Lesson 1, “Automating the Creation of User Accounts,” and Lesson 2, “Creating Users with Windows PowerShell and VBScript,” of Chapter 3, “Users,” prior to reading this chapter.

Real World

Dan Holme

“Computers are people, too,” or at least in Active Directory they are. In fact, computers have the *objectClass* attribute of *user*. They have accounts, just as users do. They can even forget their passwords, like users do. Because computers are security principals and can be used to scope Group Policy (as you’ll learn in the next chapter), it is important to treat computer accounts with the same care as you’d treat user accounts.

I’m sure you’ve run into a situation when you had to remove a computer from a domain and then had it rejoin the domain. As you will see in Lesson 3, “Supporting Computer Objects and Accounts,” that’s a bad practice, equivalent to deleting and re-creating a user’s account just because the user forgot his or her password. That’s just one example of scenarios I see regularly, in which administrators are a bit less careful with computer accounts than they probably should be.

In this lesson, you’ll learn the best practices for supporting computer accounts with the same level of respect as other security principals (including users and groups) in the domain. You’ll also learn how to use command-line tools, VBScript, and Windows PowerShell scripts to automate the creation and management of computer objects. You’ll see a lot of similarities to the procedures discussed in Chapter 3. Why? Because computers are people, too!

Lesson 1: Creating Computers and Joining the Domain

The default configuration of Windows Server 2008—as well as of Microsoft Windows Server 2003, Windows Vista, Windows XP, and Windows 2000—is that the computer belongs to a workgroup. Before you can log on to a computer with a domain account, that computer must belong to the domain. To join the domain, the computer must have an account in the domain which, like a user account, includes a logon name (*sAMAccountName*), a password, and a security identifier (SID) that uniquely represent the computer as a security principal in the domain. Those credentials enable the computer to authenticate against the domain and to create a secure relationship that then enables users to log on to the system with domain accounts. In this lesson, you will learn the steps to prepare the domain for a new computer account, and you will explore the process through which a computer joins the domain.

After this lesson, you will be able to:

- Design an OU structure for computers.
- Create computer objects in the domain.
- Delegate the creation of computer objects.
- Join computers to the domain.
- Redirect the default computer container.
- Prevent nonadministrative users from creating computers and joining the domain.

Estimated lesson time: 45 minutes

Understanding Workgroups, Domains, and Trusts

In a workgroup, each system maintains an identity store of user and group accounts against which users can be authenticated and access can begin. The local identity store on each computer is called the Security Accounts Manager (SAM) database. If a user logs on to a workgroup computer, the system authenticates the user against its local SAM database. If a user connects to another system, to access a file for example, the user is re-authenticated against the identity store of the remote system. From a security perspective, a workgroup computer is, for all intents and purposes, a standalone system.

When a computer joins a domain, it delegates the task of authenticating users to the domain. Although the computer continues to maintain its SAM database to support local user and group accounts, user accounts will typically be created in the central domain directory. When a user logs on to the computer with a domain account, the user is now authenticated by a domain controller rather than by the SAM. Said another way, the computer now *trusts* another authority to validate a user's identity. Trust relationships are generally discussed in the context of two domains, as you will learn in Chapter 12, "Domains and Forests," but there is also a trust between each domain member computer and its domain that is established when the computer joins the domain.

Identifying Requirements for Joining a Computer to the Domain

Three things are required for you to join a computer to an Active Directory domain:

- A computer object must be created in the directory service.
- You must have appropriate permissions to the computer object. The permissions allow you to join a computer with the same name as the object to the domain.
- You must be a member of the local Administrators group on the computer to change its domain or workgroup membership.

The next sections examine each of these requirements.

Computers Container

Before you create a computer object in the directory service—the first of the three requirements for joining a computer to the domain—you must have a place to put it. When you create a domain, the Computers container is created by default (CN=Computers, . . .). This container is not an organizational unit (OU); it is an object of class *container*. There are subtle but important differences between a container and an OU. You cannot create an OU within a container, so you cannot subdivide the Computers OU, and you cannot link a Group Policy object to a container. Therefore, it is highly recommended to create custom OUs to host computer objects instead of using the Computers container.

Creating OUs for Computers

Most organizations create at least two OUs for computer objects: one to host computer accounts for clients—desktops, laptops, and other user systems—and another for servers. These two OUs are in addition to the Domain Controllers OU created by default during the installation of Active Directory. In each of these OUs, computer objects are created. There is no technical difference between a computer object in a clients OU and a computer object in a servers or domain controllers OU; computer objects are computer objects. But separate OUs are typically created to provide unique scopes of management so that you can delegate management of client objects to one team and server objects to another.

Your administrative model might necessitate further dividing your client and server OUs. Many organizations create sub-OUs beneath a server OU to collect and manage specific types of servers, for example, an OU for file and print servers and an OU for database servers. By doing so, the team of administrators for each type of server can be delegated permissions to manage computer objects in the appropriate OU. Similarly, geographically distributed organizations with local desktop support teams often divide a parent OU for clients into sub-OUs for each site. This approach enables each site's support team to create computer objects in the site for client computers and join computers to the domain by using those computer objects. This is an example

only, what is most important is that your OU structure reflects your administrative model so that your OUs provide single points of management for the delegation of administration.

Figure 5-1 illustrates a typical OU design for an organization whose server administration teams are focused on specific types of servers and whose desktop support teams are focused on clients in specific geographical areas.

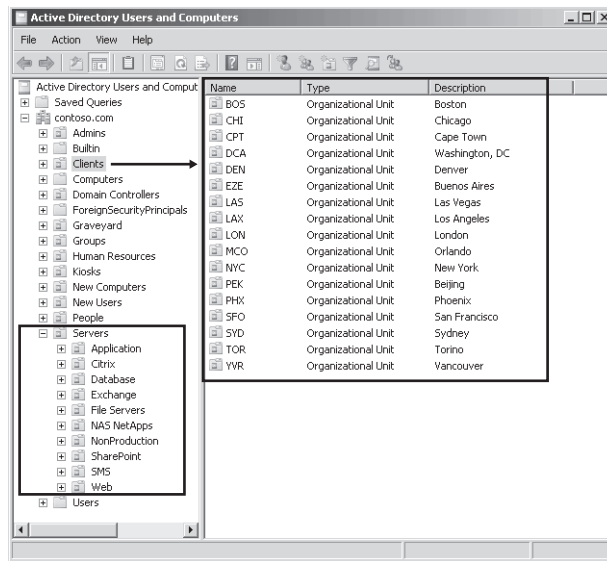


Figure 5-1 A common OU design illustrating site-based administration of clients and role-based administration of servers

Additionally, separate OUs enable you to create different baseline configurations, using different Group Policy objects (GPOs) linked to the client and the server OUs. Group Policy, discussed in detail in Chapter 6, “Group Policy Infrastructure,” enables you to specify configuration for collections of computers by linking GPOs that contain configuration instructions to OUs. It is common for organizations to separate clients into desktop and laptop OUs. GPOs specifying desktop or laptop configuration can then be linked to appropriate OUs.

If your organization has decentralized, site-based administration and wants to manage unique configurations for desktops and laptops, you face a design dilemma. Should you divide your clients OU based on administration and then subdivide desktops and laptops, or should you divide your clients OU into desktop and laptop OUs and then subdivide based on administration? The options are illustrated in Figure 5-2. Because the primary design driver for Active Directory OUs is the efficient delegation of administration through the inheritance of access control lists (ACLs) on OUs, the design on the left would be recommended.

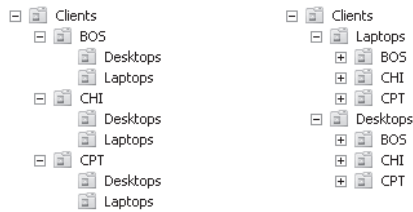


Figure 5-2 OU design options

Delegating Permission to Create Computers

By default, the Enterprise Admins, Domain Admins, Administrators, and Account Operators groups have permission to create computer objects in any new OU. However, as discussed in Chapter 4, “Groups,” it is recommended that you tightly restrict membership in the first three groups and that you do not add administrators to the Account Operators group.

Instead, delegate the permission to create computer objects to appropriate administrators or support personnel. The permission required to create a computer object is Create Computer Objects. This permission, assigned to a group for an OU, allows members of the group to create computer objects in that OU. For example, you might allow your desktop support team to create computer objects in the clients OU and allow your file server administrators to create computer objects in the file servers OU.

Practice It Exercise 3, “Delegate the Ability to Create Computer Objects,” at the end of this lesson, steps you through the procedure required to delegate the creation of computer objects.

Prestaging a Computer Account

After you have been given permission to create computer objects, you can do so by right-clicking the OU and choosing Computer from the New menu. The New Object – Computer dialog box, shown in Figure 5-3, appears.

Enter the computer name, following the naming convention of your enterprise, and select the user or group that will be allowed to join the computer to the domain with this account. The two computer names—Computer Name and Computer Name (Pre-Windows 2000)—should be the same; there is rarely, if ever, a justification for configuring them separately.

NOTE The New Object – Computer Wizard over-delegates

The permissions that are applied to the user or group you select in the New Object – Computer Wizard are more than are necessary simply to join a computer to the domain. The selected user or group is also given the ability to modify the computer object in other ways. For guidance regarding a least-privilege approach to delegating permission to join a computer to the domain, see *Windows Administration Resource Kit: Productivity Solutions for IT Professionals* by Dan Holme (Microsoft Press, 2008).

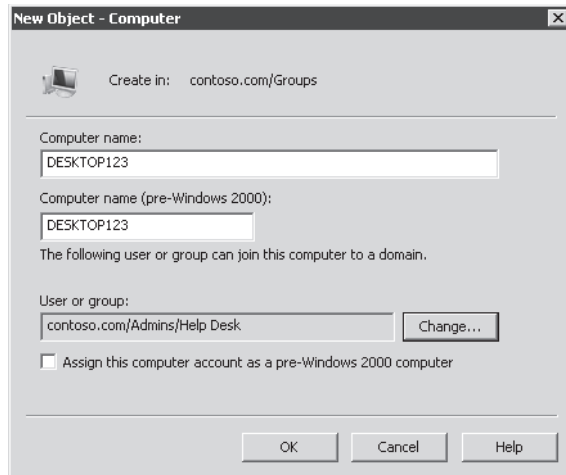


Figure 5-3 The New Object – Computer dialog box

The process you have completed to create a computer account before joining the computer to the domain is called *prestaging* the account. The advantage of performing this procedure is that the account is in the correct OU and is, therefore, delegated according to the security policy defined by the ACL of the OU and is within the scope of GPOs linked to the OU before the computer joins the domain. Prestaging is highly recommended for reasons discussed in the “Importance of Prestaging Computer Objects” section.

Joining a Computer to the Domain

By prestaging the computer object, you fulfill the first two requirements for joining a computer to a domain: the computer object exists, and you have specified who has permissions to join a computer with the same name to the domain. Now, a local administrator of the computer can change the computer’s domain membership and enter the specified domain credentials to complete the process successfully. To join a computer to the domain, follow these steps:

1. Log on to the computer with credentials that belong to the local Administrators group on the computer.
Only local administrators can alter the domain or workgroup membership of a computer.
2. Open the System properties, using one of the following methods:
 - ❑ Windows XP, Windows Server 2003: Right-click My Computer and choose Properties.
 - ❑ Windows Vista, Windows Server 2008: Right-click Computer; choose Properties; and then, in the Computer Name, Domain, And Workgroup Settings section, click Change Settings. Click if prompted.
3. Click the Computer Name tab.

4. Click Change.
5. Under Member Of, select Domain.
6. Type the name of the domain you want to join.

NOTE Use the full DNS name of the domain

Use the full DNS name of the Active Directory domain. Not only is this more accurate and more likely to succeed, but if it does not succeed, it indicates a possible problem with DNS name resolution that should be rectified before joining the computer to the domain.

7. Click OK.
8. Windows prompts for the credentials of your user account in the domain.
The domain checks to see whether a computer object already exists with the name of the computer. One of the following three things happens:

- ❑ If the object exists and a computer with that name has already joined the domain, an error message is returned, and you cannot join the computer to the domain.
- ❑ If the object exists and it is prestaged—a computer with the same name has not joined the domain—the domain confirms that the domain credentials you entered have permission to join the domain using that account. These permissions are discussed in the “Prestaging a Computer Account” section.
- ❑ If the computer account is not prestaged, Windows checks to see whether you have permissions to create a new computer object in the default computer container. If you do have permissions to create a new computer object in the default computer container, the object is created with the name of the computer. This method of joining a domain is supported for backward compatibility, but is not recommended. It is recommended to prestage the account as indicated earlier and as detailed in the next section, “The Importance of Prestaging Computer Objects.”

The computer then joins the domain by assuming the identity of its Active Directory object. It configures its SID to match the domain computer account’s SID and sets an initial password with the domain. The computer then performs other tasks related to joining the domain. It adds the Domain Admins group to the local Administrators group and the Domain Users group to the local Users group.

9. You are prompted to restart the computer. Click OK to close this message box.
10. Click Close (in Windows Vista) or OK (in Windows XP) to close the System Properties dialog box.
11. You are prompted, again, to restart the computer, after which the system is fully a member of the domain, and you can log on using domain credentials.

The *Netdom.exe* command enables you to join a computer to the domain from the command line. The basic syntax of the command is:

```
netdom join MachineName /Domain:DomainName [/OU:"DN of OU"]  
  [/UserO:LocalUsername] [/PasswordO:{LocalPassword*} ]  
  [/UserD:DomainUsername] [/PasswordD:{DomainPassword*} ]  
  [/SecurePasswordPrompt] [/REBoot[:TimeInSeconds]]
```

It can be useful to join a computer to a domain from the command line, first, because it can be included in a script that performs other actions. Second, *Netdom.exe* can be used to join a computer *remotely* to the domain. Third, *Netdom.exe* enables you to specify the OU for the computer object. The command's parameters are, for the most part, self-explanatory. *UserO* and *PasswordO* are credentials that are members of the workgroup computer's local Administrators group. Specifying * for the password causes *Netdom.exe* to prompt for the password on the command line. *UserD* and *PasswordD* are domain credentials with permission to create a computer object, if the account is not prestaged, or to join a computer to a prestaged account. The *REBoot* parameter causes the system to reboot after joining the domain. The default timeout is 30 seconds. The *SecurePasswordPrompt* parameter displays a pop-up for credentials when * is specified for either *PasswordO* or *PasswordD*.

Importance of Prestaging Computer Objects

The best practice is to prestage a computer account prior to joining the computer to the domain. Unfortunately, Windows enables you to join a computer to a domain without following best practices. You can log on to a workgroup computer as a local administrator and change the computer's membership to the domain. Then, on demand, Windows creates a computer object in the default computer container, gives you permission to join a computer to that object, and joins the system to the domain.

There are three problems with this behavior of Windows. First, the computer account created automatically by Windows is placed in the default computer container, which is not where the computer object belongs in most enterprises. Second, you must move the computer from the default computer container into the correct OU, which is an extra step that is often forgotten. Third, any user can join a computer to the domain—no domain-level administrative permissions are required. Because a computer object is a security principal, and because the creator of a computer object owns the object and can change its attributes, this exposes a potential security vulnerability. The next sections detail these disadvantages.

Configuring the Default Computer Container

When you join a computer to the domain and the computer object does not already exist in Active Directory, Windows automatically creates a computer account in the default computer container, which is called Computers (CN=Computers,DC=*domain*, by default). The problem with this relates to the discussion of OU design earlier in the lesson. If you have implemented the best practices described there, you have delegated permissions to administer computer objects in specific OUs for clients and servers. Additionally, you might have linked GPOs to

those OUs to manage the configuration of these computer objects. If a new computer object is created outside of those OUs, in the default computer container, the permissions and configuration it inherits from its parent container will be different than what it should have received. You will then need to remember to move the computer from the default container to the correct OU after joining the domain.

Two steps are recommended to reduce the likelihood of this problem. First, always try to pre-stage computer accounts. If an account is prestaged for a computer in the correct OU, then when the computer joins the domain, it will use the existing account and will be subject to the correct delegation and configuration.

Second, to reduce the impact of systems being joined to the domain without a prestaged account, change the default computer container so that it is not the Computers container itself but, instead, is an OU that is subject to appropriate delegation and configuration. For example, if you have an OU called Clients, you can instruct Windows to use that OU as the default computer container, so that if computers are joined to the domain without prestaged accounts, the objects are created in the Clients OU.

The *Redircmp.exe* command, available on domain controllers, redirects the default computer container with the following syntax:

```
redircmp "DN of OU for new computer objects"
```

Now, if a computer joins the domain without a prestaged computer account, Windows creates the computer object in the specified organizational unit.

Redirecting the Default User Container

The same concepts apply to the creation of user accounts. By default, if a user account is created using an earlier practice that does not specify the OU for the account, the object is created in the default user container (CN=Users,DC=domain, by default). The *Redirusr.exe* command, available on domain controllers, can redirect the default container to an actual OU that is delegated and configured appropriately. *Redirusr*, like *Redircmp*, takes a single parameter: the distinguished name of the OU that will become the default user container.

Exam Tip The *Redircmp.exe* command redirects the default computer container to a specified OU. *Redirusr.exe* does the same for the default user container. You might see these two commands used as *distracters*—presented as potential (but incorrect) answers to questions that have nothing to do with the default computer or user containers. As you look at any exam question, evaluate the possible answers to determine whether the answers are proposing to use real commands but in the wrong application of those commands.

Restricting the Ability of Users to Create Computers

When a computer account is prestaged, the permissions on the account determine who is allowed to join that computer to the domain. When an account is not prestaged, Windows will, by default, allow any authenticated user to create a computer object in the default computer container. In fact, Windows will allow any authenticated user to create up to ten computer objects in the default computer container. The creator of a computer object, by default, has permission to join that computer to the domain. It is through this mechanism that any authenticated user can join ten computers to the domain without any explicit permissions to do so.

The ten-computer quota is configured by the *ms-DS-MachineAccountQuota* attribute of the domain. It allows any authenticated user to join a computer to the domain, no questions asked. This is problematic from a security perspective because computers are security principals, and the creator of a security principal has permission to manage that computer's properties. In a way, the quota is like allowing any domain user to create ten user accounts, without any controls.

It is highly recommended that you close this loophole so that nonadministrative users cannot join computers to the domain. To change the *ms-DS-MachineAccountQuota* attribute, follow these steps:

1. Open ADSI Edit from the Administrative Tools folder.
2. Right-click ADSI Edit and choose Connect To.
3. In the Connection Point section, choose Select A Well Known Naming Context and, from the drop-down list, choose Default Naming Context.
4. Click OK.
5. Expand Default Naming Context.
6. Right-click the dc=contoso,dc=com domain folder, for example, and choose Properties.
7. Select ms-DS-MachineAccountQuota and click Edit.
8. Type 0.
9. Click OK.

The Authenticated Users group also is assigned the user right to add workstations to the domain, but you do not have to modify this right if you have changed the default value of the *ms-DS-MachineAccountQuota* attribute.

After you have changed the *ms-DS-MachineAccountQuota* attribute to zero, you can be assured that the only users who can join computers to the domain are those who have been specifically delegated permission to join prestaged computer objects or to create new computer objects.

Quick Check

- What two things determine whether you can join a computer account to the domain?

Quick Check Answer

- To join a computer to a prestaged account, you must be given permission on the account to join it to the domain. If the account is not prestaged, the *ms-DS-MachineAccountQuota* attribute will determine the number of computers you can join to the domain in the default computer container without explicit permission.

After you've eliminated this loophole, you must make sure you have given appropriate administrators explicit permission to create computer objects in the correct OUs, as described in the "Delegating Permission to Create Computers" section; otherwise, the error message shown in Figure 5-4 will appear.

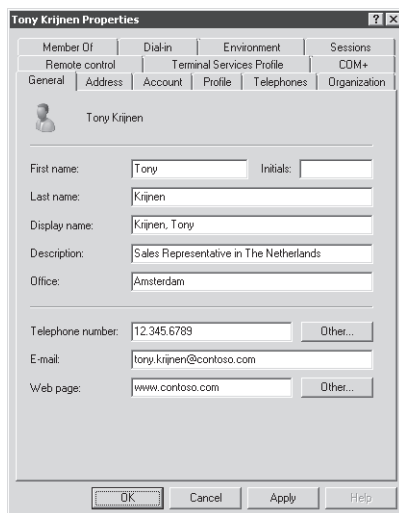


Figure 5-4 An error message appearing when a user has exceeded the default computer account quota specified by the *ms-DS-MachineAccountQuota* attribute

Lesson 2: Automating the Creation of Computer Objects

The steps you learned in Lesson 1 for creating a computer account become burdensome if you must create dozens or even hundreds of computer accounts at the same time. Commands such as *CSVDE*, *LDIFDE*, and *Dsadd*, as well as VBScript and Windows PowerShell scripts, can import and automate the creation of computer objects. Scripts can also enable you to *provision* computer objects, that is, to perform business logic such as the enforcement of computer naming conventions. In this lesson, you will learn to import, automate, and provision computer objects. You will build upon the knowledge of these commands that you gained from reading Lesson 1 and Lesson 2 of Chapter 3, which are a prerequisite for this lesson.

After this lesson, you will be able to:

- Use *CSVDE* and *LDIFDE* to import computers.
- Create computers with *Dsadd*.
- Create computers with *Netdom*.
- Create computers with Windows PowerShell.
- Create computers with VBScript.

Estimated lesson time: 30 minutes

Importing Computers with *CSVDE*

You were introduced to the *Comma-Separated Values Data Exchange* (*CSVDE*) command in Lesson 1 of Chapter 3. *CSVDE* is a command-line tool that imports or exports Active Directory objects from or to a comma-delimited text file (also known as a comma-separated value text file, or .csv file). The basic syntax of the *CSVDE* command is:

```
csvde [-i] [-f "Filename"] [-k]
```

The *-i* parameter specifies import mode; without it, the default mode of *CSVDE* is export. The *-f* parameter identifies the file name to import from or export to. The *-k* parameter is useful during import operations because it instructs *CSVDE* to ignore errors including Object Already Exists, Constraint Violation, and Attribute Or Value Already Exists.

Comma-delimited files can be created, modified, and opened with tools as familiar as Notepad and Microsoft Office Excel. The first line of the file defines the attributes by their Lightweight Directory Access Protocol (LDAP) attribute names. Each object follows, one per line, and must contain exactly the attributes listed on the first line. A sample file is shown in Excel in Figure 5-5.

When importing computers, be sure to include the *userAccountControl* attribute and set it to 4096. This attribute ensures that the computer will be able to join the account. Also include

the pre-Windows 2000 logon name of the computer, the *sAMAccountName* attribute, which is the name of the computer followed by a dollar sign (\$) as shown in Figure 5-5.

	A	B	C	D	E
1	DN	objectClass	name	userAccountControl	sAMAccountName
2	CN=DESKTOP103,OU=Clients,DC=contoso,DC=com	computer	DESKTOP103	4096	DESKTOP103\$
3	CN=DESKTOP104,OU=Clients,DC=contoso,DC=com	computer	DESKTOP104	4096	DESKTOP104\$
4	CN=SERVER02,OU=Servers,DC=contoso,DC=com	computer	SERVER02	4096	SERVER02\$

Figure 5-5 A .csv file, opened in Excel, that will create three computer accounts

MORE INFO In Chapter 3 and Chapter 4, you used the *CSVDE* command to import users and groups. For more information about *CSVDE*, including details regarding its parameters and usage to export directory objects, type *csvde /?* or search the Windows Server 2008 Help and Support Center.

Importing Computers with *LDIFDE*

Chapter 3 also introduced you to *Ldifde.exe*, which imports data from files in the Lightweight Directory Access Protocol Data Interchange Format (LDIF) format. LDIF files are text files within which operations are specified by a block of lines separated by a blank line. Each operation begins with the *DN* attribute of the object that is the target of the operation. The next line, *changeType*, specifies the type of operation: *add*, *modify*, or *delete*.

The following listing is an LDIF file that will create two server accounts:

```
dn: CN=SERVER10,OU=Servers,DC=contoso,DC=com
changetype: add
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
objectClass: computer
cn: SERVER10
userAccountControl: 4096
sAMAccountName: SERVER10$
```

```
dn: CN= SERVER11,OU= Servers,DC=contoso,DC=com
changetype: add
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
objectClass: computer
cn: SERVER11
userAccountControl: 4096
sAMAccountName: SERVER11$
```

The basic syntax of the *LDIFDE* command is similar to that of the *CSVDE* command:

```
ldifde [-i] [-f "Filename"] [-k]
```

By default, *LDIFDE* is in export mode. The *-i* parameter specifies import mode. You must specify the *-f* mode to identify the file you are using for import or export. *LDIFDE* will stop when it encounters errors unless you specify the *-k* parameter, in which case, *LDIFDE* continues processing.

Exam Tip Remember that the default mode of *CSVDE* and *LDIFDE* is export. You must use the *-i* parameter to import objects.

Creating Computers with *Dsadd*

The *Dsadd* command has been used in previous chapters to create objects in Active Directory. To create computer objects, simply type **dsadd computer *ComputerDN*** where *ComputerDN* is the distinguished name (DN) of the computer, such as CN=Desktop123,OU=Desktops,DC=contoso,DC=com.

If the computer's DN includes a space, surround the entire DN with quotation marks. The *ComputerDN* parameter can include more than one distinguished name for new computer objects, making *Dsadd Computer* a handy way to generate multiple objects at once. The parameter can be entered in one of the following ways:

- By piping a list of DNs from another command such as *Dsquery*.
- By typing each DN on the command line, separated by spaces.
- By leaving the *DN* parameter empty, at which point, you can type the DNs, one at a time, at the keyboard console of the command prompt. Press Enter after each DN. Press Ctrl+Z and Enter after the last DN.

The *Dsadd Computer* command can take the following optional parameters after the DN parameter:

- *-samid SAMName*
- *-desc Description*
- *-loc Location*

Creating Computers with *Netdom*

The *Netdom* command is also able to perform a variety of domain account and security tasks from the command line. In Lesson 1, you learned to use *Netdom* to join a computer to the domain. You can also use it to create a computer account by typing the following command:

```
netdom add ComputerName /domain:DomainName [/ou:OU DN]  
[/userd:User /passwordd:Password]
```

This command creates the computer account for *ComputerName* in the domain indicated by the *domain* parameter, using the credentials specified by *UserD* and *PasswordD*. The *ou* parameter causes the object to be created in the OU specified by the *OU DN* distinguished name following

the parameter. If no *OU* is supplied, the computer account is created in the default computer container. The user credentials must have permissions to create computer objects.

Creating Computers with Windows PowerShell

Chapter 3 introduced you to Windows PowerShell, the new administrative and automation shell for Windows platforms. You learned how to create users in that chapter. As with user objects, you can create computer objects by following these high-level steps:

1. Connect to the container (the OU) in which you want to create a computer.
2. Use the *Create* method of the container to create the computer.
3. Populate mandatory attributes.
4. Commit your changes.

To connect to an OU in Windows PowerShell, type the following at a PowerShell prompt:

```
$objOU = [ADSI]"LDAP://DN of OU"
```

The command creates an object reference stored in the *\$objOU* variable that represents the OU. You can now invoke the methods of the OU, using the *\$objOU* variable. To create a computer, use the *Create* method by typing the following:

```
$objComputer = $objOU.Create("computer", "CN=Computer CN")
```

Next, you must configure two attributes. The first is the computer's pre-Windows 2000 logon name, the *sAMAccountName* attribute, which is the computer's name appended with a dollar sign (\$). The second is the computer's *userAccountControl* attribute, which must be set to 4096 (0x1000 in hexadecimal). The *userAccountControl* attribute is a series of flags, 1 bit each. This bit indicates that the account is for a domain member. Without it, a computer will not be able to join the domain by using the account. To set these two attributes, type the following:

```
$objComputer.Put("sAMAccountName", "ComputerName$")  
$objComputer.Put("userAccountControl", 4096)
```

You can set other attributes at this time as well. For example, you can set *description* or *info*. When you have finished configuring attributes, you must commit your changes with the following code.

```
$objComputer.SetInfo()
```

Importing Computers from a Database with Windows PowerShell

On the 70-640 examination, it is highly unlikely that you will need to know how to connect to a database and create computers with Windows PowerShell. However, such knowledge can be a tremendous benefit in your production environment. Assume you receive a list of computers that are being shipped from your vendor. You want to prestage computer accounts for those systems.

You can easily do so with a Windows PowerShell script. With scripts, you can also perform business logic such as enforcing naming standards. In this section, you will learn how to do so.

Windows PowerShell can connect to and expose a data source such as a .csv file, which you can create in Excel. So, for example, you could paste the asset tags from the list of computers you received from your vendor into an Excel worksheet, as shown in Figure 5-6, and save the worksheet as a .csv file with a name such as Assets.csv.

	A	B
1	AssetTag	Type
2	A849XD	Desktop
3	D82KE8	Desktop
4	ELW938	Laptop
5	XKD8G0	Laptop
6	93JX9D	Laptop
7	SJ0GJ3	Laptop

Figure 5-6 A simple Excel data source of computer asset tags

Assume that you want to import these computers into your domain, and you want them to follow two rules. First, laptops and desktops are in separate OUs, specifically the Laptops OU and the Desktops OU, under your Clients OU. Second, your computer naming convention is to prefix the asset tag with an L or a D, for laptop or desktop, respectively. For example, the computer name for the first computer listed in Figure 5-6 would be DA849XD. These two simple rules are examples of what would be called *logic* in the context of programming.

A script that would import computers from the file would look similar to the following code. Line numbers have been added to facilitate discussing the code.

```

1. $dataSource=import-csv "Assets.csv"
2. foreach($dataRecord in $dataSource) {
3.     #map variables to data source
4.     $AssetTag = $dataRecord.AssetTag
5.     $Type = $dataRecord.Type

6.     #determine name
7.     $ComputerName = $Type.substring(0,1) + $AssetTag
8.     $SAMAccountName=$ComputerName + "$"
9.     #determine OU
10.    $strOUADsPath = "LDAP://OU=" + $Type + "s" + `
11.        ",OU=Clients,DC=contoso,DC=com"

12.    #create the computer object
13.    $objOU=[ADSI]$strOUADsPath
14.    $objComputer=$objOU.Create("computer", "CN="+$ComputerName)
15.    $objComputer.Put("sAMAccountName", $SAMAccountName)
16.    $objComputer.Put("userAccountControl", 4096)
17.    $objComputer.SetInfo()
18. }

```

Lines 13–17 are identical to the commands shown in the previous section, except that in line 13, a variable is used rather than a hard-coded path to the OU. These lines are part of a block of code, bounded by lines 2 and 18, that are repeated for each record in the data source. The data source is defined in line 1, using the same `Import-Csv` cmdlet you learned about in Chapter 3. Line 2 uses a *foreach* collection (*foreach* is an alias for `ForEach-Object`) to loop through each record in the data source.

Lines 4 and 5 assign the two fields in each record to variables. Lines 6–11 perform the business logic. Line 7 creates the computer name, using the first character of the *Type* field (a *D* or an *L*) and appending *AssetTag*. Line 8 creates the *sAMAccountName* attribute by adding a dollar sign to the computer name. Line 10 creates the path to the OU for the object. The back tick mark at the end of line 10 is a line continuation character; it means that the code continues on line 11. Therefore, lines 10 and 11 are actually a single line of code. The logic indicated that a Desktop asset *Type* goes into the Desktops OU, so the type has an *s* added to it.

As you can see from this script, it is possible and not terribly difficult to create a data-driven provisioning system for new computer objects. Define your data sources and define your business logic, and Windows PowerShell scripts can do the rest.

Creating Computers with VBScript

VBScript uses the same Active Directory Services Interface (ADSI) as does Windows PowerShell to manipulate Active Directory objects, so the steps to create a computer are identical: connect to the container, create the object, populate its attributes, and commit the changes. The following code will create a computer in the domain:

```
Set objOU = GetObject("LDAP://DN of OU")
Set objComputer = objOU.Create("computer","CN=Computer CN")
objComputer.Put "sAMAccountName", "ComputerName$"
objComputer.Put "userAccountControl", 4096
objComputer.SetInfo
```

The code is very similar to the Windows PowerShell commands in lines 13–17 of the script presented in the previous section.

NOTE VBScript does databases

In the previous section, you learned how to use a `.csv` file as a data source for a Windows PowerShell script. VBScript can also load and use data from `.csv` files, but it is not as elegant as the Windows PowerShell `Import-Csv` cmdlet.

Lesson 3: Supporting Computer Objects and Accounts

A computer account begins its life cycle when it is created and when the computer joins the domain. Day-to-day administrative tasks include configuring computer properties; moving the computer between OUs; managing the computer itself; renaming, resetting, disabling, enabling, and, eventually, deleting the computer object. This lesson looks closely at the computer properties and procedures involved with these tasks and will equip you to administer computers in a domain.

After this lesson, you will be able to:

- Configure the properties of a computer running Active Directory.
- Move a computer between OUs.
- Rename a computer.
- Disable and enable computer accounts.
- Reset the secure channel of a domain member computer.
- Perform administrative tasks with the Active Directory Users and Computers snap-in, command-line commands, VBScript, and Windows PowerShell.

Estimated lesson time: 45 minutes

Configuring Computer Properties

When you create a computer object, you are prompted to configure only the most fundamental attributes, including the computer name and the delegation to join the computer to the domain. Computers have several properties that are not visible when creating the computer object, and you should configure these properties as part of the process of staging the computer account.

Open a computer object's Properties dialog box to set its location and description, configure its group memberships and dial-in permissions, and link it to the user object of the user to whom the computer is assigned. The Operating System tab is read-only. The information will be blank until a computer has joined the domain, using that account, at which time, the client publishes the information to its account.

Several object classes in Active Directory support the *managedBy* attribute that is shown on the Managed By tab. This linked attribute creates a cross-reference to a user object. All other properties—the addresses and telephone numbers—are displayed directly from the user object. They are not stored as part of the computer object itself.

On the Member Of tab of a computer's Properties dialog box, you can add the computer to groups. The ability to manage computers in groups is an important and often underused feature of Active Directory. A group to which computers belong can be used to assign resource access permissions to the computer or to filter the application of a GPO.

As with users and groups, it is possible to multiselect more than one computer object and subsequently manage or modify properties of all selected computers simultaneously.

Configuring Computer Attributes with *Dsmod*

The *Dsmod* command, which you learned about in Chapter 3 and Chapter 4, is able to modify only the *description* and the *location* attributes. It uses the following syntax:

```
dsmod computer "DN of Computer" [-desc Description] [-loc Location]
```

Configuring Computer Attributes with Windows PowerShell or VBScript

In Windows PowerShell and VBScript, you can change attributes of a computer with three steps:

1. Connect to the computer using ADSI and the *aDSPath* attribute of the computer in the form “LDAP://Distinguished Name of Computer.”
2. Use the *Put* method of the computer object to set single-valued attributes.
3. Use the *SetInfo* method to commit changes to the object.

The Windows PowerShell commands are as follows:

```
$objComputer = [ADSI]"LDAP://DN of Computer"  
$objComputer.Put ("property", value)  
$objComputer.SetInfo()
```

The VBScript code follows this format:

```
Set objComputer = GetObject("LDAP://DN of Computer")  
objComputer.Put "property",  
value objComputer.SetInfo
```

In both cases, if the value is a text value, it must be surrounded by quotes.

Moving a Computer

Many organizations have multiple OUs for computer objects. Some domains, for example, have computer OUs based on geographic sites, as shown in Figure 5-2. If you have more than one OU for computers, it is likely that someday you will need to move a computer between OUs.

You can move a computer in the Active Directory Users and Computers snap-in using either drag and drop or the *Move* command, available when you right-click a computer.

You must have appropriate permissions to move an object in Active Directory. Default permissions allow Account Operators to move computer objects between containers, including the Computers container and any OUs *except* into or out of the Domain Controllers OU. Administrators, which include Domain Admins and Enterprise Admins, can move computer objects

between any containers, including the Computers container, the Domain Controllers OU, and any other OUs. There is no way to delegate the specific task of moving an object in Active Directory. Instead, your ability to move a computer is derived from your ability to delete an object in the source container and create an object in the destination container. When you move the object, you are not actually deleting and re-creating it; those are just the permissions that are evaluated to allow you to perform a move.

The *Dsmove* command allows you to move a computer object or any other object. The syntax of *Dsmove* is:

```
dsmove ObjectDN [-newname NewName] [-newparent ParentDN]
```

The *newname* parameter enables you to rename an object. The *newparent* parameter enables you to move an object. To move a computer named DESKTOP153 from the Computers container to the Clients OU, you would type the following:

```
dsmove "CN=DESKTOP153,CN=Computers,DC=contoso,DC=com" -newparent  
"OU=Clients,DC=contoso,DC=com"
```

To move a computer in Windows PowerShell, you must use the *psbase.MoveTo* method. The following two lines of code will move a computer:

```
$objUser=[ADSI]"LDAP://ComputerDN "  
$objUser.psbase.MoveTo("LDAP://TargetOUDN")
```

With VBScript, you connect to the source container and use the container's *MoveHere* method:

```
Set objOU = GetObject("LDAP://TargetOUDN")  
objOU.MoveHere "LDAP://ComputerDN", vbNullString
```

Before you move a computer, consider the implications to delegation and configuration. The target OU might have different permissions than the originating OU, in which case, the object will inherit new permissions affecting who is able to manage the object further. The target OU might also be within the scope of different GPOs, which would change the configuration of settings on the system itself.

Managing a Computer from the Active Directory Users and Computers Snap-In

One of the beneficial but lesser used features of the Active Directory Users and Computers snap-in is the *Manage* command. Select a computer in the Active Directory Users and Computers snap-in, right-click it, and choose Manage. The Computer Management console opens, focused on the selected computer, giving you instant access to the computer's event logs, local users and groups, shared folder configuration, and other management extensions. The tool is launched with the credentials used to run the Active Directory Users and Computers snap-in, so you must be running the Active Directory Users and Computers snap-in as a member of the

remote computer's Administrators group to gain the maximum functionality from the Computer Management console.

Understanding the Computer's Logon and Secure Channel

Every member computer in an Active Directory domain maintains a computer account with a user name (*sAMAccountName*) and password, just like a user account does. The computer stores its password in the form of a local security authority (LSA) secret and changes its password with the domain every 30 days or so. The Netlogon service uses the credentials to log on to the domain, which establishes the secure channel with a domain controller.

Recognizing Computer Account Problems

Computer accounts and the secure relationships between computers and their domain are robust. However, certain scenarios might arise in which a computer is no longer able to authenticate with the domain. Examples of such scenarios include:

- After reinstalling the operating system on a workstation, the workstation is unable to authenticate even though the technician used the same computer name. Because the new installation generated a new SID and because the new computer does not know the computer account password in the domain, it does not belong to the domain and cannot authenticate to the domain.
- A computer is completely restored from backup and is unable to authenticate. It is likely that the computer changed its password with the domain after the backup operation. Computers change their passwords every 30 days, and Active Directory remembers the current and previous password. If the restore operation restored the computer with a significantly outdated password, the computer will not be able to authenticate.
- A computer's LSA secret gets out of synch with the password known by the domain. You can think of this as the computer forgetting its password, although it did not forget its password; it just disagrees with the domain over what the password really is. When this happens, the computer cannot authenticate and the secure channel cannot be created.

The most common signs of computer account problems are:

- Messages at logon indicate that a domain controller cannot be contacted, that the computer account might be missing, that the password on the computer account is incorrect, or that the trust (another way of saying "the secure relationship") between the computer and the domain has been lost. An example is shown in Figure 5-7.



Figure 5-7 An error message indicating a failed secure channel

- Error messages or events in the event log indicating similar problems or suggesting that passwords, trusts, secure channels, or relationships with the domain or a domain controller have failed. One such error is NETLOGON Event ID 3210: Failed To Authenticate, which appears in the computer's event log.
- A computer account is missing in Active Directory.

Resetting a Computer Account

When the secure channel fails, you must reset it. Many administrators do so by removing the computer from the domain, putting it in a workgroup, and then rejoining the domain. This is not a good practice because it has the potential to delete the computer account altogether, which loses the computer's SID and, more important, its group memberships. When you rejoin the domain, even though the computer has the same name, the account has a new SID, and all the group memberships of the previous computer object must be re-created.

NOTE Do not remove a computer from the domain and rejoin it

If the trust with the domain has been lost, do not remove a computer from the domain and rejoin it. Instead, reset the secure channel.

To reset the secure channel between a domain member and the domain, use the Active Directory Users and Computers snap-in, *Dsmod.exe*, *Netdom.exe*, or *Nltest.exe*. By resetting the account, the computer's SID remains the same and it maintains its group memberships.

- **The Active Directory Users and Computers snap-in** Right-click a computer and choose Reset Account. Click Yes to confirm your choice. The computer will then need to be rejoined to the domain, requiring a reboot.
- **Dsmod** Type the command, **dsmod computer "Computer DN" -reset**. You will have to rejoin the computer to the domain and reboot the computer.
- **Netdom** Type the command **netdom reset MachineName /domain DomainName /UserO UserName /PasswordO {Password | *}** where the credentials belong to the local Administrators group of the computer. This command resets the secure channel by

attempting to reset the password on both the computer and the domain, so it does not require rejoining or rebooting.

- **Nltest** On the computer that has lost its trust, type the command **nltest /server:ServerName /sc_reset:DOMAIN\DomainController**, for example, **nltest /server:SERVER02 /sc_reset:CONTOSO\SERVER01**. This command, like *Netdom.exe*, attempts to reset the secure channel by resetting the password both on the computer and in the domain, so it does not require rejoining or rebooting.

Because *Nltest.exe* and *Netdom.exe* reset the secure channel without requiring a reboot, try those commands first. Only if not successful should you use the *Reset Account* command or *Dsmod* to reset the computer account.

Quick Check

- A user complains that when she attempts to log on, she receives an error message indicating the trust with the domain has been lost. You want to attempt to reset the secure channel without rebooting her system. Which two commands can you use?

Quick Check Answer

- The *Netdom.exe* and *Nltest.exe* commands reset the secure channel without requiring you to rejoin the computer to the domain and, therefore, they require no reboot.

Renaming a Computer

When you rename a computer, you must be careful to do it correctly. Remember that the computer uses its name to authenticate with the domain, so if you rename only the domain object, or only the computer itself, they will be out of synch. You must rename the computer in such a way that both the computer and the domain object are changed.

You can rename a computer correctly by logging on to the computer itself, either locally or with a remote desktop session. Open the System properties from Control Panel and, in the Computer Name, Domain, And Workgroup Settings section, click Change Settings. Click Continue if prompted, and then click the Change button on the Computer Name tab.

From the command prompt, you can use the *Netdom* command with the following syntax:

```
netdom renamecomputer MachineName /NewName:NewName
  [/User0:LocalUsername] [/Password0:{LocalPassword*} ]
  [/UserD:DomainUsername] [/PasswordD:{DomainPassword*} ]
  [/SecurePasswordPrompt] [/REBoot[:TimeInSeconds]]
```

In addition to specifying the computer to rename (*MachineName*) and the desired new name (*NewName*), you must have credentials that are a member of the local Administrators group on the computer and credentials that have permission to rename the domain computer object. By

default, *Netdom.exe* will use the credentials with which the command is executed. You can specify credentials, using *UserO* and *PasswordO* for the credentials in the computer's local Administrators group, and *UserD* and *PasswordD* for the domain credentials with permission to rename the computer object. Specifying * for the password causes *Netdom.exe* to prompt for the password on the command line. The *SecurePasswordPrompt* parameter displays a pop-up for credentials when * is specified for either *PasswordO* or *PasswordD*. After you rename a computer, you must reboot it. The *REBoot* parameter causes the system to reboot after 30 seconds unless otherwise specified by *TimeInSeconds*.

When you rename a computer, you can adversely affect services running on it. For example, Active Directory Certificate Services (AD CS) relies on the server's name. Be certain to consider the impact of renaming a computer before doing so. Do not use these methods to rename a domain controller.

Disabling and Enabling Computer Accounts

If a computer is taken offline or is not to be used for an extended period of time, consider disabling the account. This recommendation reflects the security principle that an identity store allows authentication only of the minimum number of accounts required to achieve the goals of an organization. Disabling the account does not modify the computer's SID or group membership, so when the computer is brought back online, the account can be enabled.

You can disable a computer by right-clicking it and choosing *Disable Account*. A disabled account appears with a down-arrow icon in the Active Directory Users and Computers snap-in, as shown in Figure 5-8.



Figure 5-8 A disabled computer account

While an account is disabled, the computer cannot create a secure channel with the domain. The result is that users who have not previously logged on to the computer, and who, therefore, do not have cached credentials on the computer, will be unable to log on until the secure channel is reestablished by enabling the account.

To enable a computer account, simply select the computer and choose the *Enable Account* command from the context menu.

To disable or enable a computer from the command prompt, use the *Dsmod* command. The syntax used to disable or enable computers is:

```
DSMOD COMPUTER ComputerDN -DISABLED YES  
DSMOD COMPUTER ComputerDN -DISABLED NO
```

Deleting Computer Accounts

You have learned that computer accounts, like user accounts, maintain a unique SID, which enables an administrator to grant permissions to computers. Also like user accounts, computers can belong to groups. Therefore, like user accounts, it is important to understand the effect of deleting a computer account. When a computer account is deleted, its group memberships and SID are lost. If the deletion is accidental, and another computer account is created with the same name, it is nonetheless a new account with a new SID. Group memberships must be reestablished, and any permissions assigned to the deleted computer must be reassigned to the new account. Delete computer objects only when you are certain that you no longer require those security-related attributes of the object.

To delete a computer account using Active Directory Users and Computers, right-click the computer object and, from the context menu, choose the *Delete* command. You will be prompted to confirm the deletion and, because deletion is not reversible, the default response to the prompt is No. Select Yes, and the object is deleted.

The *Dsrms* command introduced in Chapter 3 enables you to delete a computer object from the command prompt. To delete a computer with *Dsrms*, type:

DSRM *ObjectDN*

where *ObjectDN* is the distinguished name of the computer, such as “CN=Desktop154, OU=Clients,DC=contoso,DC=com.” Again, you will be prompted to confirm the deletion.

Recycling Computers

If a computer account’s group memberships and SID, and the permissions assigned to that SID, are important to the operations of a domain, you do not want to delete that account. So what would you do if a computer was replaced with a new system with upgraded hardware? Such is another scenario in which you would reset a computer account.

Resetting a computer account resets its password but maintains all the computer object’s properties. With a reset password, the account becomes, in effect, available for use. Any computer can then join the domain using that account, including the upgraded system. In effect, you’ve recycled the computer account, assigning it to a new piece of hardware. You can even rename the account. The SID and group memberships remain.

As you learned earlier in this lesson, the *Reset Account* command is available in the context menu when you right-click a computer object. The *Dsmod* command can also be used to reset a computer account by typing **dsmod computer "ComputerDN" -reset**.

Group Policy Infrastructure

In Chapter 1, “Installation,” you learned that Active Directory Domain Services (AD DS) provides the foundational services of an identity and access solution for enterprise networks running Microsoft Windows and that AD DS goes further to support the management and configuration of even the largest, most complex networks. Chapter 2, “Administration,” Chapter 3, “Users,” Chapter 4, “Groups,” and Chapter 5, “Computers,” focused on the administration of Active Directory directory service security principals: users, groups, and computers. Now you will begin an examination of the management and configuration of users and computers by using Group Policy. Group Policy provides an infrastructure within which settings can be defined centrally and deployed to users and computers in the enterprise.

In an environment managed by a well-implemented Group Policy infrastructure, little or no configuration needs to be made by directly touching a desktop. All configuration is defined, enforced, and updated using settings in Group Policy objects (GPOs) that affect a portion of the enterprise as broad as an entire site or domain or as narrow as a single organizational unit (OU) or group. In this chapter, you will learn what Group Policy is, how it works, and how best to implement Group Policy in your organization. The remaining chapters in this training kit will apply Group Policy to specific management tasks such as security configuration, software deployment, password policy, and auditing.

Exam objectives in this chapter:

- Creating and Maintaining Active Directory Objects
 - Create and apply Group Policy objects (GPOs).
 - Configure GPO templates.
- Maintaining the Active Directory Environment
 - Monitor Active Directory.

Before You Begin

To complete the practices in this chapter, you must have created a domain controller named SERVER01 in a domain named *contoso.com*. See Chapter 1 for detailed steps for this task.

Real World

Dan Holme

Many of my clients are attempting to do more with less: to increase security, decrease costs, and increase user productivity. All these goals are easier to achieve when you are able to manage change and configuration in your organization. When a new security concern arises, you want to be able to respond quickly to plug any holes. When help desk logs indicate a high number of calls from users requiring help to configure something on their systems, you want to be able to deploy a change centrally that proactively helps users work more effectively. If a new piece of software is required to win new business, you want to deploy it quickly. These are just a few examples of the types of change and configuration management I see tackled every day in enterprises large and small. Group Policy is a phenomenal technology that can deliver a great amount of value to an organization. Too often, I see Group Policy underused or poorly designed. In this chapter, you will learn the workings of Group Policy. Not only will your knowledge help you answer a number of Group Policy questions on the certification exam, but your expertise in Group Policy will be a great asset to your IT organization.

Lesson 1: Implementing Group Policy

A Group Policy infrastructure has a lot of moving parts. It is important that you understand not only what each part does but also how the parts work together and why you might want to assemble them in various configurations. In this lesson, you will get a comprehensive overview of Group Policy: its components, its functions, and its inner workings.

After this lesson, you will be able to:

- Identify the components of Group Policy.
- Explain the fundamentals of Group Policy processing.
- Create, edit, and link Group Policy objects.
- Create the central store for administrative templates.
- Search for specific policy settings in a GPO.
- Create a GPO from a Starter GPO.

Estimated lesson time: 90 minutes

An Overview and Review of Group Policy

Group Policy is a feature of Windows that enables you to manage change and configuration for users and computers from a central point of administration. If you are less familiar with the concepts of Group Policy, it is helpful to keep in mind at all times that Group Policy is all about configuring a setting for one or more users or one or more computers. There are thousands of configuration settings that can be managed with Group Policy, using one infrastructure that is administered with one set of tools.

Policy Settings

The most granular component of the Group Policy is an individual *policy setting*, also known simply as a *policy*, that defines a specific configuration change to apply. For example, a policy setting exists that prevents a user from accessing registry editing tools. If you define that policy setting and apply it to the user, the user will be unable to run tools such as *Regedit.exe*. Another policy setting is available that allows you to disable the local Administrator account. You can use this policy setting to disable the Administrator account on all user desktops and laptops, for example.

These two examples illustrate an important point: that some policy settings affect a user, regardless of the computer to which the user logs on, and other policy settings affect a computer, regardless of which user logs on to that computer. Policy settings such as the setting that prevents access to registry editing tools are often referred to as *user configuration settings* or *user settings*. The policy setting that disables the Administrator account and similar settings are often referred to as *computer configuration settings* or *computer settings*.

Group Policy Objects (GPOs)

Policy settings are defined and exist within a *Group Policy object (GPO)*. A GPO is an object that contains one or more policy settings and thereby applies one or more configuration settings for a user or computer.

Creating and Managing GPOs

GPOs can be managed in Active Directory by using the Group Policy Management console (GPMC), shown in Figure 6-1. They are displayed in a container named Group Policy Objects. Right-click the Group Policy Objects container and choose New to create a GPO.

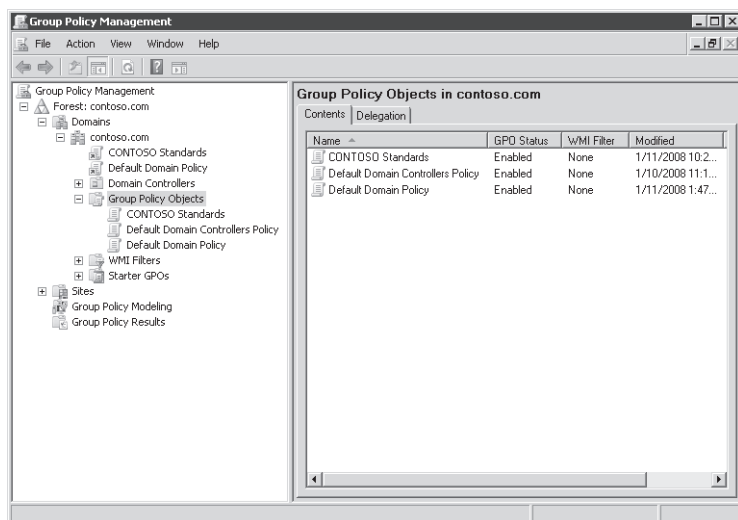


Figure 6-1 The Group Policy Management console

Editing a GPO

To modify the settings of a GPO, right-click the GPO and choose Edit. The GPO opens in the Group Policy Management Editor (GPME) snap-in, formerly known as the Group Policy Object Editor (GPO Editor), shown in Figure 6-2.

The GPME displays the thousands of policy settings available in a GPO in an organized hierarchy that begins with the division between computer settings and user settings: the Computer Configuration node and the User Configuration node. The next levels of the hierarchy are two nodes called Policies and Preferences. You will learn about the difference between these two nodes as this lesson progresses. Drilling deeper into the hierarchy, the GPME displays folders, also called nodes or policy setting groups. Within the folders are the policy settings themselves. Prevent Access To Registry Editing Tools is selected in Figure 6-2. To

define a policy setting, double-click the policy setting. The policy setting's Properties dialog box appears, as shown in Figure 6-3.

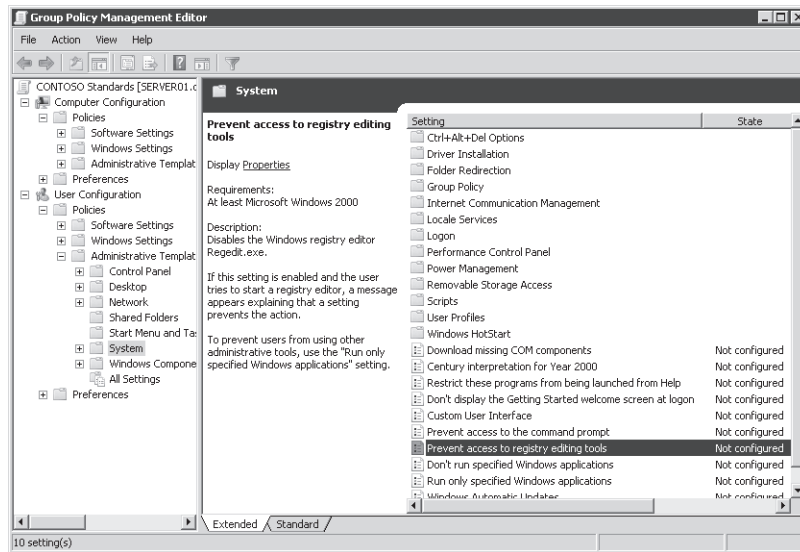


Figure 6-2 Group Policy Management Editor

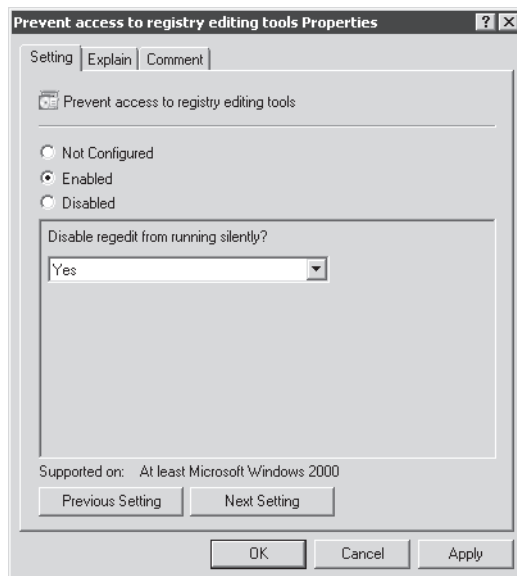


Figure 6-3 The Properties dialog box of a policy setting

Configuring a Policy Setting

A policy setting can have three states: Not Configured, Enabled, and Disabled. As you can see in Figure 6-2, in a new GPO every policy setting is Not Configured. This means that the GPO will not modify the existing configuration of that particular setting for a user or computer. If you enable or disable a policy setting, a change will be made to the configuration of users and computers to which the GPO is applied. The effect of the change depends on the policy setting itself. For example, if you enable the Prevent Access To Registry Editing Tools policy setting, users will be unable to launch the *Regedit.exe* Registry Editor. If you disable the policy setting, you ensure that users can launch the Registry Editor. Notice the double negative in this policy setting: You disable a policy that prevents an action, so you allow the action.

NOTE Understand and test all policy settings

Many policy settings are complex, and the effect of enabling or disabling them might not be immediately clear. Also, some policy settings affect only certain versions of Windows. Be sure to review a policy setting's explanatory text in the GPME detail pane, shown in Figure 6-2, or on the Explain tab of the policy setting's Properties dialog box seen in Figure 6-3. Additionally, always test the effects of a policy setting, and its interactions with other policy settings, before deploying a change in the production environment.

Some policy settings bundle several configurations into one policy and might require additional parameters. In Figure 6-3, you can see that by enabling the policy to restrict registry editing tools, you can also define whether registry files can be merged into the system silently, using *regedit /s*.

Scope

Configuration is defined by policy settings in Group Policy objects. However, the configuration changes in a GPO do not affect computers or users in your enterprise until you have specified the computers or users to which the GPO applies. This is called *scoping* a GPO. The *scope* of a GPO is the collection of users and computers that will apply the settings in the GPO.

You can use several methods to manage the scope of GPOs. The first is the *GPO link*. GPOs can be linked to sites, domains, and OUs in Active Directory. The site, domain, or OU then becomes the maximum scope of the GPO. All computers and users within the site, domain, or OU, including those in child OUs, will be affected by the configurations specified by policy settings in the GPO. A single GPO can be linked to more than one site or OU.

You can further narrow the scope of the GPO with one of two types of filters: *security filters* that specify global security groups to which the GPO should or should not apply, and *Windows Management Instrumentation (WMI) filters* that specify a scope, using characteristics of a system such as operating system version or free disk space. Use Security filters and WMI filters to narrow or specify the scope within the initial scope created by the GPO link.

Scoping GPOs is detailed in Lesson 2, “Managing Group Policy Scope.”

Resultant Set of Policy

Computers and users within the scope of a GPO will apply the policy settings specified in the GPO. An individual user or computer is likely to be within the scope of multiple GPOs linked to the sites, domain, or OUs in which the user or computer exists. This leads to the possibility that policy settings might be configured differently in multiple GPOs. You must be able to understand and evaluate the *Resultant Set of Policy (RSoP)*, which determines the settings that are applied by a client when the settings are configured divergently in more than one GPO. RSoP will be examined in Lesson 3, “Supporting Group Policy.”

Group Policy Refresh

When are policies applied? Policy settings in the Computer Configuration node are applied at system startup and every 90–120 minutes thereafter. User Configuration policy settings are applied at logon and every 90–120 minutes thereafter. The application of policies is called *Group Policy refresh*.

Manually Refreshing Group Policy with Gpupdate

When you are experimenting with Group Policy or trying to troubleshoot Group Policy processing, you might need to initiate a Group Policy refresh manually so that you do not have to wait for the next background refresh. The *Gpupdate.exe* command can be used to initiate a Group Policy refresh. Used on its own, *Gpupdate.exe* triggers processing identical to a background Group Policy refresh. Both computer policy and user policy is refreshed. Use the */target:computer* or */target:user* parameter to limit the refresh to computer or user settings, respectively. During background refresh, by default, settings are applied only if the GPO has been updated. The */force* switch causes the system to reapply all settings in all GPOs scoped to the user or computer. Some policy settings require a logoff or reboot before they actually take effect. The */logoff* and */boot* switches of *Gpupdate.exe* cause a logoff or reboot, respectively, if settings are applied that require one. In Windows 2000, the *Secedit.exe* command was used to refresh policy, so you might encounter a mention of the *Secedit.exe* command on the exam.

Group Policy Client and Client-Side Extensions

And how, exactly, are the policy settings applied? When Group Policy refresh begins, a service running on all Windows systems (called the Group Policy client in Windows Vista and Windows Server 2008) determines which GPOs apply to the computer or user. It downloads any GPOs that it does not already have cached. Then a series of processes called *client-side extensions (CSEs)* do the work of interpreting the settings in a GPO and making appropriate changes to the local computer or to the currently logged-on user. There are CSEs for each major category of policy setting. For example, a CSE applies security changes, a CSE executes

startup and logon scripts, a CSE installs software, and a CSE makes changes to registry keys and values. Each version of Windows has added CSEs to extend the functional reach of Group Policy. Several dozen CSEs are now in Windows Server 2008. One of the more important concepts to remember about Group Policy is that it is really client driven. The Group Policy client pulls the GPOs from the domain, triggering the CSEs to apply settings locally. Group Policy is not a “push” technology.

The behavior of CSEs can be configured using Group Policy, in fact. Most CSEs will apply settings in a GPO only if that GPO has changed. This behavior improves overall policy processing by eliminating redundant applications of the same settings. Most policies are applied in such a way that standard users cannot change the setting on their system—they will always be subject to the configuration enforced by Group Policy. However, some settings can be changed by standard users, and many can be changed if a user is an Administrator on that system. If users in your environment are administrators on their computers, consider configuring CSEs to reapply policy settings even if the GPO has not changed. That way, if an administrative user changes a configuration so that it is no longer compliant with policy, the configuration will be reset to its compliant state at the next Group Policy refresh.

NOTE Configure CSEs to reapply policy settings even if the GPO has not changed

You can configure CSEs to reapply policy settings, even if the GPO has not changed, at background refresh. To do so, configure a GPO scoped to computers and define the settings in the Computer Configuration\Policies\Administrative Templates\System\ Group Policy node. For each CSE you want to configure, open its policy processing policy setting, for example, Registry Policy Processing for the Registry CSE. Click Enabled and select the check box labeled Process Even If The Group Policy Objects Have Not Changed.

An important exception to the default policy processing settings is settings managed by the Security CSE. Security settings are reapplied every 16 hours even if a GPO has not changed.

NOTE The Always Wait For Network At Startup And Logon policy setting

It is highly recommended that you enable the Always Wait For Network At Startup And Logon policy setting for all Windows XP and Windows Vista clients. Without this setting, by default, Windows XP and Windows Vista clients perform only background refreshes, meaning that a client might start up and a user might log on without receiving the latest policies from the domain. The setting is located in Computer Configuration\Policies\Administrative Templates\System\Logon. Be sure to read the policy setting’s explanatory text.

Slow Links and Disconnected Systems

One of the tasks that can be automated and managed with Group Policy is software installation. Group Policy Software Installation (GPSI) is supported by the software installation CSE. You can configure a GPO to install one or more software packages. Imagine, however, if a user

were to connect to your network over a slow connection. You would not want large software packages to be transferred over the slow link because performance would be problematic.

The Group Policy client addresses this concern by detecting the speed of the connection to the domain and determining whether the connection should be considered a slow link. That determination is then used by each CSE to decide whether to apply settings. The software extension, for example, is configured to forgo policy processing so that software is not installed if a slow link is detected. By default, a link is considered to be slow if it is less than 500 kilobits per second (kbps).

If a user is working disconnected from the network, the settings previously applied by Group Policy will continue to take effect, so a user's experience is identical whether on the network or working away from the network. There are exceptions to this rule, most notably that startup, logon, logoff, and shutdown scripts will not run if the user is disconnected.

If a remote user connects to the network on a Windows Vista or Windows Server 2008 system, the Group Policy client wakes up and determines whether a Group Policy refresh window has been missed. If so, it performs a Group Policy refresh to obtain the latest GPOs from the domain. Again, the CSEs determine, based on their policy processing settings, whether settings in those GPOs are applied.

Group Policy Objects

Now that you have a broad-stroke understanding of Group Policy and its components, you can look more closely at each component. In this section, you will examine GPOs in detail. To manage configuration for users and computers, you create GPOs that contain the policy settings you require. Each computer has several GPOs stored locally on the system—the *local GPOs*—and can be within the scope of any number of domain-based GPOs.

Local GPOs

Computers running Windows 2000, Windows XP, and Microsoft Windows Server 2003 each have one local GPO, which can manage configuration of that system. The local GPO exists whether or not the computer is part of domain, workgroup, or a non-networked environment. It is stored in %SystemRoot%\System32\GroupPolicy. The policies in the local GPO affect only the computer on which the GPO is stored. By default, only the Security Settings policies are configured on a system's local GPO. All other policies are set at Not Configured.

When a computer does not belong to an Active Directory domain, the local policy is useful to configure and enforce configuration on that computer. However, in an Active Directory domain, settings in GPOs that are linked to the site, domain, or OUs will override local GPO settings and are easier to manage than GPOs on individual computers.

Windows Vista and Windows Server 2008 systems have multiple local GPOs. The Local Computer GPO is the same as the GPO in previous versions of Windows. In the Computer

Configuration node, configure all computer-related settings. In the User Configuration node, configure settings you want to apply to all users on the computer. The user settings in the Local Computer GPO can be modified by the user settings in two new local GPOs: Administrators and Non-Administrators. These two GPOs apply user settings to a logged-on user who is a member of the local Administrators group or is not, respectively. You can further refine user settings with a local GPO that applies to a specific user account. User-specific local GPOs are associated with local, not domain, user accounts.

RSOP is easy for computer settings: the Local Computer GPO is the only local GPO that can apply computer settings. User settings in a user-specific GPO will override conflicting settings in the Administrators and Non-Administrators GPOs, which themselves override settings in the Local Computer GPO. The concept is simple: the more specific the local GPO, the higher the precedence of its settings.

To create and edit local GPOs, click the Start button and, in the Start Search box, type **mmc.exe**. An empty Microsoft Management console (MMC) opens. Click File and choose Add/Remove Snap-in. Select the Group Policy Object Editor and click Add. A dialog box will appear, prompting you to select the GPO to edit. The Local Computer GPO is selected by default. If you want to edit another local GPO, click the Browse button. On the Users tab, you will find the Non-Administrators and Administrators GPOs and one GPO for each local user. Select the GPO and click OK. Click Finish and then OK to close each of the dialog boxes, and the Group Policy Object editor will be added, focused on the selected GPO.

Keep in mind that local GPOs are designed for nondomain environments. Configure them for your computer at home, for example, to manage the settings for your spouse or children. In a domain environment, settings in domain-based GPOs override conflicting settings in local GPOs, and it is a best practice to manage configuration by using domain-based GPOs.

Domain-Based GPOs

Domain-based GPOs are created in Active Directory and stored on domain controllers. They are used to manage configuration centrally for users and computers in the domain. The remainder of this training kit refers to domain-based GPOs rather than to local GPOs, unless otherwise specified.

When AD DS is installed, two default GPOs are created:

- **Default Domain Policy** This GPO is linked to the domain and has no security group or WMI filters. Therefore, it affects all users and computers in the domain (including computers that are domain controllers). This GPO contains policy settings that specify password, account lockout, and Kerberos policies. As discussed in Chapter 8, “Authentication,” modify the existing settings to align with your enterprise password and account lockout policies but do not add unrelated policy settings to this GPO. If you

need to configure other settings to apply broadly in your domain, create additional GPOs linked to the domain.

- **Default Domain Controllers Policy** This GPO is linked to the Domain Controllers OU. Because computer accounts for domain controllers are kept exclusively in the Domain Controllers OU, and other computer accounts should be kept in other OUs, this GPO affects only domain controllers. The Default Domain Controllers GPO should be modified to implement your auditing policies, as discussed in Chapter 7, “Group Policy Settings,” and in Chapter 8. It should also be modified to assign user rights required on domain controllers.

Creating, Linking, and Editing GPOs

To create a GPO, right-click the Group Policy Objects container and choose New. You must have permission to the Group Policy Objects container to create a GPO.

By default, the Domain Admins group and the Group Policy Creator Owners group are delegated the ability to create GPOs. To delegate permission to other groups, select the Group Policy Objects container in the GPME console tree and then click the Delegation tab in the console details pane.

After you have created a GPO, you can create the initial scope of the GPO by linking it to a site, domain, or OU. To link a GPO, right-click the container and choose Link An Existing GPO. Note that you will not see your sites in the Sites node of the GPMC until you right-click Sites, choose Show Sites, and select the Sites you want to manage. You can also create and link a GPO with a single step by right-clicking a site, domain, or OU and choosing Create A GPO In This Domain And Link It Here.

You must have permission to link GPOs to a site, domain, or OU. In the GPMC, select the container in the console tree and then click the Delegation tab in the console details pane. From the Permission drop-down list, select Link GPOs. The users and groups displayed hold the permission for the selected OU. Click the Add or Remove buttons to modify the delegation.

To edit a GPO, right-click the GPO in the Group Policy Objects container and choose Edit. The GPO is opened in the GPME. You must have at least Read permission to open the GPO in this way. To make changes to a GPO, you must have Write permission to the GPO. Permissions for the GPO can be set by selecting the GPO in the Group Policy Objects container and then clicking the Delegation tab in the details pane.

The GPME will display the name of the GPO as the root node. The GPME also displays the domain in which the GPO is defined and the server from which the GPO was opened and to which changes will be saved. The root node is in the *GPOName [ServerName]* format. In Figure 6-2, the root node is CONTOSO Standards [SERVER01.contoso.com] Policy. The GPO name is CONTOSO Standards, and it was opened from SERVER01.contoso.com, meaning that the GPO is defined in the *contoso.com* domain.

GPO Storage

Group Policy settings are presented as GPOs in Active Directory user interface tools, but a GPO is actually two components: a Group Policy Container (GPC) and Group Policy Template (GPT). The GPC is an Active Directory object stored in the Group Policy Objects container within the domain naming context of the directory. Like all Active Directory objects, each GPC includes a globally unique identifier (GUID) attribute that uniquely identifies the object within Active Directory. The GPC defines basic attributes of the GPO, but it does not contain any of the settings. The settings are contained in the GPT, a collection of files stored in the SYSVOL of each domain controller in the %SystemRoot%\SYSVOL\Domain\Policies\GPO GUID path, where GPO GUID is the GUID of the GPC. When you make changes to the settings of a GPO, the changes are saved to the GPT of the server from which the GPO was opened.

By default, when Group Policy refresh occurs, the CSEs apply settings in a GPO only if the GPO has been updated. The Group Policy client can identify an updated GPO by its version number. Each GPO has a version number that is incremented each time a change is made. The version number is stored as an attribute of the GPC and in a text file, GPT.ini, in the GPT folder. The Group Policy client knows the version number of each GPO it has previously applied. If, during Group Policy refresh, it discovers that the version number of the GPC has been changed, the CSEs will be informed that the GPO is updated.

Quick Check

- Describe the default Group Policy processing behavior, including refresh intervals and CSE application of policy settings.

Quick Check Answer

- Every 90–120 minutes, the Group Policy Client service determines which GPOs are scoped to the user or computer and downloads any GPOs that have been updated, based on the GPOs' version numbers. CSEs process the policies in the GPOs according to their policy processing configuration. By default, most CSEs apply policy settings only if a GPO has been updated. Some CSEs also do not apply settings if a slow link is detected.

GPO Replication

The two parts of a GPO are replicated between domain controllers by using distinct mechanisms. The GPC in Active Directory is replicated by the Directory Replication Agent (DRA), using a topology generated by the Knowledge Consistency Checker (KCC). You will learn more about these services in Chapter 11, “Sites and Replication.” The result is that the GPC is replicated within seconds to all domain controllers in a site and is replicated between sites based on your intersite replication configuration, which will also be discussed in Chapter 11.

The GPT in the SYSVOL is replicated using one of two technologies. The File Replication Service (FRS) is used to replicate SYSVOL in domains running Windows Server 2008, Windows Server 2003, and Windows 2000. If all domain controllers are running Windows Server 2008, you can configure SYSVOL replication, using Distributed File System Replication (DFS-R), a much more efficient and robust mechanism.

Because the GPC and GPT are replicated separately, it is possible for them to become out of synch for a short time. Typically, when this happens, the GPC will replicate to a domain controller first. Systems that obtained their ordered list of GPOs from that domain controller will identify the new GPC, will attempt to download the GPT, and will notice the version numbers are not the same. A policy processing error will be recorded in the event logs. If the reverse happens, and the GPO replicates to a domain controller before the GPC, clients obtaining their ordered list of GPOs from that domain controller will not be notified of the new GPO until the GPC has replicated.

You can download from the Microsoft Download Center the Group Policy Verification Tool, *Gpoutil.exe*, which is part of Windows Resource Kits. This tool reports the status of GPOs in the domain and can identify instances in which, on a domain controller, the GPC and the GPT do not have the same version. For more information about *Gpoutil.exe*, type **gpoutil /?** at the command line.

Exam Tip *Gpoutil.exe* is used to troubleshoot GPO status, including problems caused by the replication of GPOs, leading to inconsistent versions of a GPC and GPT.

Policy Settings

Group Policy settings, also known simply as policies, are contained in a GPO and are viewed and modified using the GPME. In this section, you will look more closely at the categories of settings available in a GPO.

Computer Configuration and User Configuration

There are two major divisions of policy settings: computer settings, contained in the Computer Configuration node, and user settings, contained in the User Configuration node. The Computer Configuration node contains the settings that are applied to computers, regardless of who logs on to them. Computer settings are applied when the operating system starts up and during background refresh every 90–120 minutes thereafter. The User Configuration node contains settings that are applied when a user logs on to the computer and during background refresh every 90–120 minutes thereafter.

Within the Computer Configuration and User Configuration nodes are the Policies and Preferences nodes. Policies are settings that are configured and behave similarly to the policy set-

tings in earlier versions of Windows. Preferences are introduced in Windows Server 2008. The following sections examine these nodes.

Software Settings Node

Within the Policies nodes within Computer Configuration and User Configuration are a hierarchy of folders containing policy settings. Because there are thousands of settings, it is beyond the scope of the exam and of this training kit to examine individual settings. It is worthwhile, however, to define the broad categories of settings in the folders. The first of these nodes is the Software Settings node, which contains only the Software Installation extension. The Software Installation extension helps you specify how applications are installed and maintained within your organization. It also provides a place for independent software vendors to add settings. Software deployment with Group Policy is discussed in Chapter 7.

Windows Settings Node

In both the Computer Configuration and User Configuration nodes, the Policies node contains a Windows Settings node that includes the Scripts, Security Settings, and Policy-Based QoS nodes.

The *Scripts* extension enables you to specify two types of scripts: startup/shutdown (in the Computer Configuration node) and logon/logoff (in the User Configuration node). Startup/shutdown scripts run at computer startup or shutdown. Logon/logoff scripts run when a user logs on or off the computer. When you assign multiple logon/logoff or startup/shutdown scripts to a user or computer, the scripts CSE executes the scripts from top to bottom. You can determine the order of execution for multiple scripts in the Properties dialog box. When a computer is shut down, the CSE first processes logoff scripts, followed by shutdown scripts. By default, the timeout value for processing scripts is 10 minutes. If the logoff and shutdown scripts require more than 10 minutes to process, you must adjust the timeout value with a policy setting. You can use any ActiveX scripting language to write scripts. Some possibilities include Microsoft Visual Basic, Scripting Edition (VBScript), Microsoft JScript, Perl, and Microsoft MS DOS style batch files (.bat and .cmd). Logon scripts on a shared network directory in another forest are supported for network logon across forests.

The Security Settings node allows a security administrator to configure security, using GPOs. This can be done after, or instead of, using a security template to set system security. For a detailed discussion of system security and the Security Settings node, refer to Chapter 7.

The Policy-Based QoS node defines policies that manage network traffic. For example, you might want to ensure that users in the Finance department have priority for running a critical network application during the end-of-year financial reporting period. Policy-Based QoS enables you to do that.

In the User Configuration node only, the Windows Settings folder contains the additional Remote Installation Services, Folder Redirection, and Internet Explorer Maintenance nodes.

Remote Installation Services (RIS) policies control the behavior of a remote operating system installation, using RIS. Folder Redirection enables you to redirect user data and settings folders (AppData, Desktop, Documents, Pictures, Music, and Favorites, for example) from their default user profile location to an alternate location on the network, where they can be centrally managed. *Internet Explorer Maintenance* enables you to administer and customize Microsoft Internet Explorer.

Administrative Templates Node

In both the Computer Configuration and User Configuration nodes, the Administrative Templates node contains registry-based Group Policy settings. There are thousands of such settings available for configuring the user and computer environment. As an administrator, you might spend a significant amount of time manipulating these settings. To assist you with the settings, a description of each policy setting is available in two locations:

- On the Explain tab in the Properties dialog box for the setting. In addition, the Settings tab in the Properties dialog box for the setting lists the required operating system or software for the setting.
- On the Extended tab of the GPME. The Extended tab appears on the bottom of the right details pane and provides a description of each selected setting in a column between the console tree and the settings pane. The required operating system or software for each setting is also listed.

The Administrative Templates node is discussed in detail in the “Administrative Templates” section.

Preferences Node

Underneath both Computer Configuration and User Configuration is a Preferences node. New to Windows Server 2008, preferences provide more than 20 CSEs to help you manage an incredible number of additional settings, including:

- Applications such as Microsoft Office 2003 and Office 2007
- Mapped drives
- Registry settings
- Power options
- Folder options
- Regional options
- Start menu options

Preferences also enables you to deploy the following:

- Files and folders
- Printers

- Scheduled tasks
- Network connections

Many enterprises will also benefit from Preferences because the options can be used to enable or disable hardware devices or classes of devices. For example, you can use Preferences to prevent USB hard drives, including personal media players, from being connected to computers.

The new version of the GPME that supports configuring Preferences is available for download for Windows Vista SP1 from the Microsoft Download Center at <http://www.microsoft.com/downloads>. To apply preferences, systems require the preferences CSEs, which Windows Server 2008 includes. CSEs for Windows XP, Windows Server 2003, and Windows Vista can be downloaded from the Microsoft Download Center.

The interface you use to configure many preferences looks identical to the Windows user interface in which you would make the change manually. Figure 6-4 shows a Folder Options (Windows XP) preference *item*—a collection of settings that are processed by the preferences CSE. You will recognize the similarity to the Folder Options application in Control Panel.

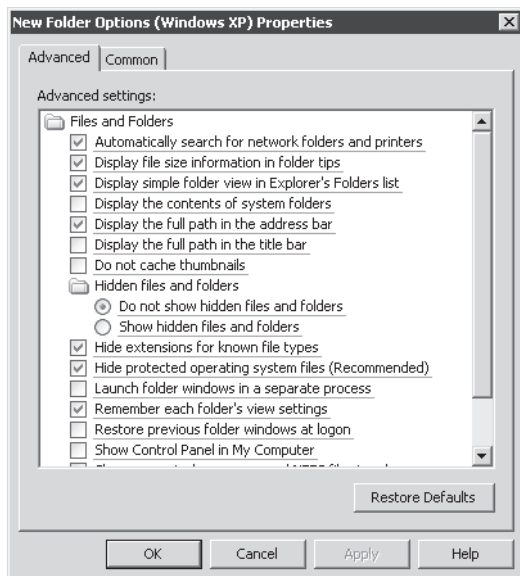


Figure 6-4 A Folder Options preference item

Administrative Templates Node

Policies in the Administrative Templates node in the Computer Configuration node modify registry values in the HKEY_LOCAL_MACHINE (HKLM) key. Policies in the Administrative Templates node in the User Configuration node modify registry values in the

HKEY_CURRENT_USER (HKCU) key. Most of the registry values that are modified by the default policies are located in one of the following four reserved trees:

- HKLM\Software\Policies (computer settings)
- HKCU\Software\Policies (user settings)
- HKLM\Software\Microsoft\Windows\CurrentVersion\Policies (computer settings)
- HKCU\Software\Microsoft\Windows\CurrentVersion\Policies (user settings)

An *administrative template* is a text file that specifies the registry change to be made and that generates the user interface to configure the Administrative Templates policy settings in the GPME. Figure 6-3 shows the properties dialog box for the Prevent Access To Registry Editing Tools. The fact that the setting exists, and that it provides a drop-down list with which to disable *Regedit.exe* from running silently, is determined in an administrative template. The registry setting that is made based on how you configure the policy is also defined in the administrative template.

You can add new administrative templates to the GPME by right-clicking the Administrative Templates node and choosing Add/Remove Templates. Some software vendors provide administrative templates as a mechanism to manage the configuration of their application centrally. For example, you can obtain administrative templates for all recent versions of Microsoft Office from the Microsoft Downloads Center. You can also create your own custom administrative templates. A tutorial on creating custom administrative templates is beyond the scope of this training kit.

In versions of Windows prior to Windows Vista, an administrative template had an .adm extension. ADM files have several drawbacks. First, all localization must be performed within the ADM file. That is, if you want to create an ADM file to help deploy configuration in a multilingual organization, you would need separate ADM files for each language to provide a user interface for administrators who speak that language. If you were to decide later to make a modification related to the registry settings managed by the templates, you would need to make the change to each ADM file.

The second problem with ADM files is the way they are stored. An ADM file is stored as part of the GPT in the SYSVOL. If an ADM file is used in multiple GPOs, it is stored multiple times, contributing to SYSVOL bloat. There were also challenges maintaining version control over ADM files.

In Windows Vista and Windows Server 2008, an administrative template is a pair of XML files, one with an .admx extension that specifies changes to be made to the registry and the other with an .adml extension that provides a language-specific user interface in the GPME. When changes need to be made to settings managed by the administrative template, they can be made to the single ADMX file. Any administrator who modifies a GPO that uses the template accesses the same ADMX file and calls the appropriate ADML file to populate the user interface.

NOTE No need to take sides

ADM and ADMX/ADML administrative templates can coexist.

Central Store

As was previously stated, ADM files are stored as part of the GPO itself. When you edit a GPO that uses administrative templates in the ADM format, the GPME loads the ADM from the GPC to produce the user interface. When ADMX/ADML files are used as administrative templates, the GPO contains only the data that the client needs for processing Group Policy, and when you edit the GPO, the GPME pulls the ADMX and ADML files from the local workstation.

This works well for smaller organizations, but for complex environments that include custom administrative templates or that require more centralized control, Windows Server 2008 introduces the central store. The central store is a single folder in SYSVOL that holds all the ADMX and ADML files that are required. After you have set up the central store, the GPME recognizes it and loads all administrative templates from the central store instead of from the local computer.

To create a central store, create a folder called PolicyDefinitions in the `\\FQDN\SYSVOL\FQDN\Policies` path. For example, the central store for the *contoso.com* domain would be `\\contoso.com\SYSVOL\contoso.com\Policies\PolicyDefinitions`. Then, copy all files from the `%SystemRoot%\PolicyDefinitions` folder of a Windows Server 2008 system to the new SYSVOL PolicyDefinitions folder. These will include the .admx files and the .adml files in a language-specific subfolder of `%SystemRoot%\PolicyDefinitions`. For example, English (United States) ADML files are located in `%SystemRoot%\PolicyDefinitions\en-us`. Copy them into `\\FQDN\SYSVOL\FQDN\Policies\PolicyDefinitions\en-us`. If additional languages are required, copy the folder that contains the ADML files to the central store. When you have copied all ADMX and ADML files, the PolicyDefinitions folder on the domain controller should contain the ADMX files and one or more folders containing language-specific ADML files.

Exam Tip If logging on to a domain controller, locally or by using Remote Desktop, the local path to the PolicyDefinitions folder is `%SystemRoot%\SYSVOL\domain\Policies\PolicyDefinitions`.

Filtering Administrative Template Policy Settings

A weakness of the Group Policy editing tools in previous versions of Windows is the inability to search for a specific policy setting. With thousands of policies to choose from, it can be difficult to locate exactly the setting you want to configure. The new GPME in Windows Server 2008 solves this problem for Administrative Template settings: you can now create filters to locate specific policy settings.

To create a filter, right-click Administrative Templates and choose Filter Options. To locate a specific policy, select Enable Keyword Filters, enter the words with which to filter, and select the fields within which to search. Figure 6-5 shows an example of a search for policy settings related to the screen saver.

In the top section of the Filter Options dialog box shown in Figure 6-5, you can filter the view to show only policy settings that are configured. This can help you locate and modify settings that are already specified in the GPO.

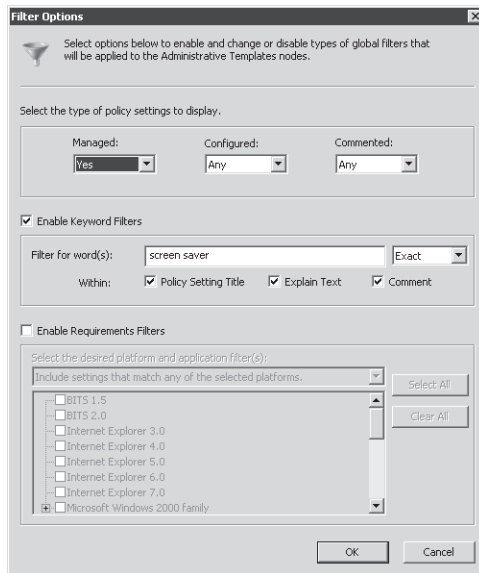


Figure 6-5 Filtering the Administrative Templates policy settings

Commenting

You can also search and filter based on policy-setting comments. Windows Server 2008 enables you to add comments to policy settings in the Administrative Templates node. Double-click a policy setting and click the Comment tab. It is a best practice to add comments to configured policy settings as a way to document the justification for a setting and its intended effect. You should also add comments to the GPO itself. Windows Server 2008 enables you to attach comments to a GPO. In the GPME, right-click the root node in the console tree and choose Properties; then click the Comment tab.

Starter GPOs

Another new Group Policy feature in Windows Server 2008 is starter GPOs. A starter GPO contains Administrative Template settings. You can create a new GPO from a starter GPO, in which case, the new GPO is prepopulated with a copy of the settings in Starter GPO. A starter

GPO is, in effect, a template. Unfortunately, Microsoft had already been using the term *template* in the context of administrative templates, so another name had to be found. When you create a new GPO, you can still choose to begin with a blank GPO, or you can select one of the preexisting starter GPOs or a custom starter GPO.

NOTE When you need more than administrative template settings

Starter GPOs can contain only Administrative Templates policy settings. You can also copy and paste entire GPOs in the Group Policy Objects container of the Group Policy Management console so that you have a new GPO with all the settings of the source GPO. To transfer settings between GPOs in different domains or forests, right-click a GPO and choose Back Up. In the target domain, create a new GPO, right-click it, and choose Import Settings. You will be able to import the settings of the backed-up GPO.

Managed and Unmanaged Policy Settings

There is a nuance to the registry policy settings configured by the Administrative Templates node that is important to understand: the difference between managed and unmanaged policy settings. The registry policy settings that have been discussed so far and that are encountered in the practices of this chapter are examples of managed policy settings. A managed policy setting effects a configuration change of some kind when the setting is applied by a GPO. When the user or computer is no longer within the scope of the GPO, the configuration reverts to its original state automatically. For example, if a GPO prevents access to registry editing tools and then the GPO is deleted, disabled, or scoped so that it no longer applies to users, those users will regain access to registry editing tools at the next policy refresh.

In contrast, an unmanaged policy setting makes a change that is persistent in the registry. If the GPO no longer applies, the setting remains. This is often called *tattooing* the registry. To reverse the effect of the policy setting, you must deploy a change that reverts the configuration to the desired state.

By default, the GPME hides unmanaged policy settings to discourage you from implementing a configuration that is difficult to revert. However, you can make many useful changes with unmanaged policy settings, particularly for custom administrative templates to manage configuration for applications. To control which policy settings are visible, right-click Administrative Templates and choose Filter Options. Make a selection from the Managed drop-down list.

Lesson 2: Managing Group Policy Scope

A GPO is, by itself, just a collection of configuration instructions that will be processed by the CSEs of computers. Until the GPO is scoped, it does not apply to any users or computers. The GPO's scope determines which computers' CSEs will receive and process the GPO, and only the computers or users within the scope of a GPO will apply the settings in that GPO. Several mechanisms are used to scope a GPO:

- The GPO link to a site, domain, or OU and whether that link is enabled
- The Enforce option of a GPO
- The Block Inheritance option on an OU
- Security group filtering
- WMI filtering
- Policy node enabling or disabling
- Preferences targeting
- Loopback policy processing

You must be able to define the users or computers to which configuration is deployed, and therefore, you must master the art of scoping GPOs. In this lesson, you will learn each of the mechanisms with which you can scope a GPO and, in the process, the concepts of Group Policy application, inheritance, and precedence.

After this lesson, you will be able to:

- Manage GPO links.
- Evaluate GPO inheritance and precedence.
- Understand the Block Inheritance and Enforced link options.
- Use security filtering to narrow the scope of a GPO.
- Apply a WMI filter to a GPO.
- Implement loopback policy preferences.

Estimated lesson time: 90 minutes

GPO Links

A GPO can be linked to one or more Active Directory sites, domains, or OUs. After a policy is linked to a site, domain, or OU, the users or computers and users in that container are within the scope of the GPO, including computers and users in child OUs.

As you learned in Lesson 1, you can link a GPO to the domain or to an OU by right-clicking it and choosing Link An Existing GPO. If you have not yet created a GPO, you can choose Create A GPO In This Domain, And Link It Here. You can choose the same commands to link a GPO

to a site, but by default, your Active Directory sites are not visible in the GPME; you must first right-click Sites and choose Show Sites.

Site-Linked GPOs and Domain Controller Placement

A GPO linked to a site affects all computers in the site without regard to the domain to which the computers belong (as long as all computers belong to the same Active Directory forest). Therefore, by linking a GPO to a site, that GPO can be applied to multiple domains within a forest. Site-linked GPOs are stored on domain controllers in the domain in which the GPO was created. Therefore, domain controllers for that domain must be accessible for site-linked GPOs to be applied correctly. If you implement site-linked policies, you must consider policy application when planning your network infrastructure. Either place a domain controller from the GPO's domain in the site to which the policy is linked or ensure that WAN connectivity provides accessibility to a domain controller in the GPO's domain.

When you link a GPO to a site, domain, or OU, you define the initial scope of the GPO. Select a GPO and click the Scope tab to identify the containers to which the GPO is linked. In the details pane of the GPMC, the GPO links are displayed in the first section of the Scope tab, as seen in Figure 6-6.



Figure 6-6 A GPO's links displayed on the Scope tab of the GPMC

The impact of the GPO's links is that the Group Policy client will download the GPO if either the computer or the user objects fall within the scope of the link. The GPO will be downloaded only if it is new or updated. The Group Policy client caches the GPO to make policy refresh more efficient.

Linking a GPO to Multiple OUs

You can link a GPO to more than one site, domain, or OU. It is common, for example, to apply configuration to computers in several OUs. You can define the configuration in a single GPO

and link that GPO to each OU. If you later change settings in the GPO, your changes will apply to all OUs to which the GPO is linked.

Deleting or Disabling a GPO Link

After you have linked a GPO, the GPO link appears in the GPMC underneath this site, domain, or OU. The icon for the GPO link has a small shortcut arrow. When you right-click the GPO link, a context menu appears, as shown in Figure 6-7.

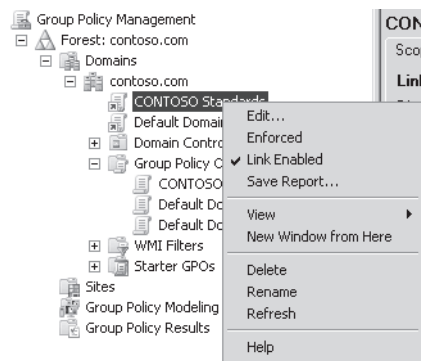


Figure 6-7 The context menu of a GPO link

You can delete a GPO link by choosing Delete from the context menu. Deleting a GPO link does not delete the GPO itself, which remains in that Group Policy Objects container. Deleting the link does change the scope of the GPO so that it no longer applies to computers and users within a site, domain, or OU to which it was previously linked.

You can also modify a GPO link by disabling it. Right-click the GPO link and deselect the Link Enabled option. Disabling the link also changes the scope of the GPO so that it no longer applies to computers and users within that container. However, the link remains so that it can be easily re-enabled.

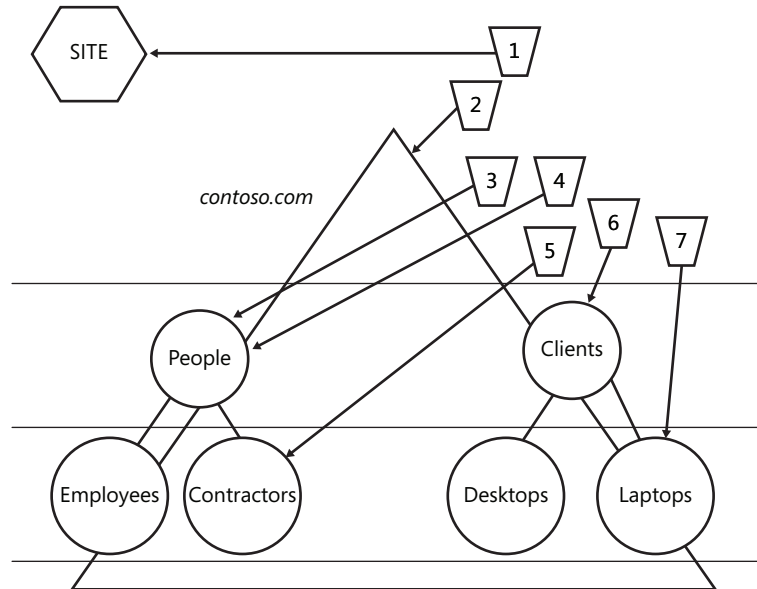
GPO Inheritance and Precedence

A policy setting can be configured in more than one GPO, and GPOs can be in conflict with one another. For example, a policy setting can be enabled in one GPO, disabled in another GPO, and not configured in a third GPO. In this case, the *precedence* of the GPOs determines which policy setting the client applies. A GPO with higher precedence will prevail over a GPO with lower precedence. Precedence is shown as a number in the GPMC. The smaller the number—that is, the closer to 1—the higher the precedence, so a GPO with a precedence of 1 will prevail over other GPOs. Select the domain or OU and then click the Group Policy Inheritance tab to view the precedence of each GPO.

When a policy setting is enabled or disabled in a GPO with higher precedence, the configured setting takes effect. However, remember that policy settings are set to Not Configured by default. If a policy setting is not configured in a GPO with higher precedence, the policy setting (either enabled or disabled) in a GPO with lower precedence will take effect.

A site, domain, or OU can have more than one GPO linked to it. The link order of GPOs determines the precedence of GPOs in such a scenario. GPOs with higher-link order take precedence over GPOs with lower-link order. When you select an OU in the GPMC, the Linked Group Policy Objects tab shows the link order of GPOs linked to that OU.

The default behavior of Group Policy is that GPOs linked to a higher-level container are inherited by lower-level containers. When a computer starts up or a user logs on, the Group Policy client examines the location of the computer or user object in Active Directory and evaluates the GPOs with scopes that include the computer or user. Then the client-side extensions apply policy settings from these GPOs. Policies are applied sequentially, beginning with the policies linked to the site, followed by those linked to the domain, followed by those linked to OUs—from the top-level OU down to the OU in which the user or computer object exists. It is a layered application of settings, so a GPO that is applied later in the process, because it has higher precedence, will override settings applied earlier in the process. This default order of applying GPOs is illustrated in Figure 6-8.



GPO processing order for the Contractors OU = 1, 2, 3, 4, 5
 GPO processing order for the Laptops OU = 1, 2, 6, 7

Figure 6-8 Default processing of site, domain, and OU GPOs

Exam Tip Be certain to memorize the default domain policy processing order: site, domain, OU; remember that domain policy settings are applied after and, therefore, take precedence over settings in local GPOs.

This sequential application of GPOs creates an effect called *policy inheritance*. Policies are inherited, so the resultant set of group policies for a user or computer will be the cumulative effect of site, domain, and OU policies.

By default, inherited GPOs have lower precedence than GPOs linked directly to the container. In a practical example, you might configure a policy setting to disable the use of registry-editing tools for all users in the domain by configuring the policy setting in a GPO linked to the domain. That GPO, and its policy setting, will be inherited by all users within the domain. However, you probably want administrators to be able to use registry-editing tools, so you will link a GPO to the OU that contains administrators' accounts and configure the policy setting to allow the use of registry-editing tools. Because the GPO linked to the administrators' OU takes higher precedence than the inherited GPO, administrators will be able to use registry-editing tools.

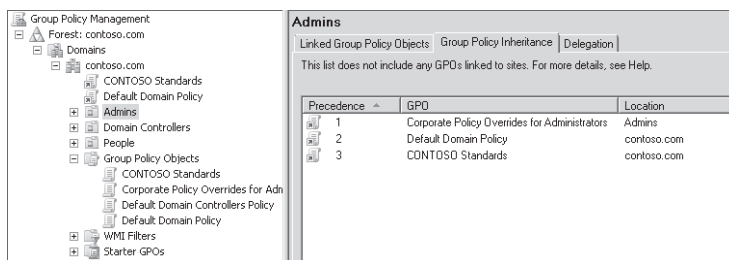


Figure 6-9 The Group Policy inheritance tab

Figure 6-9 shows this example. A policy setting that restricts registry-editing tools is defined in the CONTOSO Standards GPO, linked to the *contoso.com* domain. In the Corporate Policy Overrides For Administrators GPO, a policy setting specifically allows the use of registry-editing tools. The administrator's GPO is linked to the Admins OU. When you select an OU such as the Admins OU, the details pane of the GPMC displays a Group Policy Inheritance tab that reveals GPO precedence for that OU. You can see that the Corporate Policy Overrides For Administrators GPO has precedence. Any setting in that GPO that is in conflict with a setting in CONTOSO Standards will be applied from the administrators GPO. Therefore, users in the Admins OU will be able to use registry editing tools, although users elsewhere in the domain will not be able to. As you can see from this simple example, the default order of precedence ensures that the policy that is closest to the user or computer prevails.

Precedence of Multiple Linked Group Policy Objects

An OU, domain, or site can have more than one GPO linked to it. In the event of multiple Group Policy objects, the objects' *link order* determines their precedence. In Figure 6-10, two GPOs are linked to the People OU. The object higher on the list, with a link order of 1, has the highest precedence. Therefore, settings that are enabled or disabled in the Power User Configuration GPO will have precedence over these same settings in the Standard User Configuration GPO.

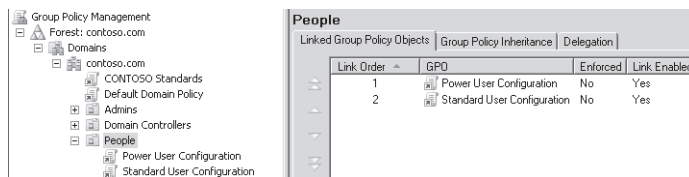


Figure 6-10 GPO link order

Blocking Inheritance

A domain or OU can be configured to prevent the inheritance of policy settings. To block inheritance, right-click the domain or OU in the GPME and choose Block Inheritance.

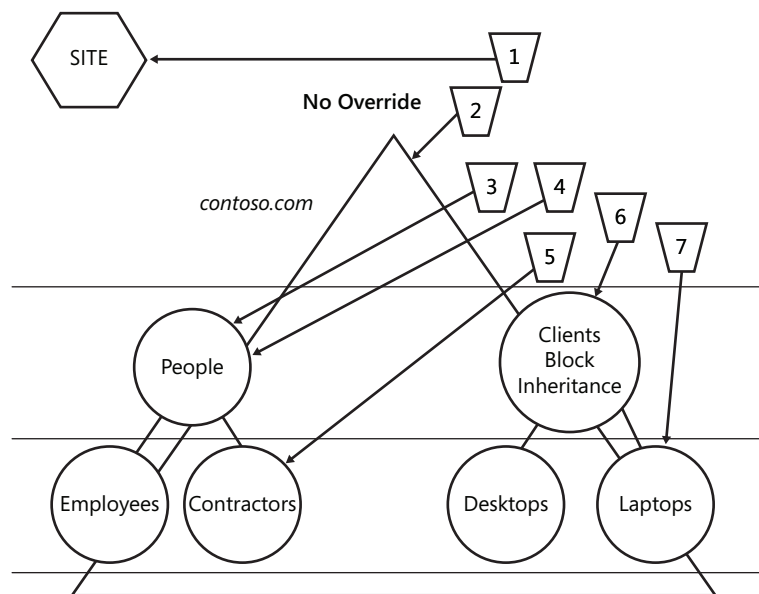
The Block Inheritance option is a property of a domain or OU, so it blocks *all* Group Policy settings from GPOs linked to parents in the Group Policy hierarchy. When you block inheritance on an OU, for example, GPO application begins with any GPOs linked directly to that OU—GPOs linked to higher-level OUs, the domain, or the site will not apply.

The Block Inheritance option should be used sparingly, if ever. Blocking inheritance makes it more difficult to evaluate Group Policy precedence and inheritance. In the section, “Using Security Filtering to Modify GPO Scope,” you will learn how to scope a GPO so that it applies to only a subset of objects or so that it is prevented from applying to a subset of objects. With security group filtering, you can carefully scope a GPO so that it applies to only the correct users and computers in the first place, making it unnecessary to use the Block Inheritance option.

Enforcing a GPO Link

In addition, a GPO link can be set to Enforced. To do this, right-click a GPO link and choose Enforced from the context menu shown in Figure 6-7. When a GPO link is set to Enforced, the GPO takes the highest level of precedence; policy settings in that GPO will prevail over any conflicting policy settings in other GPOs. In addition, a link that is enforced will apply to child containers even when those containers are set to Block Inheritance. The Enforced option causes the policy to apply to all objects within its scope. Enforced will cause policies to override any conflicting policies and will apply regardless of whether a Block Inheritance option is set.

In Figure 6-11, Block Policy Inheritance has been applied to the Clients OU. As a result, GPO 1, which is applied to the site, is blocked and does not apply to the Clients OU. However, GPO 2, linked to the domain with the Enforced option, does apply. In fact, it is applied last in the processing order, meaning that its settings will override those of GPOs 6 and 7.



GPO processing order for the Contractors OU = 1, 3, 4, 5, 2
 GPO processing order for the Laptops OU = 6, 7, 2

Figure 6-11 Policy processing with Block Inheritance and Enforced options

When you configure a GPO that defines configuration mandated by your corporate IT security and usage policies, you want to ensure that those settings are not overridden by other GPOs. You can do this by enforcing the link of the GPO. Figure 6-12 shows just this scenario. Configuration mandated by corporate policies is deployed in the CONTOSO Corporate IT Security & Usage GPO, which is linked with an enforced link to the *contoso.com* domain. The icon for the GPO link has a padlock on it—the visual indicator of an enforced link. On the People OU, the Group Policy Inheritance tab shows that the GPO takes precedence even over the GPOs linked to the People OU itself.

To facilitate evaluation of GPO precedence, you can simply select an OU (or domain) and click the Group Policy Inheritance tab. This tab will display the resulting precedence of GPOs, accounting for GPO link, link order, inheritance blocking, and link enforcement. This tab does not account for policies that are linked to a site, nor does it account for GPO security or WMI filtering.

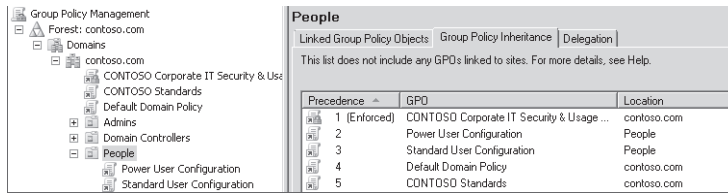


Figure 6-12 The precedence of the GPO with an enforced link

Exam Tip Although it is recommended to use the Block Inheritance and Enforced options sparingly in your Group Policy infrastructure, the 70-640 exam will expect you to understand the effect of both options.

Using Security Filtering to Modify GPO Scope

By now, you've learned that you can link a GPO to a site, domain, or OU. However, you might need to apply GPOs only to certain groups of users or computers rather than to all users or computers within the scope of the GPO. Although you cannot directly link a GPO to a security group, there is a way to apply GPOs to specific security groups. The policies in a GPO apply only to users who have Allow Read and Allow Apply Group Policy permissions to the GPO.

Each GPO has an access control list (ACL) that defines permissions to the GPO. Two permissions, Allow Read and Allow Apply Group Policy are required for a GPO to apply to a user or computer. If a GPO is scoped to a computer, for example, by its link to the computer's OU, but the computer does not have Read and Apply Group Policy permissions, it will not download and apply the GPO. Therefore, by setting the appropriate permissions for security groups, you can filter a GPO so that its settings apply only to the computers and users you specify.

By default, Authenticated Users are given the Allow Apply Group Policy permission on each new GPO. This means that by default, *all* users and computers are affected by the GPOs set for their domain, site, or OU regardless of the other groups in which they might be members. Therefore, there are two ways of filtering GPO scope:

- Remove the Apply Group Policy permission (currently set to Allow) for the Authenticated Users group but do not set this permission to Deny. Then determine the groups to which the GPO should be applied and set the Read and Apply Group Policy permissions for these groups to Allow.
- Determine the groups to which the GPO should not be applied and set the Apply Group Policy permission for these groups to Deny. If you deny the Apply Group Policy permission to a GPO, the user or computer will not apply settings in the GPO, even if the user or computer is a member of another group that is allowed the Apply Group Policy Permission.

Filtering a GPO to Apply to Specific Groups

To apply a GPO to a specific security group, select the GPO in the Group Policy Objects container in the GPMC. In the Security Filtering section, select the Authenticated Users group and click Remove. Click OK to confirm the change and then click Add. Select the group to which you want the policy to apply and click OK. The result will look similar to Figure 6-13—the Authenticated Users group is not listed, and the specific group to which the policy should apply is listed.

NOTE Use global security groups to filter GPOs

GPOs can be filtered only with global security groups—not with domain local security groups.

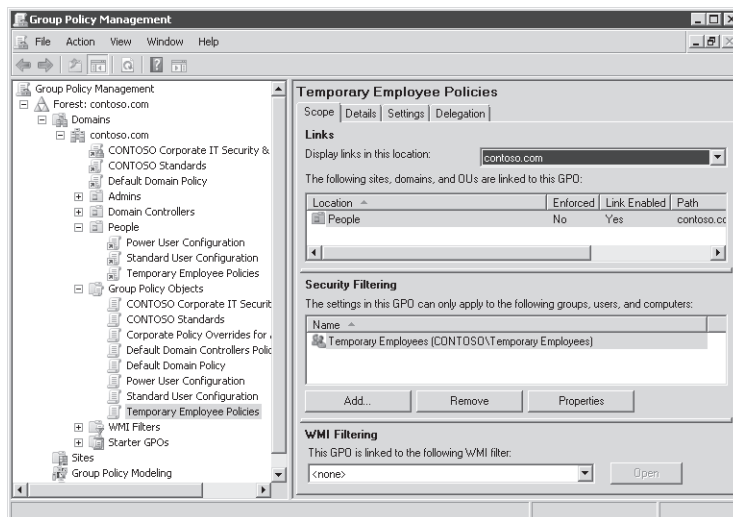


Figure 6-13 Security filtering of a GPO

Filtering a GPO to Exclude Specific Groups

Unfortunately, the Scope tab of a GPO does not allow you to exclude specific groups. To exclude a group—that is, to deny the Apply Group Policy permission—you must click the Delegation tab. Click the Advanced button, and the Security Settings dialog box appears. Click the Add button in the Security Settings dialog box, select the group you want to exclude from the GPO, and click OK. The group you selected is given the Allow Read permission by default. Deselect that permission check box and select the Deny Apply Group Policy. Figure 6-14 shows an example that denies the Help Desk group the Apply Group Policy permission and, therefore, excludes the group from the scope of the GPO.

When you click the OK button in the Security Settings dialog box, you will be warned that Deny permissions override other permissions. Because of this, it is recommended that you use

Deny permissions sparingly. Microsoft Windows reminds you of this best practice with the warning message and by the far more laborious process to exclude groups with the Deny Apply Group Policy permission than to include groups in the Security Filtering section of the Scope tab.

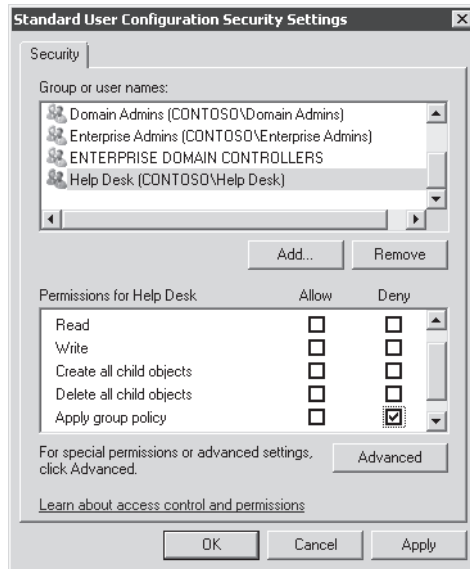


Figure 6-14 Excluding a group from the scope of a GPO with the Deny Apply Group Policy permission

NOTE Deny permissions are not exposed on the Scope tab

Unfortunately, when you exclude a group, the exclusion is not shown in the Security Filtering section of the Scope tab. This is yet one more reason to use Deny permissions sparingly.

WMI Filters

Windows Management Instrumentation (WMI) is a management infrastructure technology that enables administrators to monitor and control managed objects in the network. A WMI query is capable of filtering systems based on characteristics, including RAM, processor speed, disk capacity, IP address, operating system version and service pack level, installed applications, and printer properties. Because WMI exposes almost every property of every object within a computer, the list of attributes that can be used in a WMI query is virtually unlimited. WMI queries are written using WMI query language (WQL).

You can use a WMI query to create a WMI filter, with which a GPO can be filtered. A good way to understand the purpose of a WMI filter, both for the certification exams and for real-world

implementation, is through examples. Group Policy can be used to deploy software applications and service packs—a capability that is discussed in Chapter 7. You might create a GPO to deploy an application and then use a WMI filter to specify that the policy should apply only to computers with a certain operating system and service pack, Windows XP SP3, for example. The WMI query to identify such systems is:

```
Select * FROM Win32_OperatingSystem WHERE Caption="Microsoft  
Windows XP Professional" AND CSDVersion="Service Pack 3"
```

When the Group Policy client evaluates GPOs it has downloaded to determine which should be handed off to the CSEs for processing, it performs the query against the local system. If the system meets the criteria of the query, the query result is a logical *True*, and the CSEs will process the GPO.

WMI exposes *namespaces*, within which are classes that can be queried. Many useful classes, including *Win32_Operating System*, are found in a class called *root\CIMv2*.

To create a WMI filter, right-click the WMI Filters node in the GPME and choose New. Type a name and description for the filter, and then click the Add button. In the Namespace box, type the namespace for your query. In the Query box, enter the query. Then click OK.

To filter a GPO with a WMI filter, click the Scope tab of a GPO, click the WMI drop-down list, and select the WMI filter. A GPO can be filtered by only one WMI filter, but that WMI filter can be a complex query, using multiple criteria. A single WMI filter can be linked to, and thereby used to filter, one or more GPOs. The General tab of a WMI filter, shown in Figure 6-15, displays the GPOs that use the WMI filter.

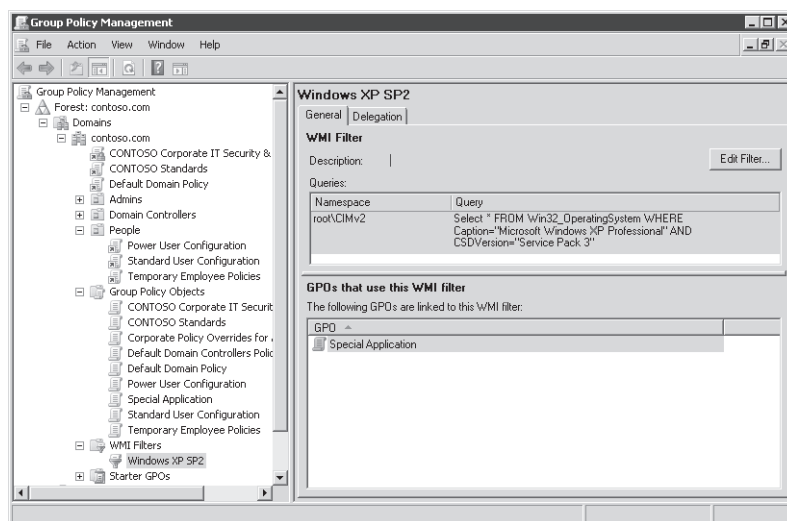


Figure 6-15 A WMI filter

There are three significant caveats regarding WMI filters. First, the WQL syntax of WMI queries can be challenging to master. You can often find examples on the Internet when you search using the keywords *WMI filter* and *WMI query* along with a description of the query you want to create.

MORE INFO WMI filter examples

You can find examples of WMI filters at <http://technet2.microsoft.com/windowsserver/en/library/a16cffa4-83b3-430b-b826-9bf81c0d39a71033.mspx?mfr=true>. You can also refer to the Windows Management Instrumentation (WMI) software development kit (SDK), located at <http://msdn2.microsoft.com/en-us/library/aa394582.aspx>.

Second, WMI filters are expensive in terms of Group Policy processing performance. Because the Group Policy client must perform the WMI query at each policy processing interval, there is a slight impact on system performance every 90–120 minutes. With the performance of today's computers, the impact might not be noticeable, but you should certainly test the effects of a WMI filter prior to deploying it widely in your production environment.

Third, WMI filters are not processed by computers running Windows 2000. If a GPO is filtered with a WMI filter, a Windows 2000 system ignores the filter and processes the GPO as if the results of the filter were *True*.

Exam Tip Although it is unlikely that you will be asked to recognize WQL queries on the 70-640 exam, you should be familiar with the basic functionality of WMI queries as discussed in this section. Be certain to remember that Windows 2000 systems will apply settings in GPOs with WMI filters because Windows 2000 ignores WMI filters during policy processing.

Enabling or Disabling GPOs and GPO Nodes

You can prevent the settings in the Computer Configuration or User Configuration nodes from being processed during policy refresh by changing GPO Status. On the Details tab of a GPO, shown in Figure 6-16, click the GPO Status drop-down list and choose one of the following:

- **Enabled** Both computer configuration settings and user configuration settings will be processed by CSEs during policy refresh.
- **All Settings Disabled** CSEs will not process the GPO to policy refresh.
- **Computer Configuration Settings Disabled** During computer policy refresh, computer configuration settings in the GPO will be applied. The GPO will not be processed during user policy refresh.
- **User Configuration Settings Disabled** During user policy refresh, user configuration settings in the GPO will be applied. The GPO will not be processed during computer policy refresh.

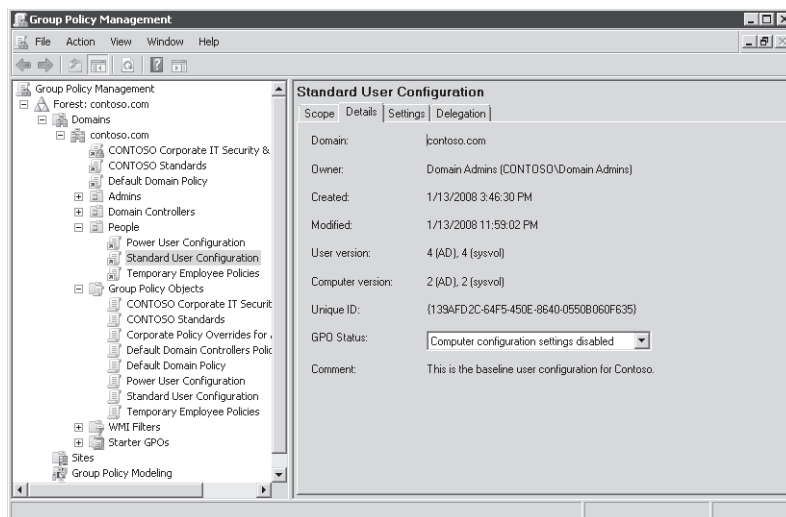


Figure 6-16 The Details tab of a GPO

You can configure GPO Status to optimize policy processing. If a GPO contains only user settings, for example, setting GPO Status to disable computer settings will prevent the Group Policy client from attempting to process the GPO during computer policy refresh. Because the GPO contains no computer settings, there is no need to process the GPO, and you can save a few cycles of the processor.

NOTE Use disabled GPOs for disaster recovery

You can define a configuration that should take effect in case of an emergency, security incident, or other disasters in a GPO and link the GPO so that it is scoped to appropriate users and computers. Then, disable the GPO. In the event that you require the configuration to be deployed, simply enable the GPO.

Targeting Preferences

Preferences, which are new to Windows Server 2008, have a built-in scoping mechanism called *item-level targeting*. You can have multiple preference items in a single GPO, and each preference item can be targeted or filtered. So, for example, you could have a single GPO with a preference that specifies folder options for engineers and another item that specifies folder options for sales people. You can target the items by using a security group or OU. There are over a dozen other criteria that can be used, including hardware and network characteristics, date and time, LDAP queries, and more.

NOTE Preferences can target within a GPO

What's new about preferences is that you can target multiple preferences items within a single GPO instead of requiring multiple GPOs. With traditional policies, you often need multiple GPOs filtered to individual groups to apply variations of settings.

Like WMI filters, item-level targeting of preferences requires the CSE to perform a query to determine whether to apply the settings in a preferences item. You must be aware of the potential performance impact of item-level targeting, particularly if you use options such as LDAP queries, which require processing time and a response from a domain controller to process. As you design your Group Policy infrastructure, balance the configuration management benefits of item-level targeting against the performance impact you discover during testing in a lab.

Group Policy Processing

Now that you have learned more about the concepts, components, and scoping of Group Policy, you are ready to examine Group Policy processing closely. As you read this section, keep in mind that Group Policy is all about applying configurations defined by GPOs, that GPOs are applied in an order (site, domain, and OU), and that GPOs applied later in the order have higher precedence; their settings, when applied, will override settings applied earlier. The following sequence details the process through which settings in a domain-based GPO are applied to affect a computer or user:

1. The computer starts, and the network starts. Remote Procedure Call System Service (RPCSS) and Multiple Universal Naming Convention Provider (MUP) are started. The Group Policy client is started.
2. The Group Policy client obtains an ordered list of GPOs scoped to the computer.

The order of the list determines the order of GPO processing, which is, by default, local, site, domain, and OU:

- a. Local GPOs. Each computer running Windows Server 2003, Windows XP, and Windows 2000 has exactly one GPO stored locally. Windows Vista and Windows Server 2008 have multiple local GPOs. The precedence of local GPOs is discussed in the “Local GPOs” section in Lesson 1.
- b. Site GPOs. Any GPOs that have been linked to the site are added to the ordered list next. When multiple GPOs are linked to a site (or domain or OU), the *link order*, configured on the Scope tab, determines the order in which they are added to the list. The GPO that is highest on the list, with the number closest to 1, has the highest precedence, and is added to the list last. It will, therefore, be applied last, and its settings will override those of GPOs applied earlier.
- c. Domain GPOs. Multiple domain-linked GPOs are added as specified by the link order.

NOTE Domain-linked policies are not inherited by child domains

Policies from a parent domain are not inherited by a child domain. Each domain maintains distinct policy links. However, computers in several domains might be within the scope of a GPO linked to a site.

- d. OU GPOs. GPOs linked to the OU highest in the Active Directory hierarchy are added to the ordered list, followed by GPOs linked to its child OU, and so on. Finally, the GPOs linked to the OU that contains the computer are added. If several group policies are linked to an OU, they are added in the order specified by the link order.
 - e. Enforced GPOs. These are added at the end of the ordered list, so their settings will be applied at the end of the process and will, therefore, override settings of GPOs earlier in the list and in the process. As a point of trivia, enforced GPOs are added to the list in reverse order: OU, domain, and then site. This is relevant when you apply corporate security policies in a domain-linked, enforced GPO. That GPO will be at the end of the ordered list and will be applied last, so its settings will take precedence.
3. The GPOs are processed synchronously in the order specified by the ordered list. This means that settings in the local GPOs are processed first, followed by GPOs linked to the site, the domain, and the OUs containing the user or computer. GPOs linked to the OU of which the computer or user is a direct member are processed last, followed by enforced GPOs.

As each GPO is processed, the system determines whether its settings should be applied based on the GPO status for the computer node (enabled or disabled) and whether the computer has the Allow Group Policy permission. If a WMI filter is applied to the GPO, and if the computer is running Windows XP or later, it performs the WQL query specified in the filter.

4. If the GPO should be applied to the system, CSEs trigger to process the GPO settings. Policy settings in GPOs will overwrite policies of previously applied GPOs in the following ways:
 - ❑ If a policy setting is configured (set to Enabled or Disabled) in a GPO linked to a parent container (OU, domain, or site), and the same policy setting is Not Configured in GPOs linked to its child container, the resultant set of policies for users and computers in the child container will include the parent's policy setting. If the child container is configured with the Block Inheritance option, the parent setting is not inherited unless the GPO link is configured with the Enforced option.
 - ❑ If a policy setting is configured (set to Enabled or Disabled) for a parent container, and the same policy setting is configured for a child, the child container's setting

overrides the setting inherited from the parent. If the parent GPO link is configured with the Enforced option, the parent setting has precedence.

- ❑ If a policy setting of GPOs linked to parent containers is Not Configured, and the child OU setting is also Not Configured, the resultant policy setting is the setting that results from the processing of local GPOs. If the resultant setting of local GPOs is also Not Configured, the resultant configuration is the Windows default setting.
5. When the user logs on, steps 2, 3, and 4 are repeated for user settings. The client obtains an ordered list of GPOs scoped to the user, examines each GPO synchronously, and hands over GPOs that should be applied to the appropriate CSEs for processing. This step is modified if User Loopback Group Policy Processing is enabled. Loopback policy processing is discussed in the next section.

NOTE Policy settings in both the Computer Configuration and User Configuration nodes

Most policy settings are specific to either the User Configuration or Computer Configuration node. A small handful of settings appear in both nodes. Although in most situations the setting in the Computer Configuration node will override the setting in the User Configuration node, it is important to read the explanatory text accompanying the policy setting to understand the setting's effect and its application.

-
6. Every 90–120 minutes after computer startup, computer policy refresh occurs, and steps 2, 3, and 4 are repeated for computer settings.
 7. Every 90–120 minutes after user logon, user policy refresh occurs, and steps 2, 3, and 4 are repeated for user settings.

NOTE Settings might not take effect immediately

Although most settings are applied during a background policy refresh, some CSEs do not apply the setting until the next startup or logon event. Newly added startup and logon script policies, for example, will not run until the next computer startup or logon. Software installation, discussed in Chapter 7, will occur at the next startup if the software is assigned in computer settings. Changes to folder redirection policies will not take effect until the next logon.

Loopback Policy Processing

By default, a user's settings come from GPOs scoped to the user object in Active Directory. Regardless of which computer the user logs on to, the resultant set of policies that determine the user's environment will be the same. There are situations, however, when you might want to configure a user differently, depending on the computer in use. For example, you might want to lock down and standardize user desktops when users log on to computers in closely managed environments such as conference rooms, reception areas, laboratories, classrooms,

and kiosks. Imagine a scenario in which you want to enforce a standard corporate appearance for the Windows desktop on all computers in conference rooms and other public areas of your office. How could you centrally manage this configuration, using Group Policy? Policy settings that configure desktop appearance are located in the User Configuration node of a GPO. Therefore, by default, the settings apply to users, regardless of which computer they log on to. The default policy processing does not give you a way to scope user settings to apply to computers, regardless of which user logs on. That's where loopback policy processing comes in.

Loopback policy processing alters the default algorithm used by the Group Policy client to obtain the ordered list of GPOs that should be applied to a user's configuration. Instead of user configuration being determined by the User Configuration node of GPOs that are scoped to the user object, user configuration can be determined by the User Configuration node policies of GPOs that are scoped to the *computer* object.

The User Group Policy Loopback Processing Mode policy, located in the Computer Configuration\Policies\Administrative Templates\System\Group Policy folder in Group Policy Management Editor, can be, like all policy settings, set to Not Configured, Enabled, or Disabled. When enabled, the policy can specify Replace or Merge mode.

- **Replace** In this case, the GPO list for the user (obtained in step 5 in the "Group Policy Processing" section) is replaced in its entirety by the GPO list already obtained for the computer at computer startup (during step 2). The settings in the User Configuration policies of the computer's GPOs are applied to the user. Replace mode is useful in a situation such as a classroom, where users should receive a *standard configuration* rather than the configuration applied to those users in a less managed environment.
- **Merge** In this case, the GPO list obtained for the computer at computer startup (step 2 in the "Group Policy Processing" section) is appended to the GPO list obtained for the user when logging on (step 5). Because the GPO list obtained for the computer is applied later, settings in GPOs on the computer's list have precedence if they conflict with settings in the user's list. This mode would be useful to apply *additional settings* to users' typical configurations. For example, you might allow a user to receive his or her typical configuration when logging on to a computer in a conference room or reception area but replace the wallpaper with a standard bitmap and disable the use of certain applications or devices.

Exam Tip The 70-640 exam is likely to include several questions that test your knowledge of Group Policy scope. Sometimes, questions that seem to be addressing the technical details of a policy setting are, in fact, testing your ability to scope the setting to appropriate systems. When you encounter Group Policy questions, ask yourself, "Is this really about a specific policy setting, or is it about the scope of that setting?"

Lesson 3: Supporting Group Policy

Group Policy application can be complex to analyze and understand, with the interaction of multiple settings in multiple GPOs scoped using a variety of methods. You must be equipped to effectively evaluate and troubleshoot your Group Policy implementation, to identify potential problems before they arise, and to solve unforeseen challenges. Microsoft Windows provides two tools that are indispensable for supporting Group Policy: Resultant Set of Policy (RSOP) and the Group Policy Operational Logs. In this lesson, you will explore the use of these tools in both proactive and reactive troubleshooting and support scenarios.

After this lesson, you will be able to:

- Analyze the set of GPOs and policy settings that have been applied to a user or computer
- Proactively model the impact of Group Policy or Active Directory changes on resultant set of policy
- Locate the event logs containing Group-Policy related events

Estimated lesson time: 30 minutes

Resultant Set of Policy

In Lesson 2, you learned that a user or computer can be within the scope of multiple GPOs. Group Policy inheritance, filters, and exceptions are complex, and it's often difficult to determine just which policy settings will apply. *Resultant Set of Policy (RSoP)* is the net effect of GPOs applied to a user or computer, taking into account GPO links, exceptions such as Enforced and Block Inheritance, and the application of security and WMI filters. RSoP is also a collection of tools that help you evaluate, model, and troubleshoot the application Group Policy settings. RSoP can query a local or remote computer and report back the exact settings that were applied to the computer and to any user who has logged on to the computer. RSoP can also model the policy settings that are anticipated to be applied to a user or computer under a variety of scenarios, including moving the object between OUs or sites or changing the object's group membership. With these capabilities, RSoP can help you manage and troubleshoot conflicting policies.

Windows Server 2008 provides the following tools for performing RSoP analysis:

- The Group Policy Results Wizard
- The Group Policy Modeling Wizard
- *Gpresult.exe*

Generating RSoP Reports with the Group Policy Results Wizard

To help you analyze the cumulative effect of GPOs and policy settings on a user or computer in your organization, the Group Policy Management console includes the Group Policy Results Wizard. If you want to understand exactly which policy settings have applied to a user or computer and why, the Group Policy Results Wizard is the tool to use.

The Group Policy Results Wizard is able to reach into the WMI provider on a local or remote computer running Windows Vista, Windows XP, Windows Server 2003, and Windows Server 2008. The WMI provider can report everything there is to know about the way Group Policy was applied to the system. It knows when processing occurred, which GPOs were applied, which GPOs were not applied and why, errors that were encountered, the exact policy settings that took precedence, and their source GPO.

There are several requirements for running the Group Policy Results Wizard:

- You must have administrative credentials on the target computer.
- The target computer must be running Windows XP or later. The Group Policy Results Wizard cannot access Windows 2000 systems.
- You must be able to access WMI on the target computer. That means that it must be powered on, connected to the network, and accessible through ports 135 and 445.

NOTE Enable remote administration of client computers

Performing RSoP analysis by using Group Policy Results Wizard is just one example of remote administration. Windows XP SP2, Windows Vista, and Windows Server 2008 include a firewall that prevents unsolicited inbound connections even from members of the Administrators group. Group Policy provides a simple way to enable remote administration. In the Computer Configuration\Policies\Administrative Templates\Network\Network Connections\Windows Firewall\Domain Profile folder, you will find a policy setting named Windows Firewall: Allow Inbound Remote Administration Exception. When you enable this policy setting, you can specify the IP addresses or subnets from which inbound remote administration packets will be accepted. As with all policy settings, review the explanatory text on the Explain tab and test the effect of the policy in a lab environment before deploying it in production.

- The WMI service must be started on the target computer.
- If you want to analyze RSoP for a user, that user must have logged on at least once to the computer. It is not necessary for the user to be currently logged on.

After you have ensured that the requirements are met, you are ready to run an RSoP analysis. Right-click Group Policy Results in the GPMC and choose Group Policy Results Wizard. The wizard prompts you to select a computer. It then connects to the WMI provider on that computer and provides a list of users that have logged on to it. You can then select one of the users or opt to skip RSoP analysis for user configuration policies.

The wizard produces a detailed RSoP report in a dynamic HTML format. If Internet Explorer ESC is enabled, you will be prompted to allow the console to display the dynamic content. Each section of the report can be expanded or collapsed by clicking the Show or Hide link or by double-clicking the heading of the section. The report is displayed on three tabs:

- **Summary** The Summary tab displays the status of Group Policy processing at the last refresh. You can identify information that was collected about the system, the GPOs that were applied and denied, security group membership that might have affected GPOs filtered with security groups, WMI filters that were analyzed, and the status of CSEs.
- **Settings** The Settings tab displays the resultant set of policy settings applied to the computer or user. This tab shows you exactly what has happened to the user through the effects of your Group Policy implementation. A tremendous amount of information can be gleaned from the Settings tab, but some data isn't reported, such as IPSec, wireless, and disk quota policy settings.
- **Policy Events** The Policy Events tab displays Group Policy events from the event logs of the target computer.

After you have generated an RSoP report with the Group Policy Results Wizard, you can right-click the report to rerun the query, print the report, or save the report as either an XML file or an HTML file that maintains the dynamic expanding and collapsing sections. Either file type can be opened with Internet Explorer, so the RSoP report is portable outside the GPMC. If you right-click the node of the report itself, underneath the Group Policy Results folder in the console tree, you can switch to Advanced View. In Advanced View, RSoP is displayed using the RSoP snap-in, which exposes all applied settings, including IPSec, wireless, and disk quota policies.

Generating RSoP Reports with *Gpresult.exe*

The *Gpresult.exe* command is the command-line version of the Group Policy Results Wizard. *Gpresult* taps into the same WMI provider as the wizard, produces the same information, and, in fact, enables you to create the same graphical reports. *Gpresult* runs on Windows Vista, Windows XP, Windows Server 2003, and Windows Server 2008. Windows 2000 includes a *Gpresult.exe* command, which produces a limited report of Group Policy processing but is not as sophisticated as the command included in later versions of Windows.

When you run the *Gpresult* command, you are likely to use the following options:

- **/s *computername*** Specifies the name or IP address of a remote system. If you use a dot (.) as the computer name, or do not include the */s* option, the RSoP analysis is performed on the local computer.
- **/scope [*user* | *computer*]** Displays RSoP analysis for user or computer settings. If you omit the */scope* option, RSoP analysis includes both user and computer settings.
- **/user *username*** Specifies the name of the user for which RSoP data is to be displayed.
- **/r** Displays a summary of RSoP data.

- **/v** Displays verbose RSoP data that presents the most meaningful information.
- **/z** Displays super verbose data, including the details of all policy settings applied to the system. Often, this is more information than you will require for typical Group Policy troubleshooting.
- **/u domain\user /p password** Provides credentials that are in the Administrators group of a remote system. Without these credentials, *Gpresult* runs using the credentials with which you are logged on.
- **[/x | /h] filename** Saves the reports in XML or HTML format, respectively. These options are available in Windows Vista SP1 and Windows Server 2008.

Quick Check

- You want to perform RSoP analysis on a remote system. Which two tools can you use?

Quick Check Answer

- The Group Policy Results Wizard and *Gpupdate.exe* can be used to perform your top analysis on a remote system.

Troubleshooting Group Policy with the Group Policy Results Wizard and *Gpresult.exe*

As an administrator, you will likely encounter scenarios that require Group Policy troubleshooting. You might need to diagnose and solve problems, including:

- GPOs are not applied at all.
- The resultant set of policies for a computer or user are not those that were expected.

The Group Policy Results Wizard and *Gpresult.exe* will often provide the most valuable insight into Group Policy processing and application problems. Remember that these tools examine the WMI RSoP provider to report exactly what happened on a system. Examining the RSoP report will often point you to GPOs that are scoped incorrectly or policy processing errors that prevented the application of GPOs settings.

Performing What-If Analyses with the Group Policy Modeling Wizard

If you move a computer or user between sites, domains, or OUs, or change its security group membership, the GPOs scoped to that user or computer will change and, therefore, the RSoP for the computer or user will be different. RSoP will also change if slow link or loopback processing occurs or if there is a change to a system characteristic that is targeted by a WMI filter.

Before you make any of these changes, you should evaluate the potential impact to the RSoP of the user or computer. The Group Policy Results Wizard can perform RSoP analysis only on

what has actually happened. To predict the future and to perform what-if analyses, you can use the Group Policy Modeling Wizard.

Right-click the Group Policy Modeling node in the GPMC. Choose Group Policy Modeling Wizard and perform the steps in the wizard. Modeling is performed by conducting a simulation on a domain controller, so you are first asked to select a domain controller that is running Windows Server 2003 or later. You do not need to be logged on locally to the domain controller, but the modeling request will be performed on the domain controller. You are then asked to specify the settings for the simulation:

- Select a user or computer object to evaluate or specify the OU, site, or domain to evaluate.
- Choose whether slow link processing should be simulated.
- Specify to simulate loopback processing and, if so, choose Replace or Merge mode.
- Select a site to simulate.
- Select security groups for the user and for the computer.
- Choose which WMI filters to apply in the simulation of user and computer policy processing.

When you have specified the settings for the simulation, a report is produced that is very similar to the Group Policy Results report discussed earlier. The Summary tab shows an overview of which GPOs will be processed, and the Settings tab details the policy settings that will be applied to the user or computer. This report, too, can be saved by right-clicking it and choosing Save Report.

Examining Policy Event Logs

Windows Vista and Windows Server 2008 improve your ability to troubleshoot Group Policy not only with RSoP tools but also with improved logging of Group Policy events. In the System log, you will find high-level information about Group Policy, including errors created by the Group Policy client when it cannot connect to a domain controller or locate GPOs. The Application log captures events recorded by CSEs. A new log, called the Group Policy Operational Log, provides detailed information about Group Policy processing. To find these logs, open the Event Viewer snap-in or console. The System and Application logs are in the Windows Logs node. The Group Policy Operational Log is found in Applications And Services Logs\Microsoft\Windows\GroupPolicy\Operational. This log will not be available until after you use the Group Policy Modeling Wizard initially.

Chapter 7

Group Policy Settings

Group Policy can be used to manage the configuration of an enormous variety of components and features of Microsoft Windows. In the previous chapter, you learned how to configure a Group Policy infrastructure. In this chapter, you will learn to apply that infrastructure to manage several types of configuration related to security and software installation. You will also discover tools, such as the Security Configuration Wizard, that make it easier to determine which settings should be configured based on a server's roles. Finally, you will learn how to configure auditing of files and folders and of Active Directory Domain Services (AD DS) changes.

Exam objectives in this chapter:

- Creating and Maintaining Active Directory Objects
 - Create and apply Group Policy objects (GPOs).
 - Configure GPO templates.
 - Configure audit policy by using GPOs.

Before You Begin

To complete the practices in this chapter, you must have created a domain controller named SERVER01 in a domain named *contoso.com*. See Chapter 1, "Installation," for detailed steps to perform this task.

Real World

Dan Holme

I am often brought in by clients to perform “sanity checks” on their Active Directory implementations. These sanity checks involve an examination of Group Policy settings and a discussion of how to take better advantage of Group Policy to manage change and configuration. It amazes me that a full eight years after the introduction of Group Policy, many organizations do not yet use its full capability, particularly in the area of security. Three of the four lessons in this chapter focus on the interaction between security configuration and Group Policy. Configuration such as the membership of the Administrators group and assignment of user rights, service startup modes, and audit policies can be effectively managed with Group Policy. What you will learn in this chapter will not only help you pass the 70-640 exam; it will also help you increase the manageability and security of your entire enterprise. This includes Active Directory itself. For the past eight years, I’ve constantly been asked, “How can I know what changes have been made by administrators in Active Directory?” Now, thanks to the new Directory Service Changes auditing in Windows Server 2008, you can simply check your security log. Even if you are already using policy to manage your security configuration, this new feature, along with the vastly improved Security Configuration Wizard, will surely take your security management capabilities to a higher level.

Lesson 1: Delegating the Support of Computers

Many enterprises have one or more members of personnel dedicated to supporting end users, a role often referred to as the *help desk*, *desktop support*, or just *support*. Help desk personnel are often asked to perform troubleshooting, configuration, or other support tasks on client computers, and these tasks often require administrative privileges. Therefore, the credentials used by support personnel must be at the level of a member of the local Administrators group on client computers, but desktop support personnel do not need the high level of privilege given to the Domain Admins group, so it is not recommended to place them in that group. Instead, configure client systems so that a group representing support personnel is added to the local Administrators group. Restricted groups policies enable you to do just that, and in this lesson, you will learn how to use restricted groups policies to add the help desk personnel to the local Administrators group of clients and, thereby, to delegate support of those computers to the help desk. The same approach can be used to delegate the administration of any scope of computers to the team responsible for those systems.

After this lesson, you will be able to:

- Delegate the administration of computers.
- Use Group Policy to modify or enforce the membership of groups.

Estimated lesson time: 30 minutes

Understanding Restricted Groups Policies

When you edit a Group Policy object (GPO) and expand the Computer Configuration node, the Policies node, the Windows Settings node, and the Security Settings node, you will find the Restricted Groups policy node, shown in Figure 7-1.

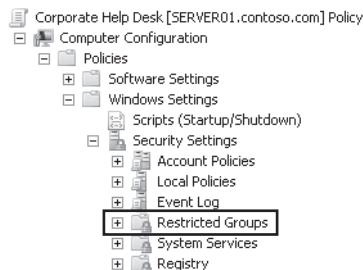


Figure 7-1 The Restricted Groups policy node of a Group Policy object

Restricted groups policy settings enable you to manage the membership of groups. There are two types of settings: This Group Is A Member Of (the Member Of setting) and Members Of This Group (the Members setting). Figure 7-2 shows examples.

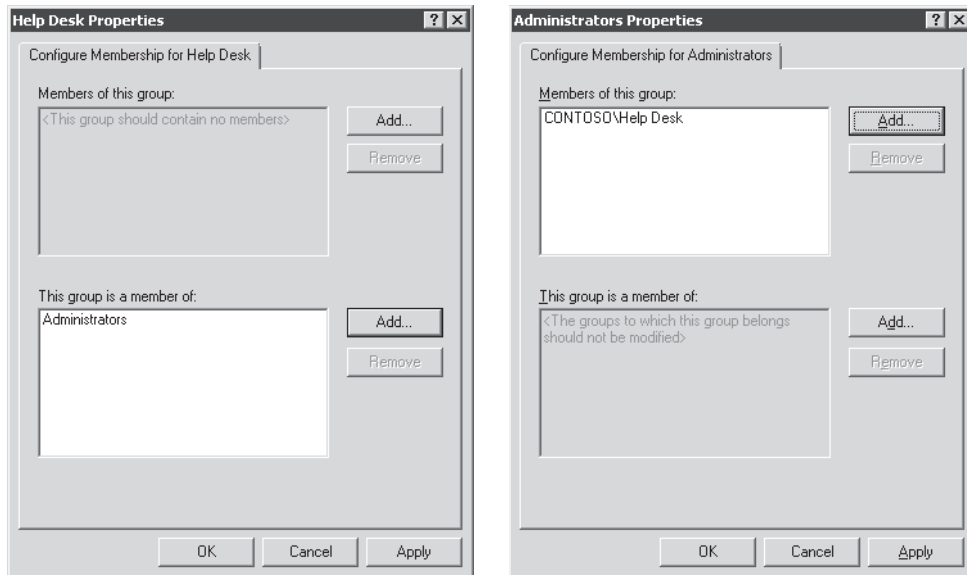


Figure 7-2 Member Of and Members restricted groups policies

It's very important to understand the difference between these two settings. A Member Of setting specifies that the group specified by the policy is a member of another group. On the left side of Figure 7-2, you can see a typical example: The `CONTOSO\Help Desk` group is a member of the Administrators group. When a computer applies this policy setting, it ensures that the Help Desk group from the domain becomes a member of its local Administrators group. If there is more than one GPO with restricted groups policies, each Member Of policy is applied. For example, if a GPO linked to the Clients organizational unit (OU) specifies `CONTOSO\Help Desk` as a member of Administrators, and a second GPO linked to the NYC OU (a sub-OU of the Clients OU) specifies `CONTOSO\NYC Support` as a member of Administrators, a computer in the NYC OU will add both the Help Desk and NYC Support groups to its Administrators group in addition to any existing members of the group such as Domain Admins. This example is illustrated in Figure 7-3. As you can see, restricted groups policies that use the Member Of setting are cumulative.

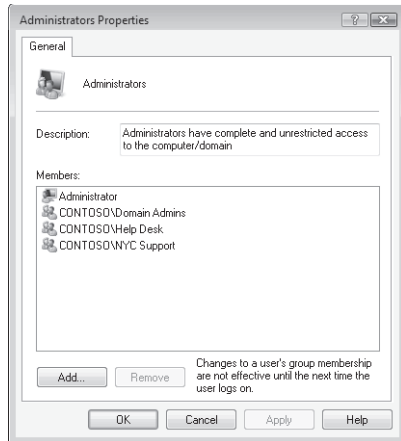


Figure 7-3 Results of restricted groups policies using the Member Of setting

The second type of restricted groups policy setting is the Members setting, which specifies the entire membership of the group specified by the policy. The right side of Figure 7-2 shows a typical example: the Administrators group's Members list is specified as `CONTOSO\Help Desk`. When a computer applies this policy setting, it ensures that the local Administrators group's membership consists *only* of `CONTOSO\Help Desk`. Any members not specified in the policy are removed, including Domain Admins. The Members setting is the authoritative policy—it defines the final list of members. If there is more than one GPO with restricted group policies, the GPO with the highest priority will prevail. For example, if a GPO linked to the Clients OU specifies the Administrators group membership as `CONTOSO\Help Desk`, and another GPO linked to the NYC OU specifies the Administrators group membership as `CONTOSO\NYC Support`, computers in the NYC OU will have only the NYC Support group in their Administrators group. This example is illustrated in Figure 7-4.

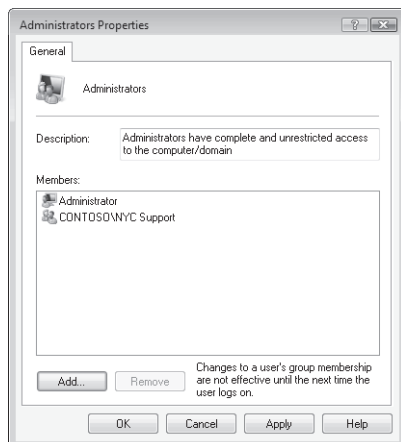


Figure 7-4 Restricted groups policies using the Members setting

In your enterprise, be careful to design and test your restricted groups policies to ensure that they achieve the desired result. Do not mix GPOs that use the Member Of and the Members settings—use one approach or the other.

Exam Tip On the 70-640 exam, be able to identify the differences between restricted groups policies that use the Member Of setting and those that use the Members setting. Remember that Member Of settings are cumulative and that if GPOs use the Members setting, only the Members setting with the highest GPO processing priority will be applied, and its list of members will prevail.

Delegating Administration Using Restricted Groups Policies with the Member Of Setting

You can use restricted groups policies with the Member Of setting to manage the delegation of administrative privileges for computers by following these steps:

1. In Group Policy Management Editor, navigate to Computer Configuration\Policies\Windows Settings\Security Settings\Restricted Groups.
2. Right-click Restricted Groups and choose Add Group.
3. Click the Browse button and, in the Select Groups dialog box, type the name of the group you want to add to the Administrators group, for example, **CONTOSO\Help Desk**, and click OK.
4. Click OK to close the Add Group dialog box.
A Properties dialog box appears.
5. Click the Add button next to the This Group Is A Member Of section.
6. Type **Administrators** and click OK.
The Properties group policy setting should look something like the left side of Figure 7-2.
7. Click OK again to close the Properties dialog box.

Delegating the membership of the local Administrators group in this manner adds the group specified in step 3 to that group. It does not remove any existing members of the Administrators group. The group policy simply tells the client, “Make sure this group is a member of the local Administrators group.” This allows for the possibility that individual systems could have other users or groups in their local Administrators group. This group policy setting is also cumulative. If multiple GPOs configure different security principals as members of the local Administrators group, all will be added to the group.

To take complete control of the local Administrators group, follow these steps:

1. In Group Policy Management Editor, navigate to Computer Configuration\Windows Settings\Security Settings\Restricted Groups.
2. Right-click Restricted Groups and choose Add Group.

3. Type **Administrators** and click OK.
A Properties dialog box appears.
4. Click the Add button next to the Members Of This Group section.
5. Click the Browse button and type the name of the group you want to make the sole member of the Administrators group—for example, **CONTOSO\Help Desk**—and click OK.
6. Click OK again to close the Add Member dialog box.
The group policy setting Properties should look something like the right side of Figure 7-2.
7. Click OK again to close the Properties dialog box.

When you use the Members setting of a restricted groups policy, the Members list defines the final membership of the specified group. The steps just listed result in a GPO that authoritatively manages the Administrators group. When a computer applies this GPO, it will add all members specified by the GPO and will remove all members not specified by the GPO, including Domain Admins. Only the local Administrator account will not be removed from the Administrators group because Administrator is a permanent and nonremovable member of Administrators.

Quick Check

- You want to add a group to the local Administrators group on computers without removing accounts that already exist in the group. Describe the restricted groups policy you should create.

Quick Check Answer

- Create a restricted groups policy for the group you wish to add. Use the Member Of policy setting (This Group Is A Member Of) and specify Administrators.

Lesson 2: Managing Security Settings

Security is a primary concern for all Windows administrators. Windows Server 2008 includes numerous settings that affect the services that are running, the ports that are open, the network packets that are allowed into or out of the system, the rights and permissions of users, and the activities that are audited. There is an enormous number of settings that can be managed, and unfortunately, there is no magic formula that applies the perfect security configuration to a server. The appropriate security configuration for a server depends on the roles that server plays, the mix of operating systems in the environment, and the security policies of the organization, which themselves depend on compliance regulations enforced from outside the organization.

Therefore, you must work to determine and configure the security settings that are required for servers in your organization, and you must be prepared to manage those settings in a way that centralizes and optimizes security configuration. Windows Server 2008 provides several mechanisms with which to configure security settings on one or more systems. In this lesson, you will discover these mechanisms and their interactions.

After this lesson, you will be able to:

- Configure security settings on a computer using the Local Security Policy.
- Create and apply security templates to manage security configuration.
- Analyze security configuration based on security templates.
- Create, edit, and apply security policies using the Security Configuration Wizard.
- Deploy security configuration with Group Policy.

Estimated lesson time: 60 minutes

Configuring the Local Security Policy

Each server running Windows Server 2008 maintains a collection of security settings that can be managed using the local GPO. You can configure the local GPO by using the Group Policy Object Editor snap-in or the Local Security Policy console. The available policy setting categories are shown in Figure 7-5.

This lesson focuses on the mechanisms with which to configure and manage security settings rather than on the details of the settings themselves. Many of the settings—including account policies, audit policy, and user rights assignment—are discussed elsewhere in this training kit.

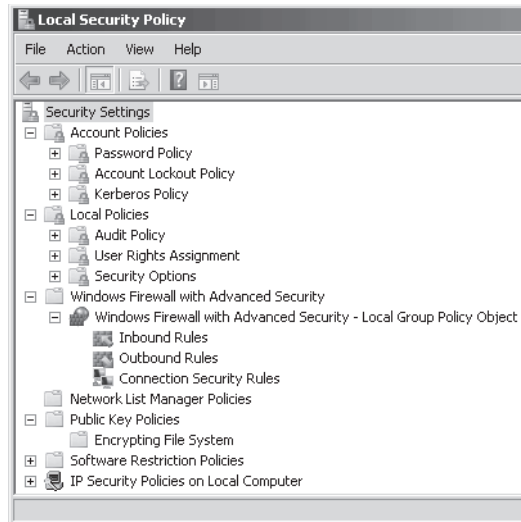


Figure 7-5 The security settings available in the local GPO

Because domain controllers (DCs) do not have local user accounts—only domain accounts—the policies in the Account Policies container of the local GPO on DCs cannot be configured. Instead, account policies for the domain should be configured as part of a domain-linked GPO such as the Default Domain Policy GPO. Account policies are discussed in the first lesson of Chapter 8, “Authentication.”

The settings found in the local Security Settings policies are a subset of the policies that can be configured using domain-based Group Policy, shown in Figure 7-6. As you learned in Chapter 6, “Group Policy Infrastructure,” it is a best practice to manage configuration by using domain-based Group Policy rather than on a machine-by-machine basis using local Group Policy. This is particularly true for domain controllers. The Default Domain Controllers Policy GPO is created when the first domain controller is promoted for a new domain. It is linked to the Domain Controllers OU and should be used to manage baseline security settings for all DCs in the domain so that DCs are consistently configured.

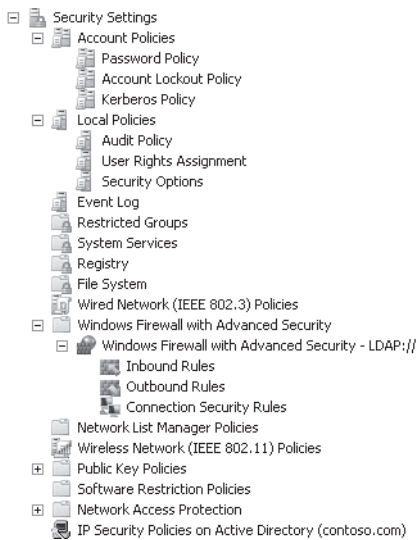


Figure 7-6 Security settings in a domain-based GPO

Managing Security Configuration with Security Templates

The second mechanism for managing security configuration is the security template. A security template is a collection of configuration settings stored as a text file with the .inf extension. As you can see in Figure 7-7, a security template contains settings that are a subset of the settings available in a domain-based GPO but a somewhat different subset than those managed by the local GPO. The tools used to manage security templates present settings in an interface that enables you to save your security configurations as files and deploy them when and where they are needed. You can also use a security template to analyze the compliance of a computer's current configuration against the desired configuration.

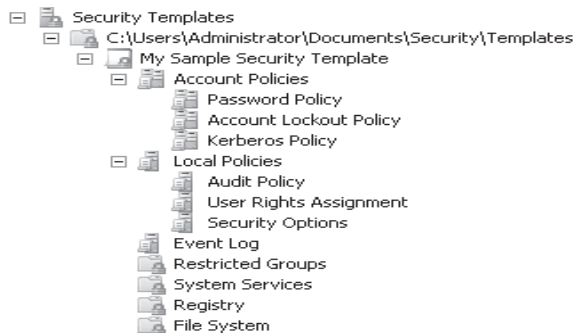


Figure 7-7 Security settings in a security template

There are several advantages to storing security configuration in security templates. For example, because the templates are plaintext files, you can work with them manually as with any text file, cutting and pasting sections as needed. Further, templates make it easy to store security configurations of various types so that you can easily apply different levels of security to computers performing different roles.

Security templates enable you to configure any of the following types of policies and settings:

- **Account Policies** Enables you to specify password restrictions, account lockout policies, and Kerberos policies
- **Local Policies** Enables you to configure audit policies, user rights assignments, and security options policies
- **Event Log Policies** Enables you to configure maximum event log sizes and rollover policies
- **Restricted Groups** Enables you to specify the users who are permitted to be members of specific groups
- **System Services** Enables you to specify the startup types and permissions for system services
- **Registry Permissions** Enables you to set access control permissions for specific registry keys
- **File System Permissions** Enables you to specify access control permissions for NTFS files and folders

You can deploy security templates in a variety of ways, using Active Directory Group Policy Objects, the Security Configuration And Analysis snap-in, or *Secedit.exe*. When you associate a security template with an Active Directory object, the settings in the template become part of the GPO associated with the object. You can also apply a security template directly to a computer, in which case, the settings in the template become part of the computer's local policies. You will learn about each of these options in this section.

Using the Security Templates Snap-in

To work with security templates, you use the Security Templates snap-in. Windows Server 2008 does not include a console with the Security Templates snap-in, so you have to create one yourself using the MMC *Add/Remove Snap-in* command. The snap-in creates a folder called Security and a subfolder called Templates in your Documents folder, and the Documents \Security\Templates folder becomes the template search path, where you can store one or more security templates.

You can create a new security template by right-clicking the node that represents your template search path—C:\Users\Administrator\Documents\Security\Templates, for example—and choosing New Template. You can also create a template that reflects the current configuration of a server; you'll learn how to do that in the "Creating a Security Template" section.

Settings are configured in the template in the same way that settings are configured in a GPO. The Security Templates snap-in is used to configure settings in a security template. It is just an editor—it does not play any role in actually applying those settings to a system. Configure security settings in a template by using the Security Templates snap-in. Although the template itself is a text file, the syntax can be confusing. Using the snap-in ensures that settings are changed using the proper syntax. The exception to this rule is adding Registry settings that are not already listed in the Local Policies\Security Option portion of the template. As new security settings become known, if they can be configured using a Registry key, you can add them to a security template. To do so, you add them to the Registry Values section of the template.

MORE INFO Adding custom registry settings

The article “How to Add Custom Registry Settings to Security Configuration Editor” helps you understand how to perform this task. You can find it at <http://support.microsoft.com/?kbid=214752>.

NOTE Save your settings

Be sure to save your changes to a security template by right-clicking the template and choosing Save.

When you install a server or promote it to a domain controller, a default security template is applied by Windows. You can find that template in the %SystemRoot%\Security\Templates folder. On a domain controller, the template is called DC security.inf. You should not modify this template directly, but you can copy it to your template search path and modify the copy.

NOTE Security templates in Windows Server 2008 and in earlier versions of Windows

In previous versions of Windows, a number of security templates were available to modify and apply to a computer. The new role-based configuration of Windows Server 2008 and the improved Security Configuration Manager have made these templates unnecessary.

Deploying Security Templates by Using Group Policy Objects

Creating and modifying security templates does not improve security unless you apply those templates. To configure a number of computers in a single operation, you can import a security template into the Group Policy Object for a domain, site, or organizational unit object in Active Directory. To import a security template into a GPO, right-click the Security Settings node and choose Import Policy. In the Import Policy From dialog box, if you select the Clear This Database Before Importing check box, all security settings in the GPO will be erased prior to importing the template settings, so the GPO’s security settings will match the template’s settings. If you leave the Clear This Database Before Importing check box deselected, the GPO’s security policy settings will remain and the templates settings will be imported. Any settings defined in the GPO that are also defined in the template will be replaced with the template’s setting.

Security Configuration and Analysis Tool

You can use the Security Configuration and Analysis snap-in to apply a security template to a computer interactively. The snap-in also provides the ability to analyze the current system security configuration and compare it to a baseline saved as a security template. This enables you to determine quickly whether someone has changed a computer's security settings and whether the system conforms to your organization's security policies.

As with the Security Templates snap-in, Windows Server 2008 does not include a console with the Security Configuration and Analysis snap-in, so you must add the snap-in to a console yourself.

To use the Security Configuration and Analysis snap-in, you must first create a database that will contain a collection of security settings. The database is the interface between the actual security settings on the computer and the settings stored in your security templates. Create a database (or open an existing one) by right-clicking the Security Configuration And Analysis node in the console tree.

You can then import one or more security templates. If you import more than one template, you must decide whether to clear the database. If the database is cleared, only the settings in the new template will be part of the database. If the database is not cleared, additional template settings that are defined will override settings from previously imported templates. If settings in newly imported templates are not defined, the settings in the database from previously imported templates will remain. To summarize, the Security Configuration and Analysis snap-in creates a database of security settings composed of imported security template settings. The settings in the database can be applied to the computer or used to analyze the computer's compliance and discrepancies with the desired state.

IMPORTANT Database settings vs. the computer's settings

Remember that settings in a database do not modify the computer's settings or the settings in a template until that database is either used to configure the computer or exported to a template.

Applying Security Templates to a Computer

After you have imported one or more templates to create the database, you can apply the database settings to the computer. Right-click Security Configuration And Analysis and choose Configure Computer Now. You will be prompted for a path to an error log that will be generated during the application of settings. After applying the settings, examine the error log for any problems.

Quick Check

- Describe the procedure used to apply a security template to a computer.

Quick Check Answer

- Use the Security Configuration and Analysis snap-in to create a database. Import the template into the database. Configure the computer by using the database.

Analyzing the Security Configuration of a Computer

Before applying the database settings to a computer, you might want to analyze the computer's current configuration to identify discrepancies. Right-click Security Configuration And Analysis and choose Analyze Computer Now. The system prompts you for the location of its error log file and then proceeds to compare the computer's current settings to the settings in the database. After the analysis is complete, the console produces a report such as the one shown in Figure 7-8.

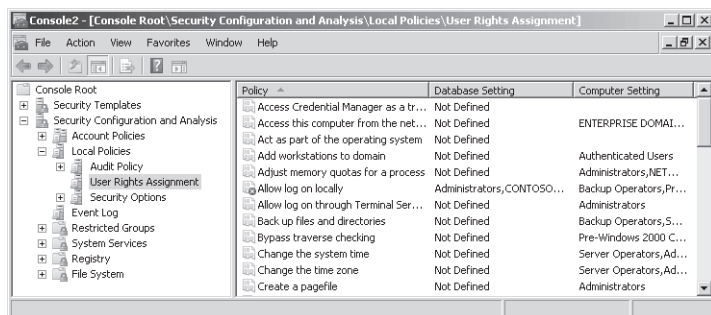


Figure 7-8 The Security Configuration and Analysis snap-in displays an analysis of the computer's configuration.

Unlike the display of policy settings in the Group Policy Management Editor, Group Policy Object Editor, Local Security Policy, or Security Templates snap-ins, the report shows for each policy the setting defined in the database (which was derived from the templates you imported) and the computer's current setting. The two settings are compared, and the comparison result is displayed as a flag on the policy name. For example, in Figure 7-8, the Allow Log On Locally policy setting is showing a discrepancy between the database setting and the computer setting. The meanings of the flags are as follows:

- **X in a red circle** Indicates that the policy is defined both in the database and on the computer but that the configured values do not match
- **Green check mark in a white circle** Indicates that the policy is defined both in the database and on the computer and that the configured values do match

- **Question mark in a white circle** Indicates that the policy is not defined in the database and, therefore, was not analyzed or that the user running the analysis did not have the permissions needed to access the policy on the computer
- **Exclamation point in a white circle** Indicates that the policy is defined in the database but does not exist on the computer
- **No flag** Indicates that the policy is not defined in the database or on the computer

Correcting Security Setting Discrepancies

As you examine the elements of the database and compare its settings with those of the computer, you might find discrepancies and want to make changes to the computer's configuration or to the database to bring the two settings into alignment. You can double-click any policy setting to display its Properties dialog box and modify its value in the database. After you've made changes to the database, you can apply the database settings to the computer by performing the steps described earlier, in the section, "Applying Security Templates to a Computer."

CAUTION Applying or exporting database changes

Modifying a policy value in the Security Configuration and Analysis snap-in changes the database value only, not the actual computer setting. For the changes you make to take effect on the computer, you must either apply the database settings to the computer using the *Configure Computer Now* command or export the database to a new template and apply it to the computer, using a GPO or the *Secedit.exe* command (discussed in the "Secedit.exe" section).

Alternatively, you can modify the computer's security settings directly by using the Local Security Policy console, by modifying the appropriate Group Policy Object, or by manually manipulating file system or registry permissions. After making such changes, return to the Security Configuration And Analysis snap-in and choose the *Analyze Computer Now* command to refresh the analysis of the computer's settings compared to the database.

Creating a Security Template

You can create a new security template from the database by right-clicking Security Configuration And Analysis and selecting Export Template. The template will contain the settings in the database, which have been imported from one or more security templates and which you have modified to reflect the current settings of the analyzed computer.

IMPORTANT Exporting the database to a template

The Export Template feature creates a new template from the current database settings at the time you execute the command, not from the computer's current settings.

Secedit.exe

Secedit.exe is a command-line utility that can perform the same functions as the Security Configuration and Analysis snap-in. The advantage of *Secedit.exe* is that you can call it from scripts and batch files, enabling you to automate your security template deployments. Another big advantage of *Secedit.exe* is that you can use it to apply only part of a security template to a computer, something you cannot do with the Security Configuration and Analysis snap-in or with Group Policy Objects. For example, if you want to apply the file system's permissions from a template but leave all the other settings alone, *Secedit.exe* is the only way to do it.

To use *Secedit.exe*, you run the program from the command prompt with one of the following six main parameters, plus additional parameters for each function:

- **Configure** Applies all or part of a security database to the local computer. You can also configure the program to import a security template into the specified database before applying the database settings to the computer.
- **Analyze** Compares the computer's current security settings with those in a security database. You can configure the program to import a security template into the database before performing the analysis. The program stores the results of the analysis in the database itself, which you can view later, using the Security Configuration and Analysis snap-in.
- **Import** Imports all or part of a security template into a specific security database.
- **Export** Exports all or part of the settings from a security database to a new security template.
- **Validate** Verifies that a security template is using the correct internal syntax.
- **Generaterollback** Creates a security template you can use to restore a system to its original configuration after applying another template.

For example, to configure the machine by using a template called *BaselineSecurity*, use the following command:

```
secedit /configure /db BaselineSecurity.sdb  
/cfg BaselineSecurity.inf /log BaselineSecurity.log
```

To create a rollback template for the *BaselineSecurity* template, use the following command:

```
secedit /generaterollback /cfg BaselineSecurity.inf  
/rbk BaselineSecurityRollback.inf  
/log BaselineSecurityRollback.log
```

MORE INFO *Secedit.exe*

For full details regarding *Secedit.exe* and its switches, see <http://technet2.microsoft.com/windowsserver/en/library/b1007de8-a11a-4d88-9370-25e2445605871033.msp?mfr=true>.

The Security Configuration Wizard

The Security Configuration Wizard can be used to enhance the security of a server by closing ports and disabling services not required for the server's roles. The Security Configuration Wizard can be launched from the home page of Server Manager, in the Security Information section, or from the Administrative Tools folder. There is also a command-line version of the tool, *scwcmd.exe*. Type **scwcmd.exe /?** at the command prompt for help on the command or see <http://technet2.microsoft.com/windowsserver2008/en/library/a222cb38-db08-4bf1-b9cf-6ec566c239e91033.aspx?mfr=true>.

The Security Configuration Wizard is a next-generation security management tool. It is more advanced than the Security Configuration and Analysis snap-in and role-based in accordance with the new role-based configuration of Windows Server 2008. The Security Configuration Wizard creates a security policy—an .xml file—that configures services, network security including firewall rules, registry values, audit policy, and other settings based on the roles of a server. That security policy can then be modified, applied to another server, or transformed into a GPO for deployment to multiple systems.

Creating a Security Policy

To create a security policy, you launch the Security Configuration Wizard from the Administrative Tools folder or the Security Information section on the home page of Server Manager. You can open the Security Configuration Wizard Help file by clicking the Security Configuration Wizard link on the first page of the wizard. Click Next and choose Create A New Security Policy. Click Next and enter the name of the server to scan and analyze. The security policy will be based on the roles being performed by the specified server. You must be an administrator on the server for the analysis of its roles to proceed. Ensure also that all applications using inbound IP ports are running prior to running the Security Configuration Wizard.

When you click Next, the Security Configuration Wizard begins the analysis of the selected server's roles. It uses a security configuration database that defines services and ports required for each server role supported by the Security Configuration Wizard. The security configuration database is a set of .xml files installed in %SystemRoot%\Security\Msscw\Kbs.

NOTE Centralizing the security configuration database

In an enterprise environment, centralize the security configuration database so that administrators use the same database when running the Security Configuration Wizard. Copy the files in the %SystemRoot%\Security\Msscw\Kbs folder to a network folder; then launch the Security Configuration Wizard with the *Scw.exe* command, using the syntax **scw.exe /kb DatabaseLocation**. For example, the command *scw.exe /kb \\server01\scwkb* launches the Security Configuration Wizard, using the security configuration database in the shared folder *scwkb* on SERVER01.

The Security Configuration Wizard uses the security configuration database to scan the selected server and identifies the following:

- Roles that are installed on the server
- Roles likely being performed by the server
- Services installed on the server but not defined in the security configuration database
- IP addresses and subnets configured for the server

The information discovered about the server is saved in a file named *Main.xml*. This server-specific file is called the *configuration database*, not to be confused with the security configuration database used by the Security Configuration Wizard to perform the analysis. You can display this file by clicking the View Configuration Database button on the Processing Security Configuration page. The initial settings in the configuration database are called the *baseline settings*.

After the server has been scanned and the configuration database has been created, you have the opportunity to modify the database, which will then be used to generate the security policy to configure services, firewall rules, registry settings, and audit policies. The security policy can then be applied to the server or to other servers playing similar roles. The Security Configuration Wizard presents each of these four categories of the security policy in a section—a series of wizard pages.

- **Role-Based Service Configuration** The outcome of this section is a set of policies that configure the startup state of services on the server. You want to ensure that only the services required by the server's roles start and that other services do not start. To achieve this outcome, the Security Configuration Wizard presents pages that display the server roles, client features, and administration and other options detected on the scanned server. You can add or remove roles, features, and options to reflect the desired role configuration. The last page of the section, titled "Confirm Service Changes" and shown in Figure 7-9, shows the changes that will be made to services based on the roles you specify.

The server shown in Figure 7-9 is a domain controller, and you can see that the AD DS service is currently configured to start automatically; the policy will also set the service to start automatically to support the AD DS role. However, audio is not required for a DC, so the service named *Audiosrv* used by the Windows Audio option will be configured by the policy as disabled.

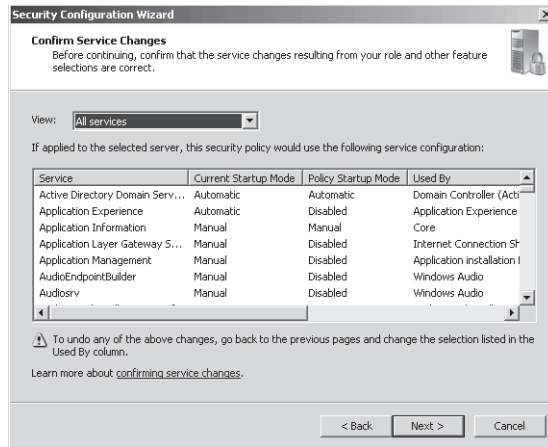


Figure 7-9 The Confirm Service Changes page of the Security Configuration Wizard

You cannot change the startup states on the Confirm Service Changes page of the Security Configuration Wizard. Instead, you must click the Back button to locate the role, service, or option indicated in the Used By column and either select or deselect that item. The service startup policies on the Confirm Service Changes page are determined by the selected roles, services, and options. Those not selected will result in service startup policy settings of disabled.

It is conceivable that the server on which you run the Security Configuration Wizard has services that are not defined by the Security Configuration Wizard security configuration database. The Select Additional Services page of the wizard enables you to include those services in the security policy so that, if the services exist on a system to which you apply the policy, those services will be started according to the startup setting in the baseline configuration database.

It is also conceivable that a server to which you apply the security policy might have services not found on the server from which you created the security policy. The Handling Unspecified Services page enables you to specify whether such services should be disabled or allowed to remain in their current startup mode.

- Network Security** The Network Security section produces the firewall settings of the security policy. Those settings will be applied by Windows Firewall with Advanced Security. Like the Role-Based Service Configuration section, the Network Security section displays a page of settings derived from the baseline settings in the configuration database. The settings in the Network Security section, however, are firewall rules rather than service startup modes. Figure 7-10 shows the rule that allows incoming ping requests to a domain controller. You can edit existing rules or add and remove custom rules.

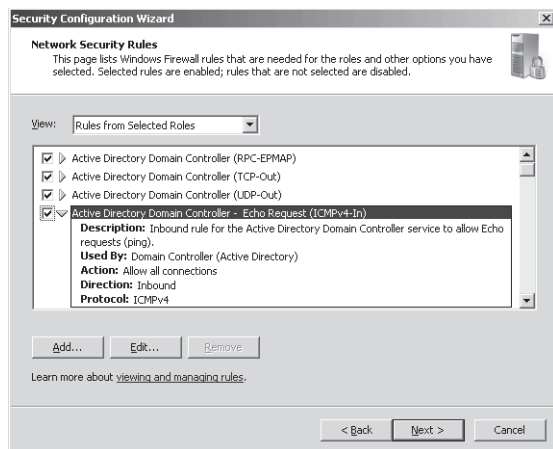


Figure 7-10 The Network Security Rules page of the Security Configuration Wizard

Windows Firewall with Advanced Security combines Internet Protocol security (IPSec) and a stateful firewall that inspects and filters all IP version 4 (IPv4) and IP version 6 (IPv6) packets, discarding unsolicited packets unless a firewall rule has been created to allow traffic explicitly to a port number, application name, or service name. The security policy generated by the Security Configuration Wizard manages firewall rules, but IPSec configuration is not provided by the Security Configuration Wizard.

- **Registry Settings** The Registry Settings section configures protocols used to communicate with other computers. These wizard pages determine server message block (SMB) packet signing, Lightweight Directory Access Protocol (LDAP) signing, LAN Manager (LM) authentication levels, and storage of password LM hash values. Each of these settings is described on the appropriate page, and a link on each page takes you to a Security Configuration Wizard Help page that details the setting.
- **Audit Policy** The Audit Policy section generates settings that manage the auditing of success and failure events and the file system objects that are audited. Additionally, the section enables you to incorporate a security template called SCWAudit.inf into the security policy. Use the Security Templates snap-in, described earlier in this lesson, to examine the settings in the template, which is located in %SystemRoot%\Security\Mscsw\Kbs.

You can skip any of the last three sections you do not want to include in your security policy. When all the configuration sections have been completed or skipped, the Security Configuration Wizard presents the Security Policy section. The Security Policy File Name page, shown in Figure 7-11, enables you to specify a path, a name, and a description for the security policy.

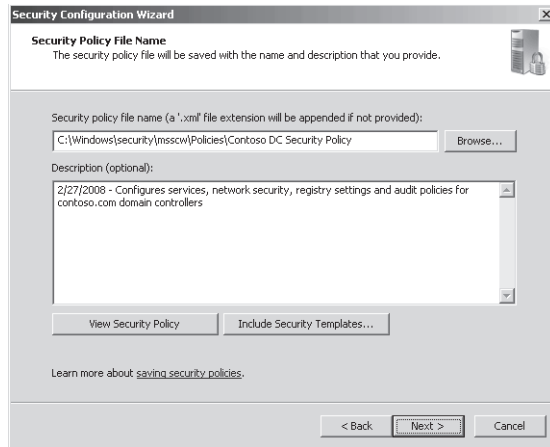


Figure 7-11 The Security Policy File Name page of the Security Configuration Wizard

Click the View Security Policy button to examine the settings of the security policy, which are very well documented by the Security Configuration Wizard. You can also import a security template into the security policy. Security templates, discussed earlier in this lesson in the “Managing Security Configuration with Security Templates” section, contain settings that are not provided by Managing Security Configuration with Security Templates, including restricted groups, event log policies, and file system and registry security policies. By including a security template, you can incorporate a richer collection of configuration settings in the security policy. If any settings in the security template conflict with the Security Configuration Wizard, the settings in the Security Configuration Wizard will take precedence. When you click the Next button, you are given the option to apply the security template to the server immediately or to apply the policy later.

Editing a Security Policy

You can edit a saved security policy by launching the Security Configuration Wizard and choosing Edit An Existing Security Policy on the Configuration Action page. Click the Browse button to locate the policy .xml file. When prompted to select a server, select the server that was used to create the security policy.

Applying a Security Policy

To apply a security policy to a server, open the Security Configuration Wizard and, on the Configuration Action page, choose Apply An Existing Security Policy. Click the Browse button to locate the policy .xml file. The server you specify on the Select Server page is the server to which the policy will be applied. Many of the changes specified in a security policy, including the addition of firewall rules for applications already running and the disabling of services, require that you restart the server. Therefore, as a best practice, it is recommended to restart a server any time you apply a security policy.

Rolling Back an Applied Security Policy

If a security policy is applied and causes undesirable results, you can roll back the changes by launching the Security Configuration Wizard and choosing Rollback The Last Applied Security Policy as the configuration action. When a security policy is applied by the Security Configuration Wizard, a rollback file is generated that stores the original settings of the system. The rollback process applies the rollback file.

Modifying Settings of an Applied Security Policy

Alternatively, if an applied security template does not produce an ideal configuration, you can manually change settings by using the Local Security Policy console discussed at the beginning of this lesson in the “Configuring the Local Security Policy” section. Thus, you can see the whole picture of security configuration, from manual settings to the generation of security templates to the creation of security policies with the Security Configuration Wizard, which can incorporate security templates, to the application of security policies and back to the manual configuration of settings.

Deploying a Security Policy Using Group Policy

You can apply a security policy created by the Security Configuration Wizard to a server by using the Security Configuration Wizard itself, by using the *Scwcmd.exe* command, or by transforming the security policy into a GPO. To transform a security policy into a GPO, log on as a domain administrator and run *Scwcmd.exe* with the *transform* command. For example, the *scwcmd transform /p:"Contoso DC Security.xml" /g:"Contoso DC Security GPO"* command will create a GPO called *Contoso DC Security GPO* with settings imported from the *Contoso DC Security.xml* security policy file. The resulting GPO can then be linked to an appropriate scope—site, domain, or OU—by using the Group Policy Management console. Be sure to type **scwcmd.exe transform /?** for help and guidance about this process.

Settings, Templates, Policies, and GPOs

As suggested in the introduction to this lesson, there are a number of mechanisms with which to manage security settings. You can use tools such as the Local Security Policy console to modify settings on an individual system. You can use security templates, which have existed since Windows 2000, to manage settings on one or more systems and to compare the current state of a system’s configuration against the desired configuration defined by the template. Security policies generated by the Security Configuration Wizard are the most recent addition to the security configuration management toolset. They are role-based .xml files that define service startup modes, firewall rules, audit policies, and some registry settings. Security policies can incorporate security templates. Both security templates and security policies can be deployed using Group Policy.

The plethora of tools available can make it difficult to identify the best practice for managing security on one or more systems. Plan to use Group Policy whenever possible to deploy security configuration. You can generate a GPO from a role-based security policy produced by the Security Configuration Wizard, which itself incorporates additional settings from a security template. After the GPO has been generated, you can make additional changes to the GPO by using the Group Policy Management Editor snap-in. Settings not managed by Group Policy can be configured on a server-by-server basis, using the local GPO security settings.

Lesson 3: Managing Software with Group Policy Software Installation

You might be aware of several tools that can be used to deploy software within an organization, including Microsoft System Center Configuration Manager (Configuration Manager) and its predecessor, Microsoft Systems Management Server (SMS). Although these tools provide great benefits, including features to meter software use and inventory systems, you can effectively deploy most software without these tools, using only Group Policy software installation (GPSI).

After this lesson, you will be able to:

- Deploy software using GPSI to computers and users.
- Remove software installed originally with GPSI.

Estimated lesson time: 45 minutes

Understanding Group Policy Software Installation

Group Policy software installation (GPSI) is used to create a managed software environment that has the following characteristics:

- Users have access to the applications they need to do their jobs, no matter which computer they log on to.
- Computers have the required applications, without intervention from a technical support representative.
- Applications can be updated, maintained, or removed to meet the needs of the organization.

The software installation extension is one of the many client-side extensions (CSEs) that support change and configuration management using Group Policy. CSEs were discussed in Chapter 6. The extension enables you to manage the initial deployment, the upgrades, and the removal of software centrally. All configuration of the software deployment is managed within a GPO, using procedures detailed later in this lesson.

Windows Installer Packages

GPSI uses the Windows Installer service to install, maintain, and remove software. The Windows Installer service manages software, using information contained in the application's Windows Installer package. The Windows Installer package is in a file with an .msi extension that describes the installed state of the application. The package contains explicit instructions regarding the installation and removal of an application. You can customize Windows Installer packages by using one of the following types of files:

- **Transform (.mst) files** These files provide a means for customizing the installation of an application. Some applications provide wizards or templates that permit a user to create transforms. For example, Adobe provides an enterprise deployment tool for Adobe Acrobat Reader that generates a transform. Many enterprises use the transform to configure agreement with the end user license agreement and to disable certain features of the application such as automatic updates that involve access to the Internet.
- **Patch (.msp) files** These files are used to update an existing .msi file for security updates, bug fixes, and service packs. An .msp file provides instructions about applying the updated files and registry keys in the software patch, service pack, or software update. For example, updates to Microsoft Office 2003 and later are provided as .msp files.

NOTE Installation of .msp and .mst files

You cannot deploy .mst or .msp files alone. They must be applied to an existing Windows Installer package.

GPSI can make limited use of non-MSI application files (.zap file), also known as down-level application packages, that specify the location of the software distribution point (SDP) and the setup command. See knowledge base article 231747 at <http://support.microsoft.com/?kbid=231747> for details. Most organizations do not use .zap files, however, because the installation of the application requires the user to have administrative privileges on the system. When GPSI installs an application by using a Windows Installer package, the user does not require administrative privileges, allowing for a more secure enterprise.

NOTE GPSI and Windows Installer packages

GPSI can fully manage applications only if the applications are deployed using Windows Installer packages. Other tools, including Configuration Manager and SMS, can manage applications that use other deployment mechanisms.

The .msi file, transforms, and other files required to install an application are stored in a shared SDP.

Software Deployment Options

You can deploy software by assigning applications to users or computers or by publishing applications for users. You *assign* required or mandatory software to users or to computers. You *publish* software that users might find useful in performing their jobs.

Exam Tip Know the difference between assigning applications and publishing applications.

Assigning Applications When you assign an application to a user, the application's local registry settings, including filename extensions, are updated and its shortcuts are created on the Start menu or desktop, thus advertising the availability of the application. The application advertisement follows the user regardless of which physical computer he or she logs on to. This application is installed the first time the user activates the application on the computer, either by selecting the application on the Start menu or by opening a document associated with the application. When you assign an application to the computer, the application is installed during the computer's startup process.

Publishing Applications When you publish an application to users, the application does not appear as if it is installed on the users' computers. No shortcuts are visible on the desktop or Start menu. Instead, the application appears as an available application for the user to install using Add Or Remove Programs in Control Panel on a Windows XP system or in Programs And Features on a Windows Server 2008 and Windows Vista system. Additionally, the application can be installed when a user opens a file type associated with the application. For example, if Acrobat Reader is advertised to users, it will be installed if a user opens a file with a .pdf extension.

Given that applications can be either assigned or published and targeted to users or computers, you can establish a workable combination to meet your software management goals. Table 7-1 details the different software deployment options.

Table 7-1 Software Deployment Options

	Publish (User Only)	Assign (User)	Assign (Computer)
After deployment of the GPO, the software is available for installation:	The next time a user logs on.	The next time a user logs on.	The next time the computer starts.
Typically, the user installs the software from:	The Control Panel Add Or Remove Programs (Windows XP) or Programs And Features (Windows Server 2008 and Windows Vista) applications.	Start menu or desktop shortcut. An application can also be configured to install automatically at logon.	The software is installed automatically when the computer starts up.
If the software is not installed and the user opens a file associated with the software, does the software install?	Yes (if auto-install is enabled).	Yes.	Does not apply; the software is already installed.

Table 7-1 Software Deployment Options

	Publish (User Only)	Assign (User)	Assign (Computer)
Can the user remove the software by using Control Panel?	Yes, and the user can choose to install it again from Control Panel.	Yes, and the software is available for installation again from the Start menu shortcuts or file associations.	No. Only a local administrator can remove the software; a user can run a repair on the software.
Supported installation files:	Windows Installer packages (.msi files), .zap files.	Windows Installer packages (.msi files).	Windows Installer packages (.msi files).

Quick Check

- You want to use GPSI to deploy an administrative tool so that it is available for administrators on any system to which they log on. You do not want the tool to install automatically because administrators do not need the tool on each computer, but you want the tool to be easily installed. Should you publish or assign the application? Describe how an administrator will install the tool.

Quick Check Answer

- Publish the application. An administrator will use the Programs And Features Control Panel application on a Windows Server 2008 and Windows Vista system or Add/Remove Programs on a Windows XP system to install the application.

Preparing an SDP

Now that you understand GPSI at a high level, you are ready to prepare the SDP. The SDP is simply a shared folder from which users and computers can install applications. Create a shared folder and create a separate folder for each application. Then copy the software package, modifications, and all other necessary files to the application folders. Set appropriate permissions on the folders that allow users or computers Read And Execute permission—the minimum permission required to install an application successfully from the SDP. The administrators of the SDP must be able to change and delete files to maintain the SDP over time.

Creating a Software Deployment GPO

To create a software deployment GPO, use the Group Policy Management console to create a new GPO or select an existing GPO. Edit the GPO, using Group Policy Management Editor. Expand the console nodes User Configuration\Policies\Software Settings\Software Installation. Alternatively, select the Software Installation node in the Computer Configuration branch. Right-click Software Installation, choose New, and then select Package. Browse to locate the .msi file for the application. Click Open. The Deploy Software dialog box appears,

shown in Figure 7-12. Select Published, Assigned, or Advanced. You cannot publish an application to computers, so the option will not be available if you are creating the package in the Software Installation node in Computer Configuration.

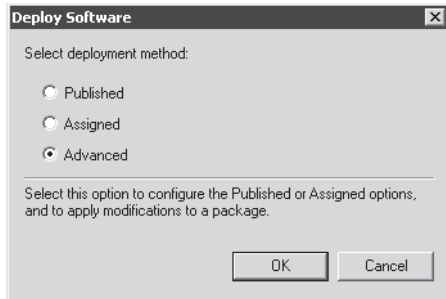


Figure 7-12 The Deploy Software dialog box

The Advanced option enables you to specify whether the application is published or assigned and gives you the opportunity to configure advanced properties of the software package. Therefore, it is recommended that you choose Advanced. The package properties dialog box then appears. Among the more important properties that you can configure are the following choices:

- **Deployment Type** On the Deployment tab, configure Published or Assigned.
- **Deployment Options** Based on the selected deployment type, different choices will appear in the Deployment Options section. These options, along with other settings on the Deployment tab, manage the behavior of the application installation.
- **Uninstall This Application When It Falls Out Of The Scope Of Management** If this option is selected, the application will be automatically removed when the GPO no longer applies to the user or computer.
- **Upgrades** On the Upgrades tab, you can specify the software that this package will upgrade. Upgrades are discussed in the “Maintaining Applications Deployed with Group Policy” section later in this lesson.
- **Categories** The Categories tab enables you to associate the package with one or more categories. Categories are used when an application is published to a user. When the user goes to Control Panel to install a program, applications published using GPSI are presented in groups based on these categories.

To create categories that are available to associate with packages, right-click Software Installation and choose Properties; then click the Categories tab.

- **Modifications** If you have a transform (.mst file) that customizes the package, click the Add button to associate the transform with the package. Most tabs in the package Properties dialog box are available for you to change settings at any time. However, the Modifications tab is available only when you create the new package and choose the Advanced option shown in Figure 7-12.

Managing the Scope of a Software Deployment GPO

After you have created a software deployment GPO, you can scope the GPO to distribute the software to appropriate computers or users. In many software management scenarios, applications should be assigned to computers rather than to users. This is because most software licenses allow an application to be installed on one computer, and if the application is assigned to a user, the application will be installed on each computer to which the user logs on.

As you learned in Chapter 6, you can scope a GPO by linking the GPO to an OU or by filtering the GPO so that it applies only to a selected global security group. Many organizations have found that it is easiest to manage software by linking an application's GPO to the domain and filtering the GPO with a global security group that contains the users and computers to which the application should be deployed. For example, a GPO that deploys the XML Notepad tool (available from the Microsoft downloads site at <http://www.microsoft.com/downloads>) would be linked to the domain and filtered with a group containing developers that require the tool. The group would have a descriptive name that indicates its purpose to manage the deployment of XML Notepad—*APP_XML Notepad*, for example.

Exam Tip On the 70-640 exam, you are likely to encounter questions that present software installation scenarios but are in fact testing your knowledge of how to scope a GPO effectively. As you read questions on the exam, try to identify what knowledge the question is really targeting.

Maintaining Applications Deployed with Group Policy

After a computer has installed an application by using the Windows Installer package specified by a GPO, the computer will not attempt to reinstall the application at each Group Policy refresh. There might be scenarios in which you want to force systems to reinstall the application. For example, small changes might have been made to the original Windows Installer package. To redeploy an application deployed with Group Policy, right-click the package in the GPO, choose All Tasks, and then select Redeploy Application.

You can also upgrade an application that has been deployed with GPSI. Create a package for the new version of the application in the Software Installation node of the GPO. The package can be in the same GPO as the package for the previous version or in any different GPO. Right-click the package and choose Properties. Click the Upgrades tab, and then click the Add button. The Add Upgrade Package dialog box appears, shown in Figure 7-13.

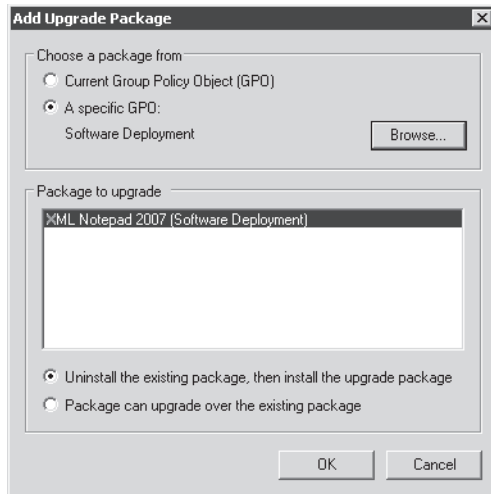


Figure 7-13 The Add Upgrade Package dialog box

Select whether the package for the previous version of the application is in the current GPO or in another GPO. If the previous package is in another GPO, click the Browse button to select that GPO. Then select the package from the Package To Upgrade list. Based on your knowledge of the application's upgrade behavior, choose one of the upgrade options shown at the bottom of Figure 7-13. Then click OK.

You can also remove an application that was deployed with GPSI. To do so, right-click the package, choose All Tasks, and then select Remove. In the Remove Software dialog box, choose one of the following two options:

- **Immediately Uninstall The Software From Users And Computers** This option, known as *forced removal*, causes computers to remove the application. The software installation extension will remove an application when the computer restarts if the application was deployed with a package in the Computer Configuration portion of the GPO. If the package is in the User Configuration portion, the application will be uninstalled the next time the user logs on.
- **Allows Users To Continue To Use The Software, But Prevents New Installations** This setting, known as *optional removal*, causes the software installation extension to avoid adding the package to systems that do not yet have the package installed. Computers that had previously installed the application do not forcibly uninstall the application, so users can continue using it.

If you use one of these two options to remove software using GPSI, it is important that you allow the settings in the GPO to propagate to all computers within the scope of the GPO before you delete, disable, or unlink the GPO. Clients need to receive this setting that specifies forced or optional removal. If the GPO is deleted or no longer applied before all clients have received

this setting, the software is not removed according to your instructions. This is particularly important in environments with mobile users on laptop computers that might not connect to the network on a regular basis.

If, when creating the software package, you chose the Uninstall This Application When It Falls Out Of The Scope Of Management option, you can simply delete, disable, or unlink the GPO, and the application will be forcibly removed by all clients that have the installed package with that setting.

GPSI and Slow Links

When a client performs a Group Policy refresh, it tests the performance of the network to determine whether it is connected using a slow link defined by default as 500 kilobits per second (kbps). Each client-side extension is configured to process Group Policy or to skip the application of settings on a slow link. By default, GPSI does not process Group Policy settings over a slow link because the installation of software over a slow link could cause significant delays.

You can change the slow link policy processing behavior of each client-side extension, using policy settings located in Computer Configuration\Policies\Administrative Templates\System\Group Policy. For example, you could modify the behavior of the software installation extension so that it does process policies over a slow link.

You can also change the connection speed threshold that constitutes a slow link. By configuring a low threshold for the connection speed, you can convince the client-side extensions that a connection is not a slow link, even if it actually is. There are separate Group Policy Slow Link Detection policy settings for computer policy processing and user policy processing. The policies are in the Administrative Templates\System\Group Policy folders in Computer Configuration and User Configuration.

Lesson 4: Auditing

Auditing is an important component of security. Auditing logs specified activities in your enterprise to the Windows Security log, which you can then monitor to understand those activities and to identify issues that warrant further investigation. Auditing can log successful activities to provide documentation of changes. It can also log failed and potentially malicious attempts to access enterprise resources. Auditing involves up to three management tools: audit policy, auditing settings on objects, and the Security log. In this lesson, you will learn how to configure auditing to address several common scenarios.

After this lesson, you will be able to:

- Configure audit policy.
- Configure auditing settings on file system and directory service objects.
- Implement Windows Server 2008 new Directory Service Changes auditing.
- View the Security log, using the Event Viewer snap-in.

Estimated lesson time: 45 minutes

Audit Policy

Audit Policy configures a system to audit categories of activities. If Audit Policy is not enabled, a server will not audit those activities. Figure 7-14 shows the Audit Policy node of a GPO expanded.

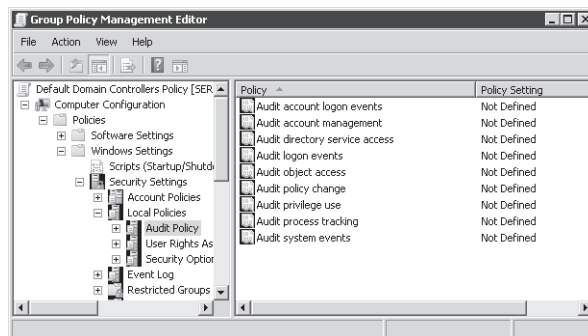


Figure 7-14 The Audit Policy node of a GPO

To configure auditing, you must define the policy setting. Double-click any policy setting and select the Define These Policy Settings check box. Then select whether to enable auditing of Success events, Failure events, or both. Table 7-2 defines each audit policy and its default settings on a Windows Server 2008 domain controller.

Table 7-2 Audit Policies

Audit Policy Setting	Explanation	Default Setting for Windows Server 2008 Domain Controllers
Audit Account Logon Events	Creates an event when a user or computer attempts to authenticate using an Active Directory account. For example, when a user logs on to any computer in the domain, an account logon event is generated.	Successful and failed account logons are audited.
Audit Logon Events	Creates an event when a user logs on interactively (locally) to a computer or over the network (remotely). For example, if a workstation and a server are configured to audit logon events, the workstation audits a user logging on directly to that workstation. When the user connects to a shared folder on the server, the server logs that remote logon. When a user logs on, the domain controller records a logon event because logon scripts and policies are retrieved from the DC.	Successful and failed logons are audited.
Audit Account Management	Audits events, including the creation, deletion, or modification of user, group, or computer accounts and the resetting of user passwords.	Successful account management activities are audited.
Audit Directory Service Access	Audits events that are specified in the system SACL, which is seen in an Active Directory object's Properties Advanced Security Settings dialog box. In addition to defining the audit policy with this setting, you must also configure auditing for the specific object or objects using the SACL of the object or objects. This policy is similar to the Audit Object Access policy used to audit files and folders, but this policy applies to Active Directory objects.	Successful directory service access events are audited, but few objects' SACLs specify audit settings. See the discussion in the "Auditing Directory Services Changes" section for more information.
Audit Policy Change	Audits changes to user rights assignment policies, audit policies, or trust policies.	Successful policy changes are audited.
Audit Privilege Use	Audits the use of a privilege or user right. See the explanatory text for this policy in Group Policy Management Editor (GPME).	No auditing is performed, by default.
Audit System Events	Audits system restart, shutdown, or changes that affect the system or security log.	Successful and failed system events are audited.

Table 7-2 Audit Policies

Audit Policy Setting	Explanation	Default Setting for Windows Server 2008 Domain Controllers
Audit Process Tracking	Audits events such as program activation and process exit. See the explanatory text for this policy in GPME.	Successful process tracking events are audited.
Audit Object Access	Audits access to objects such as files, folders, registry keys, and printers that have their own SACLs. In addition to enabling this audit policy, you must configure the auditing entries in objects' SACLs.	Successful object access events are audited.

Exam Tip Microsoft certification exams often test your knowledge of audit policies at a high level. Commit the information in Table 7-2 to memory and you are likely to be able to answer one or more exam items correctly.

As you can see, most major Active Directory events are already audited by domain controllers, assuming that the events are successful. Therefore, the creation of a user, the resetting of a user's password, the logon to the domain, and the retrieval of a user's logon scripts are all logged.

However, not all failure events are audited by default. You might need to implement additional failure auditing based on your organization's IT security policies and requirements. Auditing failed account logon events, for example, will expose malicious attempts to access the domain by repeatedly trying to log on as a domain user account without yet knowing the account's password. Auditing failed account management events can reveal someone attempting to manipulate the membership of a security-sensitive group.

One of the most important tasks you must fulfill is to balance and align Audit Policy with your corporate policies and reality. Your corporate policy might state that all failed logons and successful changes to Active Directory users and groups must be audited. That's easy to achieve in Active Directory. But how, exactly, are you going to use that information? Verbose auditing logs are useless if you don't know how or don't have the tools to manage those logs effectively. To implement auditing, you must have the business requirement to audit, a well-configured audit policy, and the tools with which to manage audited events.

Auditing Access to Files and Folders

Many organizations elect to audit file system access to provide insight into resource usage and potential security issues. Windows Server 2008 supports granular auditing based on user or group accounts and the specific actions performed by those accounts. To configure auditing, you must complete three steps: specify auditing settings, enable audit policy, and evaluate events in the security log.

Specifying Auditing Settings on a File or Folder

You can audit access to a file or folder by adding auditing entries to its SACL. To access the SACL and its audit entries, open the Properties dialog box and click the Security tab. Then click the Advanced button and click the Auditing tab. The Advanced Security Settings dialog box of a folder named Confidential Data is shown in Figure 7-15.

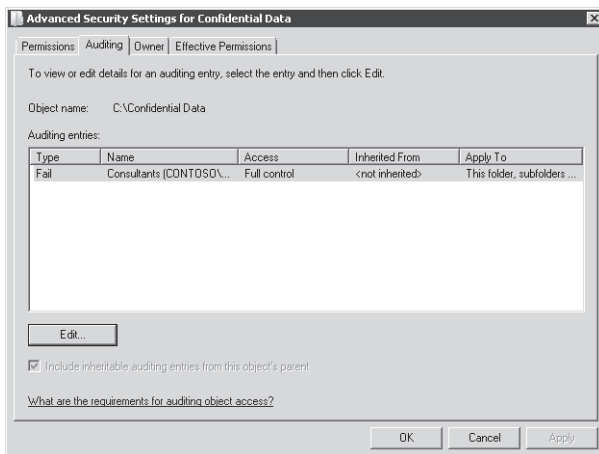


Figure 7-15 The Advanced Security Settings dialog box of a folder named Confidential Data

To add an entry, click the Edit button to open the Auditing tab in Edit mode. Click the Add button to select the user, group, or computer to audit. Then, in the Auditing Entry dialog box shown in Figure 7-16, indicate the type of access to audit.

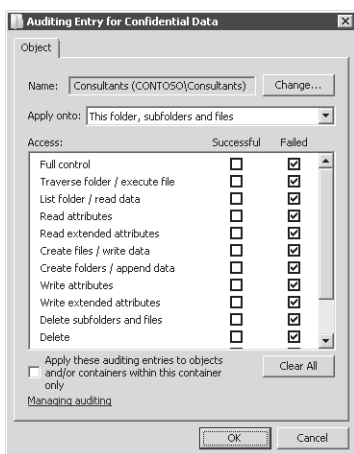


Figure 7-16 The Auditing Entry dialog box

You are able to audit for successes, failures, or both as the specified user, group, or computer attempts to access the resource by using one or more of the granular access levels.

You can audit successes for the following purposes:

- To log resource access for reporting and billing
- To monitor access that would suggest users are performing actions greater than what you had planned, indicating that permissions are too generous
- To identify access that is out of character for a particular account, which might be a sign that a user account has been breached by a hacker

Auditing failed events enables you:

- To monitor for malicious attempts to access a resource to which access has been denied.
- To identify failed attempts to access a file or folder to which a user does require access. This would indicate that the permissions are not sufficient to achieve a business requirement.

Auditing entries direct Windows to audit the successful or failed activities of a security principal (user, group, or computer) to use a specific permission. The example in Figure 7-15 audits for unsuccessful attempts by users in the Consultants group to access data in the Confidential Data folder at any level. It does that by configuring an auditing entry for Full Control access. Full Control includes all the individual access levels, so this entry covers any type of access. If a Consultant group member attempts access of any kind and fails, the activity will be logged.

Typically, auditing entries reflect the permission entries for the object. In other words, you would configure the Confidential Data folder with permissions that prevent members of the Consultants group from accessing its contents. You would then use auditing to monitor members of the Consultants group who nonetheless attempt to access the folder. Keep in mind, of course, that a member of the Consultants group can also belong to another group that does have permission to access the folder. Because that access will be successful, the activity is not logged. Therefore, if you really are concerned about keeping users out of a folder and making sure they do not access it in any way, monitor failed access attempts; however, also audit successful access to identify situations in which a user is accessing the folder through another group membership that is potentially incorrect.

NOTE Don't over-audit

Audit logs have the tendency to get quite large quite rapidly, so a golden rule for auditing is to configure the bare minimum required to achieve the business task. Specifying to audit the successes and failures on an active data folder for the Everyone group using Full Control (all permissions) would generate enormous audit logs that could affect the performance of the server and make locating a specific audited event all but impossible.

Enabling Audit Policy

Configuring auditing entries in the security descriptor of a file or folder does not, in itself, enable auditing. Auditing must be enabled by defining the Audit Object Access setting shown in Figure 7-17. After auditing is enabled, the security subsystem begins to pay attention to the audit settings and to log access as directed by those settings.

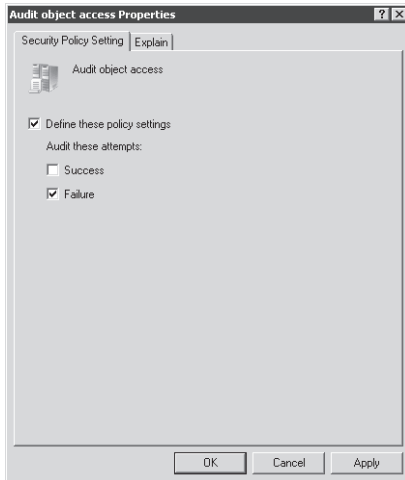


Figure 7-17 The Audit Object Access policy

The policy setting must be applied to the server that contains the object being audited. You can configure the policy setting in the server's local GPO or use a GPO scoped to the server.

You can define the policy then to audit Success events, Failure events, or both. The policy setting (shown in Figure 7-17) must specify auditing of Success or Failure attempts that match the type of auditing entry in the object's SACL (shown in Figure 7-16). For example, to log a failed attempt by a member of the Consultants group to access the Confidential Data folder, you must configure the Audit Object Access policy to audit failures, and you must configure the SACL of the Confidential Data folder to audit failures. If Audit Policy audits successes only, the failure entries in the folder's SACL will not trigger logging.

NOTE Making sure Audit Policy matches auditing entries

Remember that access that is audited and logged is the combination of the audit entries on specific files and folders and the settings in Audit Policy. If you've configured audit entries to log failures, but the policy enables only logging for successes, your audit logs will remain empty.

Evaluating Events in the Security Log

After you have enabled the Audit Object Access policy setting and specified the access you want to audit, using object SACLs, the system will begin to log access according to the audit entries. You can view the resulting events in the Security log of the server. Open the Event Viewer console from Administrative Tools. Expand Windows Logs\Security.

Exam Tip Auditing access to objects such as files and folders requires three components. First, the Audit Object Access policy must be enabled and configured to audit Success or Failure events as appropriate for the scenario. Second, the SACL of the object must be configured to audit successful or failed access. Third, you must examine the Security log. Audit Policy is often managed using a GPO, so the GPO must be scoped to apply to the server with the file or folder, which is usually a file server rather than a domain controller. Some exam questions that appear to be testing your knowledge of auditing are actually testing your ability to scope a GPO with Audit Policy to the correct servers.

Auditing Directory Service Changes

Just as the Audit Object Access policy enables you to log attempts to access objects such as files and folders, the Audit Directory Service Access policy enables you to log attempts to access objects in Active Directory. The same basic principles apply. You configure the policy to audit success or failure. You then configure the SACL of the Active Directory object to specify the types of access you want to audit.

As an example, if you want to monitor changes to the membership of a security-sensitive group such as Domain Admins, you can enable the Audit Directory Service Access policy to audit Success events. You can then open the SACL of the Domain Admins group and configure an auditing entry for successful modifications of the group's *member* attribute. You will do this in an exercise in this lesson's practice.

In Microsoft Windows Server 2003 and Windows 2000 Server, you could audit directory service access and you would be notified that an object, or the property of an object, had been changed, but you could not identify the previous and new values of the attribute that had changed. For example, an event could be logged indicating that a particular user changed the *member* attribute of Domain Admins, but you could not determine exactly what change was made.

Windows Server 2008 adds an auditing category called Directory Service Changes. The important distinction between Directory Service Changes and Directory Service Access is that with Directory Service Changes auditing, you can identify the previous and current values of a changed attribute.

Directory Service Changes is not enabled in Windows Server 2008 by default. Instead, Directory Service Access is enabled to mimic the auditing functionality of previous versions of Windows.

To enable auditing of successful Directory Service Changes, open a command prompt on a domain controller and type this command:

```
auditpol /set /subcategory:"directory service changes" /success:enable
```

Exam Tip The *auditpol* command is used to enable auditing of directory service changes.

You must still modify the SACL of objects to specify which attributes should be audited. Although you can use the preceding command to enable Directory Service Changes auditing in a lab and explore the events that are generated, don't implement this in a domain until you've read the documentation on TechNet, starting with the step-by-step guide found at <http://technet2.microsoft.com/windowsserver2008/en/library/a9c25483-89e2-4202-881c-ea8e02b4b2a51033.mspx>.

When Directory Service Changes auditing is enabled, and auditing entries are configured in the SACL of directory service objects, events are logged to the Security log that clearly indicate the attribute that was changed and when the change was made. In most cases, event log entries will show the previous and current value of the changed attribute.

Quick Check

- You want to audit changes to properties of user accounts provided for temporary employees. When a change is made, you want to see the previous and new value of the changed attribute. What type of auditing do you perform?

Quick Check Answer

- Directory Services Changes auditing

Chapter 8

Authentication

When a user logs on to an Active Directory Domain Services (AD DS) domain, she enters her user name and password, and the client uses those credentials to *authenticate* the user—to validate the user’s identity against her Active Directory account. In Chapter 3, “Users,” you learned how to create and manage user accounts and their properties, including their passwords. In this chapter, you will explore the domain-side components of authentication, including the policies that specify password requirements and the auditing of authentication-related activities. You will also discover two new options, password settings objects (PSOs) and read-only domain controllers (RODCs).

Exam objectives in this chapter:

- Creating and Maintaining Active Directory Objects
 - Configure account policies.
 - Configure audit policy by using GPOs.
- Configuring the Active Directory Infrastructure
 - Configure Active Directory replication.
- Configuring Additional Active Directory Server Roles
 - Configure the read-only domain controller (RODC).

Before You Begin

To complete the lessons in this chapter, you must have installed a domain controller named SERVER01 in the *contoso.com* domain.

Real World

Dan Holme

As I work with clients to implement AD DS, I must constantly balance the need to maintain high levels of security with the need to continue conducting the client's business. With versions of Microsoft Windows prior to Windows Server 2008, I constantly ran into two scenarios in which this balance was particularly difficult to reach. The first relates to the security of user accounts with high levels of privilege within the enterprise. Such accounts are particularly attractive to hackers, so they should be locked down with particularly lengthy and complex passwords. In earlier versions of Windows, only one password policy could be applied to all accounts in the domain. Therefore, I either had to apply the highly restrictive password policy to all users in the domain, which was never a palatable solution, or ask administrators to follow the more restrictive policy but with no way to require compliance. Windows Server 2008 introduces fine-grained password policies that can be used to apply more or less restrictive passwords after requirements to groups or users in a domain.

Branch offices were also highly problematic because I had to balance the user's need to be authenticated quickly and reliably against the branch office's desire to centralize control over the physical security of domain controllers. Placing a domain controller in a branch office would clearly improve performance for users in the office but would also typically expose the domain controller to lower levels of security than those maintained at the data center. Coming to the rescue once again, Windows Server 2008 can act as a read-only domain controller, authenticating users and the branch office without storing all domain user credentials, thus reducing the risk to the enterprise in the event of a stolen branch office domain controller.

If you have worked with Active Directory for any period of time, you already appreciate the value of fine-grained password policies and read-only domain controllers. If you are new to Active Directory, you are lucky to be able to work with these much-anticipated new features.

Lesson 1: Configuring Password and Lockout Policies

In a Windows Server 2008 domain, users are required to change their password every 42 days, and a password must be at least seven characters long and meet complexity requirements including the use of three of four character types: uppercase, lowercase, numeric, and nonalphanumeric. Three password policies—maximum password age, password length, and password complexity—are among the first policies encountered by administrators and users alike in an Active Directory domain. Rarely do these default settings align precisely with the password security requirements of an organization. Your organization might require passwords to be changed more or less frequently or to be longer. In this lesson, you'll learn how to implement your enterprise's password and lockout policies by modifying the Default Domain Policy Group Policy object (GPO).

There are exceptions to every rule, and you likely have exceptions to your password policies. To enhance the security of your domain, you can set more restrictive password requirements for accounts assigned to administrators, for accounts used by services such as Microsoft SQL Server, or for a backup utility. In earlier versions of Windows, this was not possible; a single password policy applied to all accounts in the domain. In this lesson, you will learn to configure fine-grained password policies, a new feature in Windows Server 2008 that enables you to assign different password policies to users and groups in your domain.

After this lesson, you will be able to:

- Implement your domain password policy.
- Configure and assign fine-grained password policies.

Estimated lesson time: 45 minutes

Understanding Password Policies

Your domain's password policy is configured by a GPO scoped to the domain. Within the GPO, in the Computer Configuration\Policies\Windows Settings\Security Settings\Account Policies\Password Policy node, you can configure the policy settings that determine password requirements. The Password Policy node is shown in Figure 8-1.

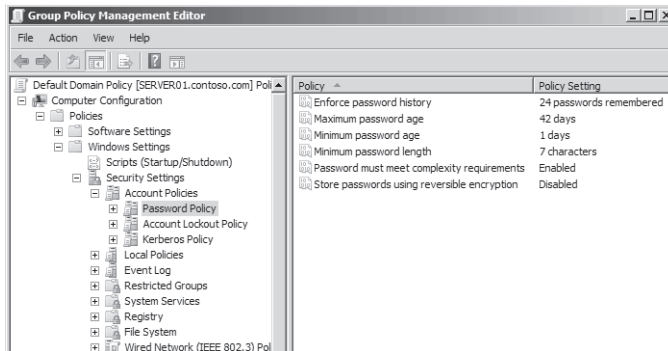


Figure 8-1 The Password Policy node of a GPO

You can understand the effects of the policies by considering the life cycle of a user password. A user will be required to change his or her password within the number of days specified by the Maximum Password Age policy setting. When the user enters a new password, the length of the new password will be compared to the number of characters in the Minimum Password Length policy. If the Password Must Meet Complexity Requirements policy is enabled, the password must contain at least three of four character types:

- Uppercase—for example, A–Z
- Lowercase—for example, a–z
- Numeric—0–9
- Nonalphanumeric—symbols such as !, #, %, or &

If the new password meets requirements, Active Directory puts the password through a mathematical algorithm that produces a representation of the password called the *hash code*. The hash code is unique; no two passwords can create the same hash code. The algorithm used to create the hash code is called a one-way function. You cannot put the hash code through a reverse function to derive the password. The fact that a hash code, and not the password itself, is stored in Active Directory helps increase the security of the user account.

Occasionally, applications require the ability to read a user's password. This is not possible because, by default, only the hash code is stored in Active Directory. To support such applications, you can enable the Store Passwords Using Reversible Encryption policy. This policy is not enabled by default, but if you enable the policy, user passwords are stored in an encrypted form that can be decrypted by the application. Reversible encryption significantly reduces the security of your domain, so it is disabled by default, and you should strive to eliminate applications that require direct access to passwords.

Additionally, Active Directory can check a cache of the user's previous hash codes to make sure that the new password is not the same as the user's previous passwords. The number of previous passwords against which a new password is evaluated is determined by the Enforce Password History policy. By default, Windows maintains the previous 24 hash codes.

If a user is determined to reuse a password when the password expiration period occurs, he or she could simply change the password 25 times to work around the password history. To prevent that from happening, the Minimum Password Age policy specifies an amount of time that must pass between password changes. By default, it is one day. Therefore, the determined user would have to change his or her password once a day for 25 days to reuse a password. This type of deterrent is generally successful at discouraging such behavior.

Each of these policy settings affects a user who changes his or her password. The settings do not affect an administrator using the Reset Password command to change another user's password.

Understanding Account Lockout Policies

An intruder can gain access to the resources in your domain by determining a valid user name and password. User names are relatively easy to identify because most organizations create user names from an employee's e-mail address, initials, combinations of first and last names, or employee IDs. When a user name is known, the intruder must determine the correct password by guessing or by repeatedly logging on with combinations of characters or words until the logon is successful.

This type of attack can be thwarted by limiting the number of incorrect logons that are allowed. That is exactly what account lockout policies achieve. Account lockout policies are located in the node of the GPO directly below Password Policy. The Account Lockout Policy node is shown in Figure 8-2.

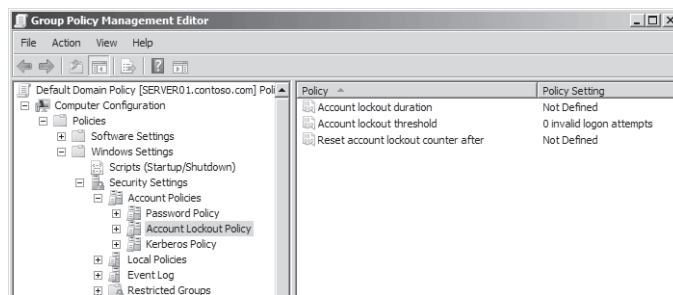


Figure 8-2 The Account Lockout Policy node of a GPO

Three settings are related to account lockout. The first, Account Lockout Threshold, determines the number of invalid logon attempts permitted within a time specified by the Account Lockout Duration policy. If an attack results in more unsuccessful logons within that time-frame, the user account is locked out. When an account is locked out, Active Directory will deny logon to that account, even if the correct password is specified.

An administrator can unlock a locked user account by following the procedure you learned in Chapter 3. You can also configure Active Directory to unlock the account automatically after a delay specified by the Reset Account Lockout Counter After policy setting.

Configuring the Domain Password and Lockout Policy

Active Directory supports one set of password and lockout policies for a domain. These policies are configured in a GPO that is scoped to the domain. A new domain contains a GPO called the Default Domain Policy that is linked to the domain and that includes the default policy settings shown in Figure 8-1 and Figure 8-2. You can change the settings by editing the Default Domain Policy.

Practice It You can practice configuring a domain's password and lockout policies in Exercise 1, "Configure the Domain's Password and Lockout Policies," in the practice for this lesson.

The password settings configured in the Default Domain Policy affect all user accounts in the domain. The settings can be overridden, however, by the password-related properties of the individual user accounts. On the Account tab of a user's Properties dialog box, you can specify settings such as Password Never Expires or Store Passwords Using Reversible Encryption. For example, if five users have an application that requires direct access to their passwords, you can configure the accounts for those users to store their passwords, using reversible encryption.

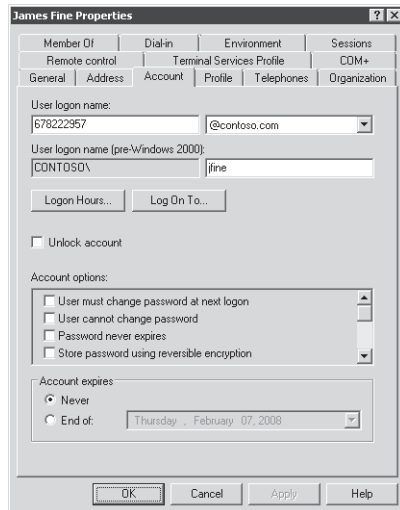


Figure 8-3 Password-related properties of a user account

Fine-Grained Password and Lockout Policy

You can also override the domain password and lockout policy by using a new feature of Windows Server 2008 called *fine-grained password and lockout policy*, often shortened to simply *fine-grained password policy*. Fine-grained password policy enables you to configure a policy

that applies to one or more groups or users in your domain. To use fine-grained password policy, your domain must be at the Windows Server 2008 domain functional level described in Chapter 12, “Domains and Forests.”

This feature is a highly anticipated addition to Active Directory. There are several scenarios for which fine-grained password policy can be used to increase the security of your domain. Accounts used by administrators are delegated privileges to modify objects in Active Directory; therefore, if an intruder compromises an administrator’s account, more damage can be done to the domain than could be done through the account of a standard user. For that reason, consider implementing stricter password requirements for administrative accounts. For example, you might require greater password length and more frequent password changes.

Accounts used by services such as SQL Server also require special treatment in a domain. A service performs its tasks with credentials that must be authenticated with a user name and password just like those of a human user. However, most services are not capable of changing their own password, so administrators configure service accounts with the Password Never Expires option enabled. When an account’s password will not be changed, make sure the password is difficult to compromise. You can use fine-grained password policies to specify an extremely long minimum password length and no password expiration.

Understanding Password Settings Objects

The settings managed by fine-grained password policy are identical to those in the Password Policy and Accounts Policy nodes of a GPO. However, fine-grained password policies are not implemented as part of Group Policy, nor are they applied as part of a GPO. Instead, there is a separate class of object in Active Directory that maintains the settings for fine-grained password policy: the *password settings object* (PSO).

Exam Tip There can be one, and only one, authoritative set of password and lockout policy settings that applies to all users in a domain. Those settings are configured in the Default Domain Policy GPO. Fine-grained password policies, which apply to individual groups or users in the domain, are implemented using PSOs.

Most Active Directory objects can be managed with user-friendly graphical user interface (GUI) tools such as the Active Directory Users and Computers snap-in. You manage PSOs, however, with low-level tools, including ADSI Edit.

MORE INFO Password Policy Basic

Although it will not be addressed on the 70-640 exam, it is highly recommended that you use Password Policy Basic by Special Operations Software to manage fine-grained password policy. The GUI tool can be downloaded free from <http://www.specopssoft.com>.

You can create one or more PSOs in your domain. Each PSO contains a complete set of password and lockout policy settings. A PSO is applied by linking the PSO to one or more global security groups or users. For example, to configure a strict password policy for administrative accounts, create a global security group, add the service user accounts as members, and link a PSO to the group. Applying fine-grained password policies to a group in this manner is more manageable than applying the policies to each individual user account. If you create a new service account, you simply add it to the group, and the account becomes managed by the PSO.

PSO Precedence and Resultant PSO

A PSO can be linked to more than one group or user, an individual group or user can have more than one PSO linked to it, and a user can belong to multiple groups. So which fine-grained password and lockout policy settings apply to a user? One and only one PSO determines the password and lockout settings for a user; this PSO is called the *resultant PSO*. Each PSO has an attribute that determines the precedence of the PSO. The precedence value is any number greater than 0, where the number 1 indicates highest precedence. If multiple PSOs apply to a user, the PSO with the highest precedence (closest to 1) takes effect. The rules that determine precedence are as follows:

- If multiple PSOs apply to groups to which the user belongs, the PSO with the highest precedence prevails.
- If one or more PSOs are linked directly to the user, PSOs linked to groups are ignored, regardless of their precedence. The user-linked PSO with highest precedence prevails.
- If one or more PSOs have the same precedence value, Active Directory must make a choice. It picks the PSO with the lowest globally unique identifier (GUID). GUIDs are like serial numbers for Active Directory objects—no two objects have the same GUID. GUIDs have no particular meaning—they are just identifiers—so choosing the PSO with the lowest GUID is, in effect, an arbitrary decision. Configure PSOs with unique, specific precedence values so that you avoid this scenario.

These rules determine the resultant PSO. Active Directory exposes the resultant PSO in a user object attribute, so you can readily identify the PSO that will affect a user. You will examine that attribute in the practice at the end of this lesson. PSOs contain all password and lockout settings, so there is no inheritance or merging of settings. The resultant PSO is the authoritative PSO.

PSOs and OUs

PSOs can be linked to global security groups or users. PSOs cannot be linked to organizational units (OUs). If you want to apply password and lockout policies to users in an OU, you must create a global security group that includes all the users in the OU. This type of group is called a *shadow group*—its membership shadows, or mimics, the membership of an OU.

Quick Check

- You want to require that administrators maintain a password of at least 15 characters and change the password every 45 days. The administrators' user accounts are in an OU called Admins. You do not want to apply the restrictive password policy to all domain users. What do you do?

Quick Check Answer

- Create a global security group that contains all users in the Admins OU. Create a PSO that configures the password policies and link the PSO to the group.

Shadow groups are conceptual, not technical objects. You simply create a group and add the users that belong to the OU. If you change the membership of the OU, you must also change the membership of the group.

MORE INFO Shadow groups

Additional information about PSOs and shadow groups is available at <http://technet2.microsoft.com/windowsserver2008/en/library/2199dcf7-68fd-4315-87cc-ade35f8978ea1033.aspx?mfr=true>.

MORE INFO Maintaining shadow group membership with scripts

You can use scripts to maintain the membership of shadow groups dynamically so that they always reflect the users in OUs. You can find example scripts in *Windows Administration Resource Kit: Productivity Solutions for IT Professionals* by Dan Holme (Microsoft Press, 2008).

Lesson 2: Auditing Authentication

In Chapter 7, “Group Policy Settings,” you learned to configure auditing for several types of activities, including access to folders and changes to directory service objects. Windows Server 2008 also enables you to audit the logon activity of users in a domain. By auditing successful logons, you can look for instances in which an account is being used at unusual times or in unexpected locations, which might indicate that an intruder is logging on to the account. Auditing failed logons can reveal attempts by intruders to compromise an account. In this lesson, you will learn to configure logon auditing.

After this lesson, you will be able to:

- Configure auditing of authentication-related activity.
- Distinguish between account logon and logon events.
- Identify authentication-related events in the Security log.

Estimated lesson time: 30 minutes

Account Logon and Logon Events

This lesson examines two specific policy settings: Audit Account Logon Events and Audit Logon Events. It is important to understand the difference between these two similarly named policy settings.

When a user logs on to any computer in the domain using his or her domain user account, a domain controller authenticates the attempt to log on to the domain account. This generates an account logon event on the domain controller.

The computer to which the user logs on—for example, the user’s laptop—generates a logon event. The computer did not authenticate the user against his or her account; it passed the account to a domain controller for validation. The computer did, however, allow the user to log on interactively to the computer. Therefore, the event is a logon event.

When the user connects to a folder on a server in the domain, that server authorizes the user for a type of logon called a network logon. Again, the server does not authenticate the user; it relies on the ticket given to the user by the domain controller. But the connection by the user generates a logon event on the server.

Exam Tip Be certain that you can distinguish between *account logon events* and *logon events*. The simplest way to remember the difference is that an account logon event occurs where the account lives: on the domain controller that authenticates the user. A logon event occurs on the computer to which the user logs on interactively. It also occurs on the file server to which the user connects using a network logon.

Configuring Authentication-Related Audit Policies

Account logon and logon events can be audited by Windows Server 2008. The settings that manage auditing are located in a GPO in the Computer Configuration\Policies\Windows Settings\Security Settings\Local Policies\Audit Policy node. The Audit Policy node and the two settings are shown in Figure 8-4.

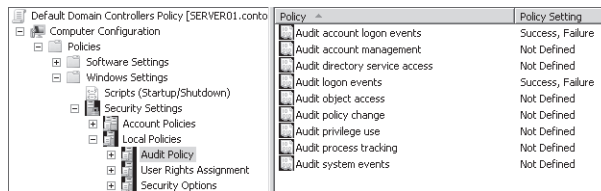


Figure 8-4 Authentication-related policy settings

To configure an audit policy, double-click the policy, and its properties dialog box appears. The Audit Account Logon Events Properties dialog box is shown in Figure 8-5.

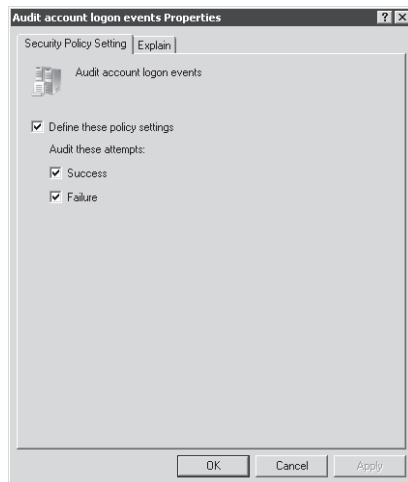


Figure 8-5 The Audit Account Logon Events Properties dialog box

The policy setting can be configured to one of the following four states:

- **Not defined** If the Define These Policy Settings check box is cleared, the policy setting is not defined. In this case, the server will audit the event based on its default settings or on the settings specified in another GPO.
- **Defined for no auditing** If the Define These Policy Settings check box is selected, but the Success and Failure check boxes are cleared, the server will not audit the event.
- **Audit successful events** If the Define These Policy Settings check box is selected, and the Success checkbox is selected, the server will log successful events in its Security log.

- **Audit failed to events** If the Define These Policy Settings check box is selected, and the Failure check box is selected, the server will log unsuccessful events in its Security log.

A server's audit behavior is determined by the setting that wins based on the rules of policy application discussed in Chapter 6, "Group Policy Infrastructure."

Scoping Audit Policies

As with all policy settings, be careful to scope settings so that they affect the correct systems. For example, if you want to audit attempts by users to connect to file servers in your enterprise, you can configure logon event auditing in a GPO linked to the OU that contains your file servers. Alternatively, if you want to audit logons by users to desktops in your human resources department, you can configure logon event auditing in a GPO linked to the OU containing human resources computer objects. Remember that domain users logging on to a client computer or connecting to a server will generate a logon event—not an account logon event—on that system.

Only domain controllers generate account logon events for domain users. Remember that an account logon event occurs on the domain controller that authenticates a domain user, regardless of where that user logs on. If you want to audit logons to domain accounts, scope account logon event auditing to affect only domain controllers. In fact, the Default Domain Controllers GPO that is created when you install your first domain controller is an ideal GPO in which to configure account logon audit policies.

In the previous section, you learned that if an event auditing policy is not defined, the system will audit based on the settings in other GPOs or on its default setting. In Windows Server 2008, the default setting is to audit successful account logon events and successful logon events, so both types of events are, if successful, entered in the server's Security log. If you want to audit failures or turn off auditing, you will need to define the appropriate setting in the audit policy.

Quick Check

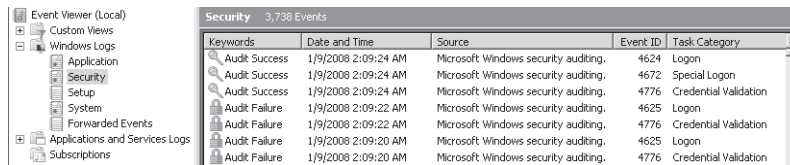
- You are concerned that an intruder is attempting to gain access to your network by guessing a user's password. You want to identify the times at which the intruder is trying to log on. What type of event should you audit? Should you configure the policy setting in the Default Domain Policy or in the Default Domain Controllers Policy?

Quick Check Answer

- Enable auditing of failed account logon events (not logon events) in the Default Domain Controllers GPO. Only domain controllers generate account logon events related to the authentication of domain users. The Default Domain Controllers GPO is scoped correctly to apply only to domain controllers.

Viewing Logon Events

Account logon and logon events, if audited, appear in the Security log of the system that generated the event. Figure 8-6 shows an example. Thus, if you are auditing logons to computers in the human resources department, the events are entered in each computer's Security log. Similarly, if you are auditing unsuccessful account logons to identify potential intrusion attempts, the events are entered in each domain controller's Security log. This means, by default, you will need to examine the Security logs of all domain controllers to get a complete picture of account logon events in your domain.



The screenshot shows the Windows Event Viewer interface. On the left, the 'Event Viewer (Local)' tree is expanded to 'Windows Logs' > 'Security'. The main pane displays a list of 3,738 events in the Security log. The following table represents the data shown in the screenshot:

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	1/9/2008 2:09:24 AM	Microsoft Windows security auditing.	4624	Logon
Audit Success	1/9/2008 2:09:24 AM	Microsoft Windows security auditing.	4672	Special Logon
Audit Success	1/9/2008 2:09:24 AM	Microsoft Windows security auditing.	4776	Credential Validation
Audit Failure	1/9/2008 2:09:22 AM	Microsoft Windows security auditing.	4625	Logon
Audit Failure	1/9/2008 2:09:22 AM	Microsoft Windows security auditing.	4776	Credential Validation
Audit Failure	1/9/2008 2:09:20 AM	Microsoft Windows security auditing.	4625	Logon
Audit Failure	1/9/2008 2:09:20 AM	Microsoft Windows security auditing.	4776	Credential Validation

Figure 8-6 Authentication events in the Security log

As you can imagine, in a complex environment with multiple domain controllers and many users, auditing account logons or logons can generate a tremendous number of events. If there are too many events, it can be difficult to identify problematic events worthy of closer investigation. Balance the amount of logging you perform with the security requirements of your business and the resources you have available to analyze logged events.

Lesson 3: Configuring Read-Only Domain Controllers

Branch offices present a unique challenge to an enterprise's IT staff: if a branch office is separated from the hub site by a wide area network (WAN) link, should you place a domain controller (DC) in the branch office? In previous versions of Windows, the answer to this question was not a simple one. Windows Server 2008, however, introduces a new type of DC—the read-only domain controller (RODC)—that makes the question easier to answer. In this lesson, you will explore the issues related to branch office authentication and DC placement, and you will learn how to implement and support a branch-office RODC.

After this lesson, you will be able to:

- Identify the business requirements for RODCs.
- Install an RODC.
- Configure password replication policy.
- Monitor the caching of credentials on an RODC.

Estimated lesson time: 60 minutes

Authentication and Domain Controller Placement in a Branch Office

Consider a scenario in which an enterprise is characterized by a hub site and several branch offices. The branch offices connect to the hub site over WAN links that might be congested, expensive, slow, or unreliable. Users in the branch office must be authenticated by Active Directory to access resources in the domain. Should a DC be placed in the branch office?

In branch office scenarios, many IT services are centralized in a hub site. The hub site is carefully maintained by the IT staff and includes secure facilities for services. The branch offices, however, offer inadequate security for servers and might have insufficient IT staff to support the servers.

If a DC is not placed in the branch office, authentication and service ticket activities will be directed to the hub site over the WAN link. Authentication occurs when a user first logs on to his or her computer in the morning. *Service tickets* are a component of the Kerberos authentication mechanism used by Windows Server 2008 domains. You can think of a service ticket as a key issued by the domain controller to a user. The key allows the user to connect to a service such as the File and Print services on a file server. When a user first tries to access a specific service, the user's client requests a service ticket from the domain controller. Because users typically connect to multiple services during a workday, service ticket activity happens regularly. Authentication and service ticket activity over the WAN link between a branch office and a hub site can result in slow or unreliable performance.

If a DC is placed in the branch office, authentication is much more efficient, but there are several potentially significant risks. A DC maintains a copy of all attributes of all objects in its domain, including secrets such as information related to user passwords. If a DC is accessed or stolen, it becomes possible for a determined expert to identify valid user names and passwords, at which point the entire domain is compromised. At a minimum, you must reset the passwords of every user account in the domain. Because the security of servers at branch offices is often less than ideal, a branch office DC poses a considerable security risk.

A second concern is that the changes to the Active Directory database on a branch office DC replicate to the hub site and to all other DCs in the environment. Therefore, corruption to the branch office DC poses a risk to the integrity of the enterprise directory service. For example, if a branch office administrator performs a restore of the DC from an outdated backup, there can be significant repercussions for the entire domain.

The third concern relates to administration. A branch office domain controller might require maintenance, for example, a new device driver. To perform maintenance on a standard domain controller, you must log on as a member of the Administrators group on the domain controller, which means you are effectively an administrator of the domain. It might not be appropriate to grant that level of capability to a support team at a branch office.

Read-Only Domain Controllers

These concerns—security, directory service integrity, and administration—left many enterprises with a difficult choice to make, and there was no best practices answer. The RODC is designed specifically to address the branch office scenario. An RODC is a domain controller, typically placed in the branch office, that maintains a copy of all objects in the domain and all attributes except secrets such as password-related properties. When a user in the branch office logs on, the RODC receives the request and forwards it to a domain controller in the hub site for authentication.

You are able to configure a password replication policy (PRP) for the RODC that specifies user accounts the RODC is allowed to cache. If the user logging on is included in the PRP, the RODC caches that user's credentials, so the next time authentication is requested, the RODC can perform the task locally. As users who are included in the PRP log on, the RODC builds its cache of credentials so that it can perform authentication locally for those users. These concepts are illustrated in Figure 8-7.

Because the RODC maintains only a subset of user credentials, if the RODC is compromised or stolen, the effect of the security exposure is limited; only the user accounts that had been cached on the RODC must have their passwords changed. Writable domain controllers maintain a list of all cached credentials on individual RODCs. When you delete the account of the stolen or compromised RODC from Active Directory, you are given the option to reset the passwords of all user accounts that were cached on the RODC. The RODC replicates changes

to Active Directory from DCs in the hub site. Replication is one way (from a writable domain controller to a RODC); no changes to the RODC are replicated to any other domain controller. This eliminates the exposure of the directory service to corruption resulting from changes made to a compromised branch office DC. Finally, RODCs, unlike writable DCs, have a local Administrators group. You can give one or more local support personnel the ability to maintain an RODC fully, without granting them the equivalence of domain administrators.

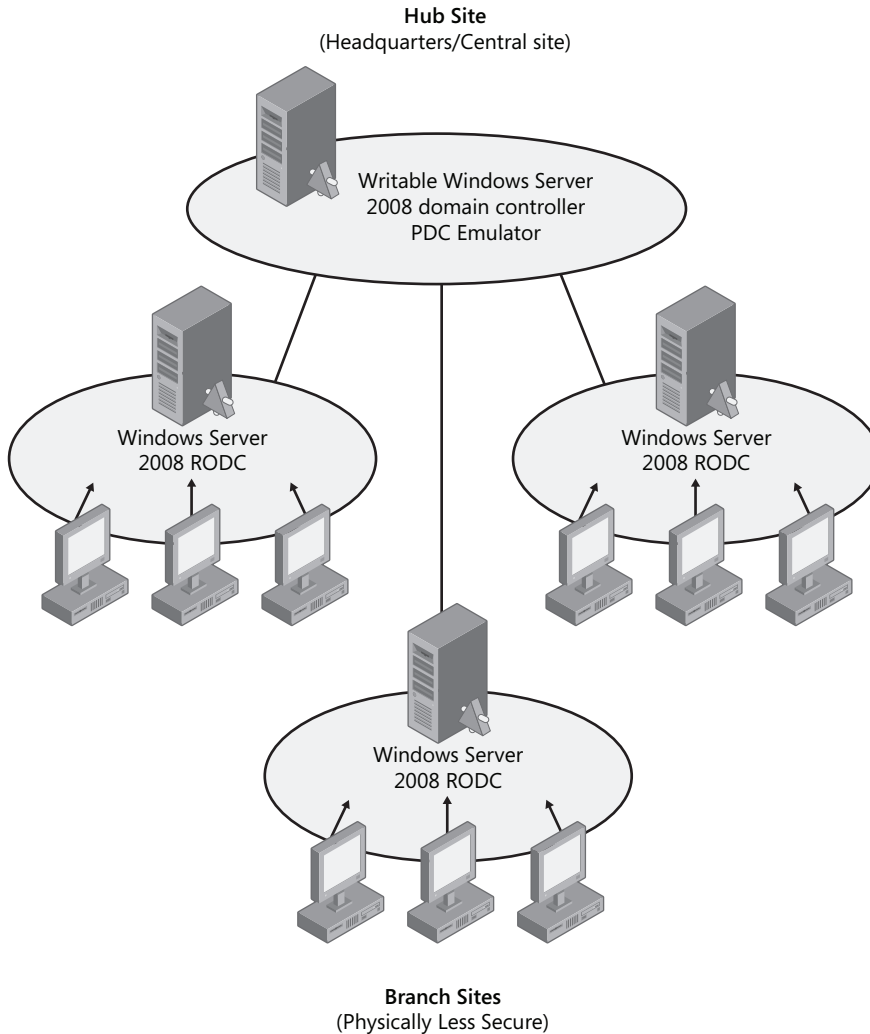


Figure 8-7 A branch office scenario supported by RODCs

Deploying an RODC

The high-level steps to install an RODC are as follows:

1. Ensure that the forest functional level is Windows Server 2003 or higher.
2. If the forest has any DCs running Microsoft Windows Server 2003, run *Adprep /rodcprep*.
3. Ensure that at least one writable DC is running Windows Server 2008.
4. Install the RODC.

Each of these steps is detailed in the following sections.

Verifying and Configuring Forest Functional Level of Windows Server 2003 or Higher

Functional levels enable features unique to specific versions of Windows and are, therefore, dependent on the versions of Windows running on domain controllers. If all domain controllers are Windows Server 2003 or later, the domain functional level can be set to Windows Server 2003. If all domains are at Windows Server 2003 domain functional level, the forest functional level can be set to Windows Server 2003. Domain and forest functional levels are discussed in detail in Chapter 12.

RODCs require that the forest functional level is Windows Server 2003 or higher. That means that all domain controllers in the entire forest are running Windows Server 2003 or later. To determine the functional level of your forest, open Active Directory Domains And Trusts from the Administrative Tools folder, right-click the name of the forest, choose Properties, and verify the forest functional level, as shown in Figure 8-8. Any user can verify the forest functional level in this way.

If the forest functional level is not at least Windows Server 2003, examine the properties of each domain to identify any domains for which the domain functional level is not at least Windows Server 2003. If you find such a domain, you must ensure that all domain controllers in the domain are running Windows Server 2003. Then, in Active Directory Domains And Trusts, right-click the domain and choose Raise Domain Functional Level. After you have raised each domain functional level to at least Windows Server 2003, right-click the root node of the Active Directory Domains And Trusts snap-in and choose Raise Forest Functional Level. In the Select An Available Forest Functional Level drop-down list, choose Windows Server 2003 and click Raise. You must be an administrator of a domain to raise the domain's functional level. To raise the forest functional level, you must be either a member of the Domain Admins group in the forest root domain or a member of the Enterprise Admins group.

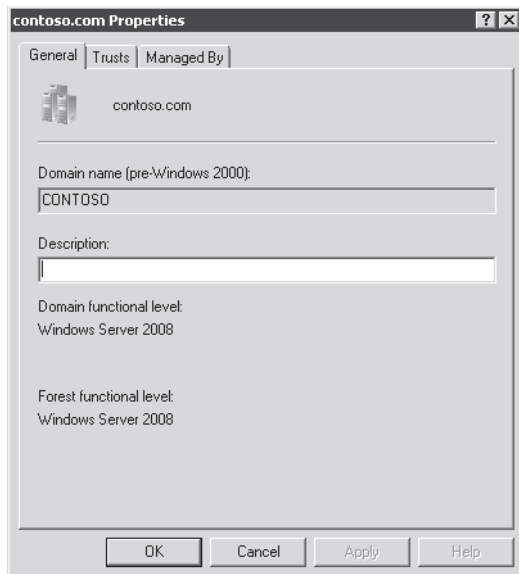


Figure 8-8 The forest Properties dialog box

Running *Adprep /rodcrep*

If you are upgrading an existing forest to include domain controllers running Windows Server 2008, you must run *Adprep /rodcrep*. This command configures permissions so that RODCs are able to replicate DNS application directory partitions. DNS application directory partitions are discussed in Chapter 9, “Integrating Domain Name System with AD DS.” If you are creating a new Active Directory forest and it will have only domain controllers running Windows Server 2008, you do not need to run *Adprep /rodcrep*.

You can find this command in the *cdrom\Sources\Adprep* folder of the Windows Server 2008 installation DVD. Copy the folder to the domain controller acting as the schema master. The schema master role is discussed in Chapter 10, “Domain Controllers.” Log on to the schema master as a member of the Enterprise Admins group, open a command prompt, change directories to the *Adprep* folder, and type **adprep /rodcrep**.

Placing a Writable Windows Server 2008 Domain Controller

An RODC must replicate domain updates from a writable domain controller running Windows Server 2008. It is critical that an RODC is able to establish a replication connection with a writable Windows Server 2008 domain controller. Ideally, the writable Windows Server 2008 domain controller should be in the closest site—the hub site. In Chapter 11, “Sites and Replication,” you’ll learn about Active Directory replication, sites, and site links. If you want the RODC to act as a DNS server, the writable Windows Server 2008 domain controller must also host the DNS domain zone.

Quick Check

- Your domain consists of a central site and four branch offices. A central site has two domain controllers. Each branch office site has one domain controller. All domain controllers run Windows Server 2003. Your company decides to open a fifth branch office, and you want to configure it with a new Windows Server 2008 RODC. What must you do before introducing the first RODC into your domain?

Quick Check Answer

- You must first ensure that the forest functional level is Windows Server 2003. Then, you must upgrade one of the existing domain controllers to Windows Server 2008 so that there is one writable Windows Server 2008 domain controller. You must also run *Adprep /rodcprep* from the Windows Server 2008 installation DVD.

Installing an RODC

After completing the preparatory steps, you can install an RODC. An RODC can be either a full or Server Core installation of Windows Server 2008. With a full installation of Windows Server 2008, you can use the Active Directory Domain Services Installation Wizard to create an RODC. Simply select Read-Only Domain Controller (RODC) on the Additional Domain Controller Options page of the wizard, as shown in Figure 8-9.

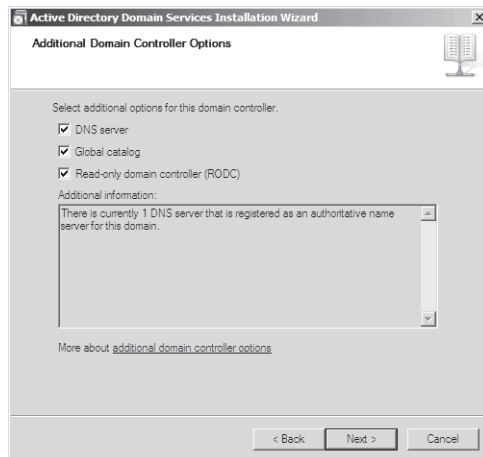


Figure 8-9 Creating an RODC with the Active Directory Domain Services Installation Wizard

Practice It Exercise 1, “Install an RODC,” in the practice at the end of this lesson walks you through the use of the Active Directory Domain Services Installation Wizard to create an RODC.

Alternatively, you can use the *Dcpromo.exe* command with the */unattend* switch to create the RODC. On a Server Core installation of Windows Server 2008, you must use the *Dcpromo.exe /unattend* command.

It is also possible to delegate the installation of the RODC, which enables a user who is not a domain administrator to create the RODC, by adding a new server in the branch office and running *Dcpromo.exe*. To delegate the installation of an RODC, pre-create the computer account for the RODC in the Domain Controllers OU and specify the credentials that will be used to add the RODC to the domain. That user can then attach a server running Windows Server 2008 to the RODC account. The server must be a member of a workgroup—not of the domain—when creating an RODC by using delegated installation.

MORE INFO Options for installing an RODC

For details regarding other options for installing an RODC, including delegated installation, see "Step-by-Step Guide for Read-only Domain Controllers" at <http://technet2.microsoft.com/windowsserver2008/en/library/ea8d253e-0646-490c-93d3-b78c5e1d9db71033.mspx?mfr=true>.

Password Replication Policy

Password Replication Policy (PRP) determines which users' credentials can be cached on a specific RODC. If PRP allows an RODC to cache a user's credentials, then authentication and service ticket activities of that user can be processed by the RODC. If a user's credentials cannot be cached on an RODC, authentication and service ticket activities are referred by the RODC to a writable domain controller.

A PRP of an RODC is determined by two multivalued attributes of the RODC computer account. These attributes are commonly known as the Allowed List and the Denied List. If a user's account is on the Allowed List, the user's credentials are cached. You can include groups on the Allowed List, in which case all users who belong to the group can have their credentials cached on the RODC. If the user is on both the Allowed List and the Denied List, the user's credentials will not be cached—the Denied List takes precedence.

Configure Domain-Wide Password Replication Policy

To facilitate the management of PRP, Windows Server 2008 creates two domain local security groups in the Users container of Active Directory. The first, named Allowed RODC Password Replication Group, is added to the Allowed List of each new RODC. By default, the group has no members. Therefore, by default, a new RODC will not cache any user's credentials. If there are users whose credentials you want to be cached by all domain RODCs, add those users to the Allowed RODC Password Replication Group.

The second group is named Denied RODC Password Replication Group. It is added to the Denied List of each new RODC. If there are users whose credentials you want to ensure are

never cached by domain RODCs, add those users to the Denied RODC Password Replication Group. By default, this group contains security-sensitive accounts that are members of groups including Domain Admins, Enterprise Admins, and Group Policy Creator Owners.

NOTE Computers are people, too

Remember that it is not only users who generate authentication and service ticket activity. Computers in a branch office also require such activity. To improve performance of systems in a branch office, allow the branch RODC to cache computer credentials as well.

Configure RODC-Specific Password Replication Policy

The two groups described in the previous section provide a method to manage PRP on all RODCs. However, to support a branch office scenario most efficiently, you need to allow the RODC in each branch office to cache user and computer credentials in that specific location. Therefore, you need to configure the Allowed List and the Denied List of each RODC.

To configure an RODC PRP, open the properties of the RODC computer account in the Domain Controllers OU. On the Password Replication Policy tab, shown in Figure 8-10, you can view the current PRP settings and add or remove users or groups from the PRP.

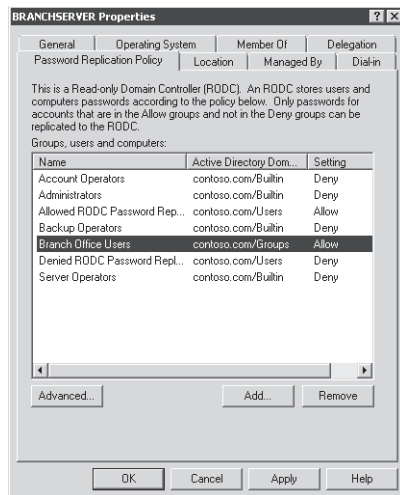


Figure 8-10 The Password Replication Policy tab of an RODC

Administer RODC Credentials Caching

When you click the Advanced button on the Password Replication Policy tab shown in Figure 8-10, an Advanced Password Replication Policy dialog box appears. An example is shown in Figure 8-11.

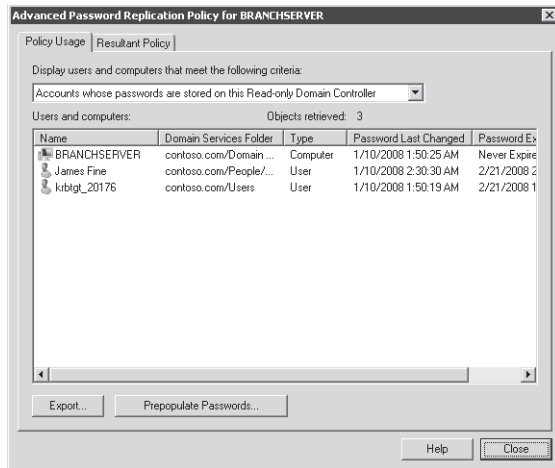


Figure 8-11 The Advanced Password Replication Policy dialog box

The drop-down list at the top of the Policy Usage tab enables you to select one of two reports for the RODC:

- Accounts Whose Passwords Are Stored On This Read-Only Domain Controller** Displays the list of user and computer credentials that are currently cached on the RODC. Use this list to determine whether credentials are being cached that you do not want to be cached on the RODC; modify the PRP accordingly.
- Accounts That Have Been Authenticated To This Read-Only Domain Controller** Displays the list of user and computer credentials that have been referred to a writable domain controller for authentication or service ticket processing. Use this list to identify users or computers that are attempting to authenticate with the RODC. If any of these accounts are not being cached, consider adding them to the PRP.

In the same dialog box, the Resultant Policy tab enables you to evaluate the effective caching policy for an individual user or computer. Click the Add button to select a user or computer account for evaluation.

You can also use the Advanced Password Replication Policy dialog box to prepopulate credentials in the RODC cache. If a user or computer is on the allow list of an RODC, the account credentials can be cached on the RODC but will not be cached until the authentication or service ticket events cause the RODC to replicate the credentials from a writable domain controller. By prepopulating credentials in the RODC cache for users and computers in the branch office, for example, you can ensure that authentication and service ticket activity will be processed locally by the RODC even when the user or computer is authenticating for the first time. To prepopulate credentials, click the Prepopulate Passwords button and select the appropriate users and computers.

Administrative Role Separation

RODCs in branch offices can require maintenance such as an updated device driver. Additionally, small branch offices might combine the RODC with the file server role on a single system, in which case it will be important to be able to back up the system. RODCs support local administration through a feature called *administrative role separation*. Each RODC maintains a local database of groups for specific administrative purposes. You can add domain user accounts to these local roles to enable support of a specific RODC.

You can configure administrative role separation by using the *Ddsmgmt.exe* command. To add a user to the Administrators role on an RODC, follow these steps:

1. Open a command prompt on the RODC.
2. Type **dsmgmt** and press Enter.
3. Type **local roles** and press Enter.

At the *local roles* prompt, you can type **?** and press Enter for a list of commands. You can also type **list roles** and press Enter for a list of local roles.

4. Type **add *username* administrators**, where *username* is the pre-Windows 2000 logon name of a domain user, and press Enter.

You can repeat this process to add other users to the various local roles on an RODC.

MORE INFO Improving authentication and security

RODCs are a valuable new feature for improving authentication and security in branch offices. Be sure to read the detailed documentation on the Microsoft Web site at <http://technet2.microsoft.com/windowsserver2008/en/library/ea8d253e-0646-490c-93d3-b78c5e1d9db71033.mspx>.

Chapter 9

Integrating Domain Name System with AD DS

Without the Domain Name System (DNS), using the Internet would not be easy. Oh, you could still use the Internet because the underlying technology for the Internet is really TCP/IP, but going to *http://207.46.198.248* isn't quite like going to *http://Technet.microsoft.com*, especially when you have to type the address in your browser. When you look up a new technology such as Windows Server 2008 in Windows Live Search and receive a collection of IP addresses hosting information as the result of your query, it doesn't inspire confidence that these sites are safe to navigate to. IP addresses do not mean much to humans whereas domain names do.

This is why users rely so much on DNS: it translates IP addresses into common terms or domain names that humans can relate to more easily. In fact, DNS is at the very core of the TCP/IP protocol, whether it is IPv4—the traditional, 32-bit addressing scheme—or IPv6, the new, 128-bit addressing scheme that is built into Windows Server 2008. Each time you set up a system in a network, it will be identified by its IP address or addresses. In a Windows Server 2008 network running Active Directory Domain Services (AD DS), each of the devices linked to the directory will also be linked to the DNS name resolution system and will rely on it to identify each of the services it interacts with.

For example, when you boot a computer that is part of a domain, a standard process takes place. This process begins by the identification of service location records (SRV) from a DNS server to identify the closest domain controller (DC). Then, after DNS has done its work, the authentication process between the computer and the DC can begin. However, without the name resolution for the SRV by DNS, it would be difficult for AD DS to authenticate a member computer.

Because it provides the translation of IP addresses to names, DNS enables programming standards through common names in applications. When programmers know they need a process that will support the discovery of a specific service, they use a common name for that service; then, when the customer implements the DNS service along with the new application, DNS will render the common name to the actual IP address assigned to the computer hosting the service.

In addition, because it is a technology designed to manage naming on the Internet, DNS is one of the technologies contained within Windows Server 2008 that enables you to extend the authority of your network to the outside world. Like Active Directory Certificate Services

(AD CS), Active Directory Rights Management Services (AD RMS), Active Directory Lightweight Directory Services (AD LDS), and Active Directory Federation Services (AD FS), DNS is integrated with AD DS, but it can also run independently in a perimeter network and beyond. (See Figure 9-1.) When it does so, it enables other organizations and individuals to locate you from anywhere in the world. When they find you, they can interact with you or the applications you might share with customers, partners, mobile users, and anyone else through some form of electronic communication.

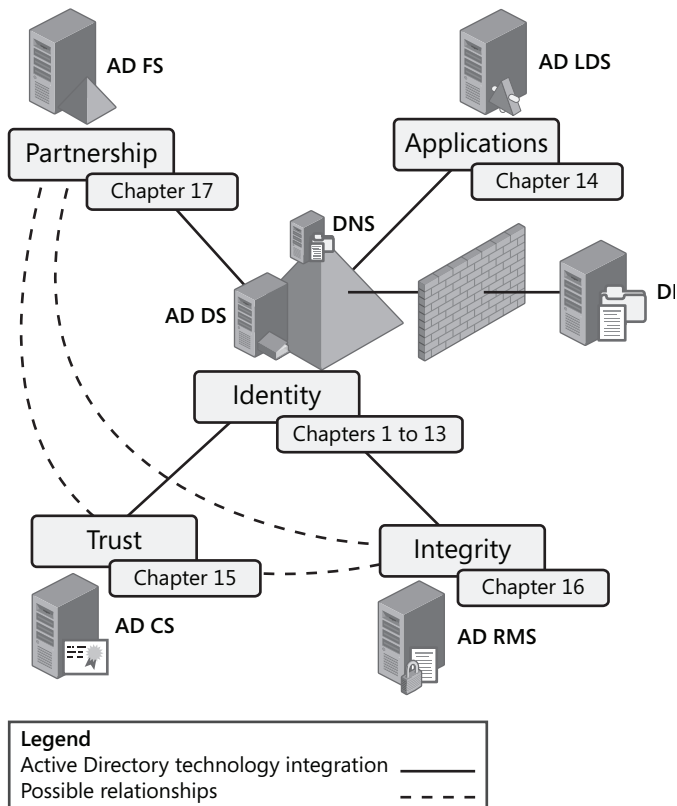


Figure 9-1 DNS extends your organization's authority beyond the borders of your internal network

Whether it communicates on the Internet or in your internal network, DNS always relies on TCP/IP port 53. All clients and servers are tuned to this port to locate and identify information about the computer names they need to interact with.

The naming structure supported by DNS is hierarchical. Names begin with a root and extend from the root when additional tiers are added to the hierarchy. The actual root of the DNS hierarchy is the dot (.) itself. However, this dot is not used in Internet naming. Commonly, standard root names are registered on the Internet and include names such as .com, .biz., .net,

.info, .name, .ms, .edu, .gov, .org, and so on. Organizations can link to the Internet through the binding of a common name with the root name. For example, *Microsoft.com* is two levels down from the root name but three levels down from the actual DNS root, as shown in Figure 9-2. *TechNet.microsoft.com* is three levels down from the name but four from the DNS root and so on. AD DS relies on this hierarchy to create the domain structure of a forest.

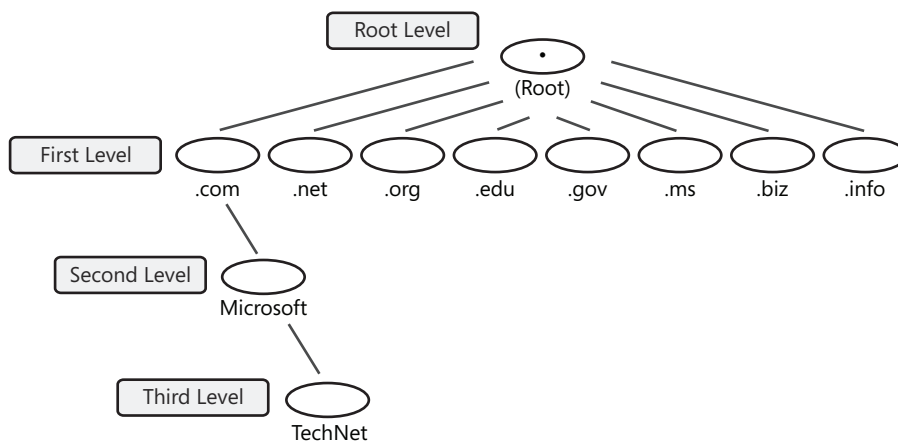


Figure 9-2 The DNS hierarchy of the Internet

DNS and IPv6

In Windows Server 2008, DNS has been updated to integrate with IPv6. Unlike IPv4, which is composed of four octets of binary digits to form the 32-bit IP address, IPv6 uses eight 16-bit pieces to form the 128-bit IP that is usually displayed in hexadecimal format. For example, FE80:: refers to the autogenerated link-local IPv6 address Windows Vista or Windows Server 2008 will assign to your computer if you rely on the Dynamic Host Configuration Protocol (DHCP) and there is no available DHCP server to respond with an actual address. The FE80:: address is the same as the Automatic Private IP Addressing (APIPA) address your system will generate if the same thing happens with an IPv4 address allocation.

In IPv6, each time a 16-bit address piece is composed of all zeros, you can concatenate the address and represent it with two colons (::). The two colons will represent any number of 16-bit sections that are composed of all zeros as long as they are contiguous. This facilitates writing out IPv6 addresses; otherwise, IPv6 notation could become quite complex.

Like IPv4, IPv6 provides several types of addresses:

- **Link-local** Addresses that enable direct neighbors to communicate with each other. Any computer on the same network segment will be able to communicate with any other

by using this address type. This is the address type assigned by default when IPv6 is turned on but does not use a static address and cannot communicate with a dynamic address provider such as a DHCPv6 server. These addresses are similar to the 169.254.0.0/16 addresses used by the APIPA process.

- **Site-local** Addresses that support private address spaces and you can use internally without having your own IPv6 address allocation. Site-local addresses can be routed, but should never have a routed connection to the Internet. They are similar to the 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16 addresses organizations use internally with IPv4.
- **Global unicast** Addresses that are entirely unique and can be used on the Internet to identify an interface. These addresses are routable on the Internet and enable direct communication to any device. These are comparable to the public IPv4 addresses organizations use on the Internet today.

The boon of IPv6 is the sheer number of addresses it provides. With the world population booming, the number of services and devices requiring IP addresses increasing, and the number of IPv4 addresses dwindling, it is time for the IP infrastructure of the Internet to evolve to the next level. By providing 340 billion billion billion billion—or 2^{128} addresses—IPv6 should support the next stage of the Internet for a long time. All you have to do is compare it to the 4 billion IPv4 addresses to see the difference.

Table 9-1 outlines the most common IPv6 address types.

Table 9-1 Common IPv6 Address Types

Address Type	Format	Description
Unspecified	::	Indicates the absence of an address. Comparable to 0.0.0.0 in IPv4.
Loopback	::1	Indicates the loopback interface and enables a node to send packets to itself. Comparable to 127.0.0.1 in IPv4.
Link-local	FE80::	Local network browsing address only. Comparable to APIPA or addresses in the 169.254.0.0/16 range in IPv4. Unroutable by IPv6 routers.
Site-local	FEC0::	Site-level internal address space. Routable but not to the Internet. Comparable to addresses in the 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16 ranges in IPv4.
Global unicast	All others	Unique addresses assigned to specific interfaces.

To comply with Internet standards and support the move to IPv6, DNS in Windows Server 2008 has been updated to support the longer address form of the IPv6 specification. IPv6 is installed and enabled by default in both Windows Vista and Windows Server 2008. This means that you can use this technology, at least internally, with little risk. It will be some time before all the elements that require an IPv6 connection to the Internet—intrusion detection

systems, firewalls, anti-spam filtering, and so on—have been upgraded to support secure IPv6 transmissions.

MORE INFO IPv6

For more information on IPv6, go to <http://www.microsoft.com/technet/network/ipv6/ipv6rfc.mspc>.

The Peer Name Resolution Protocol

Because they fully support IPv6, Windows Server 2008 and Windows Vista also include a secondary name resolution system called Peer Name Resolution Protocol (PNRP). Unlike DNS, which relies on a hierarchical naming structure, PNRP relies on peer systems to resolve the location of a computer system. Basically, PNRP is a referral system that performs lookups based on known data. For example, if you need to look up Computer A and you are near Computers B and C, your system will ask Computer B if it knows Computer A. If Computer B says yes, it will provide you with a link to Computer A. If not, your system will ask Computer C if it knows Computer A and then use the same process as with Computer B. If neither Computer B nor Computer C knows Computer A, your system will send a request to other computers near it until it finds one that knows of Computer A.

PNRP includes several features that are different from the DNS service:

- It is a distributed naming system that does not rely on a central server to locate objects. It is almost entirely serverless, but in some instances, servers are required to develop the name resolution process by themselves. Windows Server 2008 includes PNRP server components as an add-on feature.
- PNRP can scale to billions of names, unlike the DNS service, which will host only a small number of names and will then rely on another DNS server to locate the names over which it is not authoritative.
- Because it is distributed and relies on clients as much as servers, PNRP is fault tolerant. Several computers can host the same name, providing multiple paths to that name.
- Name publication is instantaneous, free, and does not require administrative intervention in the way DNS does.
- Names are updated in real time, unlike DNS, which relies heavily on caching to improve performance. Because of this, PNRP does not return stale addresses the way a DNS server, especially an earlier, nondynamic DNS server, can.
- PNRP also supports the naming of services as well as of computers because the PNRP name includes an address, a port, and a potential payload such as a service's function.

- PNRP names can be protected through digital signatures. Protecting the names in this way ensures that they cannot be spoofed or replaced with counterfeit names by malicious users.

To provide resolution services, PNRP relies on the concept of a cloud. Two different clouds can exist. The first is the global cloud and includes the entire IPv6 global address scope, which encompasses the entire Internet. The second is a link-local cloud and is based on the link-local IPv6 address scope. Local links usually represent a single subnet. There can be several link-local clouds but only a single global cloud.

Just as the world has not fully moved to IPv6 yet, it also hasn't moved to PNRP and continues to rely on DNS to provide name resolution services. However, PNRP is an important new technology that will have a greater and greater impact on Internet operation as organizations move to IPv6.

MORE INFO PNRP

For more information on PNRP, go to <http://technet.microsoft.com/en-us/library/bb726971.aspx>.

DNS Structures

DNS has been around since the Internet was first developed and has evolved with it. Because of this, the DNS service in Windows Server 2008 can provide a number of roles. There are three possible types of DNS servers:

- **Dynamic DNS servers** Servers that are designed to accept name registrations from a wide variety of devices through dynamic updates are deemed to be dynamic DNS (DDNS) servers. DDNS is designed to enable devices—clients and servers—to self-register to the DNS server so that other devices can locate them. When the DNS service runs on a DC and is integrated with the directory service, it runs in DDNS mode, enabling computers that use DHCP to register their own names within it automatically. This enables AD DS to locate the client when it needs to send it management data such as Group Policy objects (GPOs). DDNS servers are read-write servers, but they accept registrations from known entities only.

Exam Tip Note that the exam does not include direct references to dynamic DNS. It will, however, refer to dynamic updates as well as to Active Directory–integrated DNS zones. Any time a DNS server is updated automatically through authorized clients, it is a DDNS. Keep this in mind when taking the exam.

- **Read-write DNS servers** Earlier DNS servers that are not running in dynamic mode but that will accept writes from known sources such as authorized operators are deemed read-write DNS servers. The most common type of read-write DNS server is the primary

DNS server. Primary DNS servers are usually deployed in perimeter networks and are not integrated with AD DS.

- **Read-only DNS servers** DNS servers that hold a read-only copy of DNS data that originates from another location are deemed read-only DNS servers. In Windows Server 2008, there are two types of read-only DNS servers. The first is the secondary DNS server. Secondary DNS servers are linked to primary DNS servers and will accept and host DNS data provided by the primary parent server. They make data available locally but cannot be modified because they support only one-way replication from the primary. A second type of read-only DNS server in Windows Server 2008 is the DNS server that runs on a read-only DC (RODC). These servers run primary read-only zones when integrated with RODCs.

Using these three types of DNS servers, you can construct a name resolution strategy that meets all your naming requirements. (See Figure 9-3.) For example, you could pair DDNS servers with every domain controller in your network because the DNS data is usually integrated with the directory store. Because it is contained in the directory store, the DNS data is replicated to every DC in a domain and sometimes in a forest through the same mechanism that replicates directory traffic. This means each DC has a local copy of DNS data. Installing the DNS service on the DC automatically gives it access to this data and can provide local rather than remote name resolution services, avoiding wide area network (WAN) traffic. In addition, you can use the RODC DNS service in read-only mode in unsecured locations within your network, locations that warrant local services but do not have local administrative staff. You can also use the standalone primary DNS service in perimeter networks. These servers contain few records but support access to any application or service you host in your perimeter. Last, you could use read-only secondary DNS servers in unsecured locations facing the Internet.

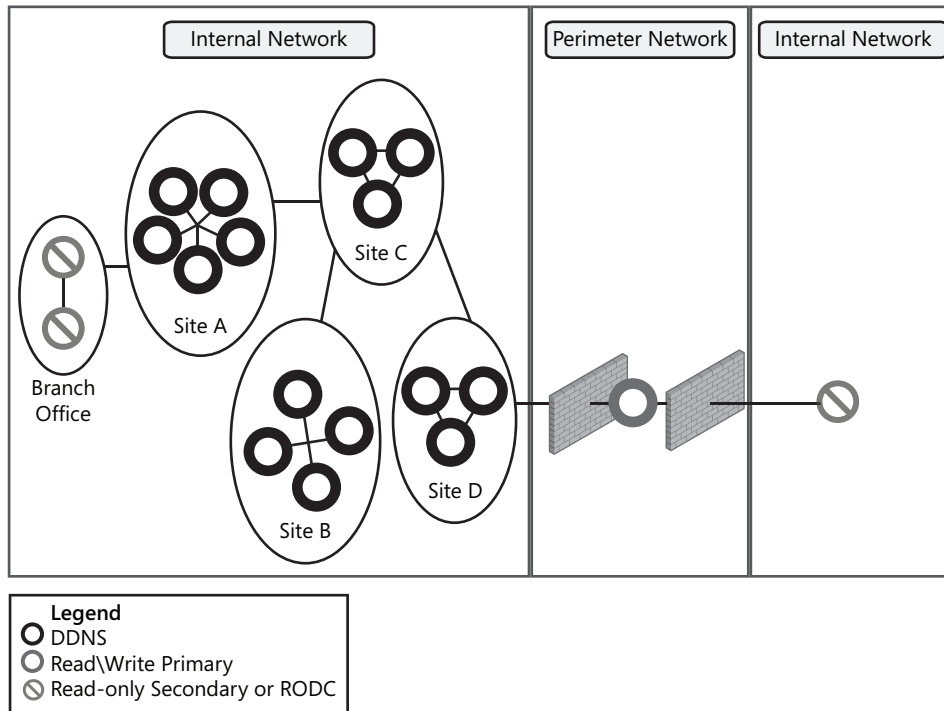


Figure 9-3 DNS server placement in a Windows Server 2008 network: DDNS follows DCs, primaries are protected, and RODCs are internal, whereas secondaries are external

MORE INFO Domain Name System

For more information on DNS, go to <http://technet2.microsoft.com/windowsserver2008/en/servermanager/dnsserver.mspx>.

The Split-Brain Syndrome

One of the most basic tenets of internetworking is the segregation of the internal network from the Internet. Small and large organizations alike will endeavor to protect their internal network through a variety of systems and technology. The most common protection mechanism is the firewall, which protects your network by blocking entry of undesirable traffic through the control of TCP/IP ports. Accepted ports are open and all unacceptable ports are closed. It's as simple as that.

Similarly, when you work with Windows Server 2008 and especially with AD DS, you will need to work with two namespaces. Because AD DS directories are based on the DNS hierarchical naming system, you must use a properly formed DNS name, often called a fully qualified domain name (FQDN), to name your directory forests and the domains they contain. Frequently, organizations use the same name they use to represent themselves on the Internet.

For example, this book suggests names such as *contoso.com* or *woodgrovebank.com* as potential names for your internal networks. This is by no means a best practice. This book uses these names because they are legally acceptable names that Microsoft Press is allowed to use to represent fictitious organizations. However, using the same name internally for your AD DS directory structure as you use for your external exposure to the Internet means you must implement a split-brain DNS service.

That's because you need to maintain two namespaces for two purposes all across a firewall. Nothing could be more complex. Your users must be able to locate both internal and external resources that rely on the same name. If Contoso, Ltd., used *contoso.com* for both its internal and its external namespaces in real life, its DNS administrators would need to manage the separation manually between internal and external name resolution mechanisms.

However, if Contoso used *contoso.com* exclusively for its external presence and used a corresponding name with a different extension, for example, *.net*, for its internal AD DS namespace, the DNS administrators would have to do nothing to segregate the two namespaces. The very fact that they use different roots automatically segregates the names and the two DNS services that would be used to support them. The only thing that needs to transit through the firewall is standard name references you would normally use for any name that is not located within your network.

In addition, it is very easy for Contoso to buy and maintain all the possible combinations of its Internet name, including common roots such as *.com*, *.net*, *.info*, *.ms*, *.ws*, and more. This way, Contoso knows it can use any of the names it owns for any forest implementation, production, testing, development, or staging or for whichever purpose it needs it and never conflict with anyone else even if it faces a merger or an acquisition.

Issues that commonly arise around this topic are often based on the ownership of the DNS service. Traditionally, network operators own previous DNS services and, very often, these DNS services are maintained in environments that are not Microsoft Windows-based. However, Windows and, especially, AD DS are designed to rely very tightly on the Windows DNS service. Although it is possible to use Windows with non-Windows DNS servers, it is not recommended because it requires so much more work. When you use the Windows DNS service and integrate it with your AD DS service, everything becomes automatic. When you don't, everything remains manual and, very often, you'll find that specific components don't work because the manual configuration was not completed or was misconfigured by non-Windows system administrators.

If you are in this situation and you must run two DNS technologies, the best and ideal network configuration is to use a whole-brain approach and rely on two different namespaces, integrate the internal namespace with Windows DNS servers running on DCs, and simply link the two namespaces through standard DNS resolution mechanisms. This will provide you with the implementation that will require the least amount of administrative effort and ensure that all services work at all times. (See Figure 9-4.)

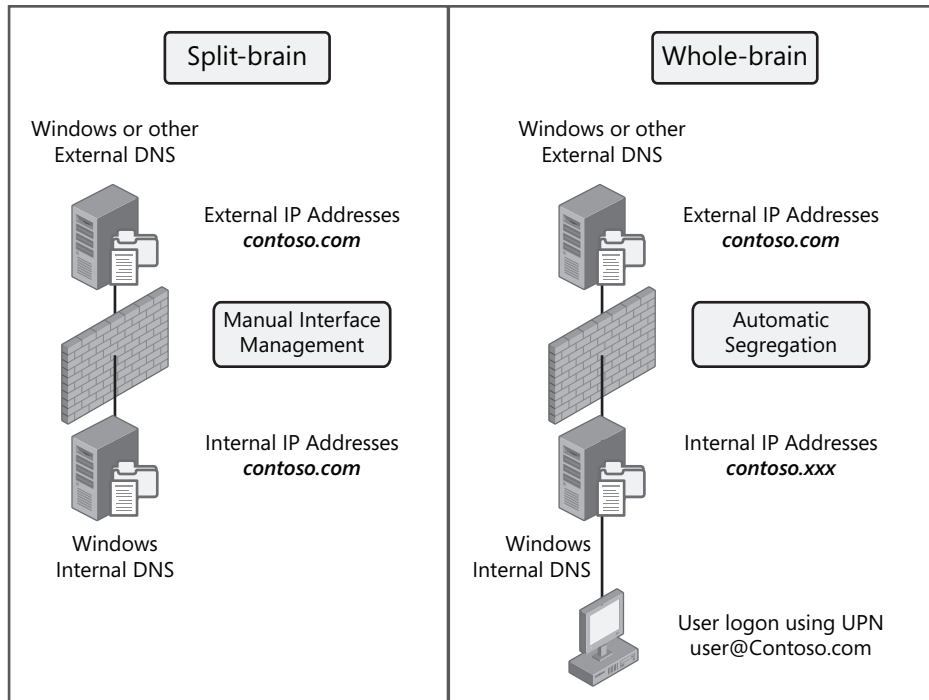


Figure 9-4 Split-brain vs. whole-brain DNS structures

Further, you needn't worry about users. If you are using a different namespace internally, but you want them to log on with the external network name, for example, *contoso.com*, just add it as the preferred user principal name (UPN) suffix in your directory. DNS will be simpler to manage, your internal network will be protected from external access, and your users won't know the difference!

MORE INFO Split-brain DNS

For more information on split-brain DNS setups, go to <http://www.microsoft.com/serviceproviders/resources/techresarticlesdnssplit.msp>.

Exam objectives in this chapter:

- Configuring Domain Name System (DNS) for Active Directory
 - Configure zones.
 - Configure DNS server settings.
 - Configure zone transfers and replication.

Before You Begin

To complete the lessons in this chapter, you must have done the following:

- Installed Windows Server 2008 on a physical or virtual computer that should be named SERVER10 and should be a standalone server. This computer will host the DNS server and DC service you will install and create through the exercises in this chapter. Assign an IPv4 address from one of the private ranges, for example, 192.168.x.x, and map its DNS server address to its own address.
- Installed Windows Server 2008 on a physical or virtual computer that should be named SERVER20 and should be a standalone server. This computer will host the DNS server and DC service you will install and create through the exercises in this chapter. Assign an IPv4 address from one of the private ranges, for example, 192.168.x.x, and map its DNS server address to the address you assigned to SERVER10.
- Installed Windows Server 2008 on a physical or virtual computer that should be named SERVER30 and should be a standalone server. This computer will host the DNS server and DC service you will install and create through the exercises in this chapter. Assign an IPv4 address from one of the private ranges, for example, 192.168.x.x, and map its DNS server address to the address you assigned to SERVER10.

We strongly recommend using virtual machines (VMs) in support of the exercises. The DC and DNS server roles are ideal for virtualization through either Microsoft Virtual Server 2005 R2 or Windows Server 2008 Hyper-V.

Real World

Danielle Ruest and Nelson Ruest

In late 2002, we were putting finishing touches to our second book: *Windows Server 2003, Best Practices for Enterprise Deployments* for McGraw-Hill Osborne. This book was based on our experiences with customers in designing and deploying Windows 2000-based Active Directory (AD) structures. One feature that intrigued us the most was the new application directory partition feature in Microsoft Windows Server 2003. According to the documentation provided with the beta versions, application directory partitions would be used to store DNS data within the directory and control their replication scope.

As our customers would create best-practices forests, using a forest root domain and a single, global child production domain, we discovered that when you created the forest root domain, DNS data was properly located within the forest root domain partition, but when you created a child domain, the data would not be stored automatically within the child domain partition. This caused a serious problem with DNS data. All our customers would use a two-DC forest root to keep it as secure as possible and to control access to forest root administration tightly. Because of this, the forest root DCs would always be located within central headquarter sites. The child production domain, however, would be highly distributed and include domain controllers within each remote site that had more than a certain number of users.

Because DNS data for the child domain was actually included in the forest root domain partition and not in the child partition, each client had to perform DNS lookups over the WAN to contact the forest root DCs. However, if DNS data was to be stored in the directory and made available to DCs, it should be in the local DC, not in a remote DC.

We discovered that we could change the replication scope of the child domain DNS data after the directory service was deployed, but we have always been proponents of doing things right in the first place, not correcting them afterward. This meant we needed to find a way to make sure the DNS data would be stored in the proper location during installation rather than later.

We contacted the Microsoft Active Directory development team and reported this DNS behavior as a bug, and they agreed that this should be corrected at installation, not afterward. Further research demonstrated that because a child domain namespace is an extension of the root domain's namespace, the child domain name would resolve properly during the verification checks the Active Directory Installation Wizard performed. Because of this, the wizard would store the data within the forest root domain. In fact, the wizard was behaving properly; we just didn't give it enough information.

Further investigation revealed that if you created a manual DNS delegation before creating the child domain, the wizard would locate the data properly within the child domain partition—the manual delegation would point to a server that did not exist yet because the child domain was not created. For example, if you had a root domain named *treymresearch.net* and a child domain that would be named *intranet.treymresearch.net*, you would point the delegation to a server named *servername.intranet.treymresearch.net*. Because no server of that name existed until the child domain was actually created, the delegation would contain dummy data and would be called a dummy DNS delegation. When the wizard would run, it would find this server in DNS and try to use it to resolve the child domain's DNS name. The resolution would fail and force the wizard to install DNS during the creation of the domain and create the appropriate DNS data partition.

The Active Directory Domain Services Installation Wizard now properly creates delegations for child domains. Many AD implementations based on Windows Server 2003 did not locate DNS data in the proper partition during installation, and only IT administrators who knew how to use the dummy delegation before creating a child domain were aware of the issue. Windows Server 2008 has resolved this problem.

Lesson 1: Understanding and Installing Domain Name System

Domain name resolution is a complex process that relies on a naming hierarchy to match IP addresses, both IPv4 and IPv6, to system names. DNS name resolution also supports the identification of service locations. This is how the AD DS logon process works. In fact, DNS plays an essential role in this process and, because of this, services such as those provided by AD DS would simply not be possible without the DNS service.

To do this, the DNS service relies on name records. Records can be inscribed manually, such as in a primary DNS server that provides read-write services. However, writes are supported only from administrators, or they can be inscribed automatically such as with dynamic DNS servers that accept name records from devices. Smart devices such as computers running editions of Windows 2000, Windows XP, Windows 2003, Windows Vista, or Windows Server 2008 can register their own names within a DDNS, but devices running earlier operating systems such as Microsoft Windows NT cannot. Former devices will rely on the DHCP to perform the inscription for them; however, this is a less secure implementation of a DDNS infrastructure.

DNS contains a host of record types that can be used to provide name resolution for specific service types or specific computer types. In addition, these records are stored within DNS zones—special placeholders that provide a given name resolution functionality for a specific namespace.

Understanding the various components of the Windows Server 2008 DNS service is critical to understanding how it works and how it should be used.

MORE INFO DNS in Windows Server 2008

For more information on DNS in Windows Server 2008, go to <http://technet2.microsoft.com/windowsserver2008/en/servermanager/dnsserver.mspx>.

After this lesson, you will be able to:

- Understand when to use DNS.
- Install DNS.
- Locate and view the DNS installation.

Estimated lesson time: 70 minutes

Understanding DNS

The first thing to understand when working with DNS is how it works to resolve a name. You already know that DNS relies on a hierarchy of servers because a DNS server cannot hold all

possible name records within itself. Because of this, the DNS service relies on name referrals to perform name resolution. (See Figure 9-5.)

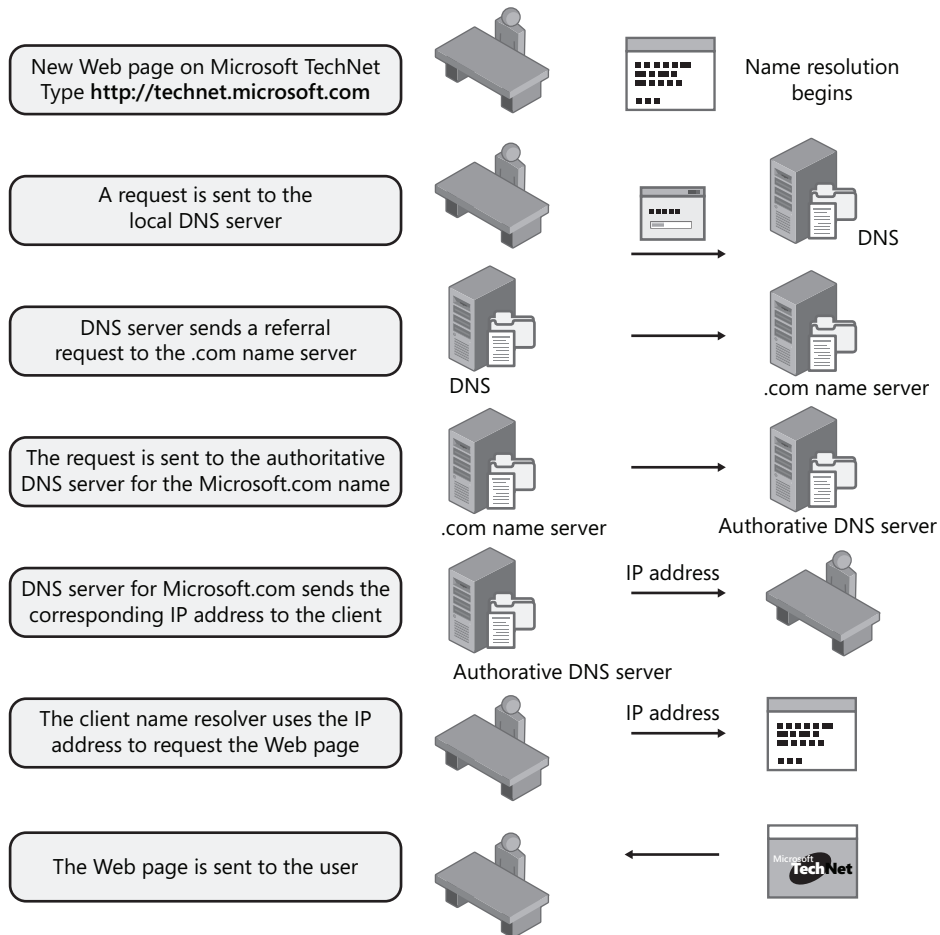


Figure 9-5 The DNS name resolution process

Here's how name resolution works:

1. You try to look up a Web page on the Microsoft TechNet Web site. To do so, you type `http://technet.microsoft.com` in the address bar of your browser and press Enter. That's when name resolution begins.
2. Your computer sends out a request to its local DNS server or at least to one of the servers listed in its IP configuration settings for the name.
3. If this server does not include the name in its own database or cache, it sends a referral request to the name server. Because the Microsoft site name ends in `.com`, the DNS server sends the referral to the `.com` name server.

4. The .com name server is the authority for all names that end in the .com suffix. This server knows the location of all DNS servers that are the final authorities for a particular name ending with .com. In this case, it sends the request to the authoritative DNS server for the *microsoft.com* name.
5. The DNS server for *microsoft.com* sends the corresponding IP address for the requested page to the client computer.
6. The name resolver on the client uses the IP address to request the actual page from its Internet provider.
7. If the page is not already in the local cache of the Internet provider, it requests the actual page and sends it to the client.

This name resolution procedure occurs within seconds, and the Web page appears almost as fast as you type it, depending on the speed of your connection and the current load of the requested server. That's what happens when you look at the green progress bar at the bottom of your browser. The actual progress also includes downloading content such as text and graphics to your own computer.

DNS is a system that does not and cannot work alone. It must rely on other servers to operate. In addition, the DNS service includes a terminology of its own. Table 9-2 outlines the most common terms you will encounter when working with DNS.

Table 9-2 DNS Terms and Concepts

Term	Description
Active Directory Integrated (ADI) zone	When a DNS zone is integrated with Active Directory, it is hosted in the AD DS database, <i>NTDS.dit</i> , and is replicated throughout the directory with other directory data.
Aging	DNS records are associated with an age or a Time to Live (TTL). When the record lasts beyond its age, it is no longer valid and can result in false positives, giving users the impression they are going to a specific location when the location is no longer valid.
Application directory partitions	When DNS data is stored within AD DS directory databases, it is replicated by default with the directory data it is associated with. However, you can define a custom replication scope for DNS data. For example, DNS data that belongs to a root domain for a forest must be made available to the entire forest whereas DNS data for a specific domain is really required only for that domain. You control DNS data replication scopes through application directory partitions.

Table 9-2 DNS Terms and Concepts

Term	Description
DDNS	This is a DNS service that can be automatically updated by the clients that rely on it. In Windows Server 2008, you install DDNS whenever you choose to install the DNS service with AD DS. Because all the clients in a DDNS implementation have AD DS accounts, they are deemed secure clients and are authorized to update the DNS server with their record information.
DNS Notify	Traditional or former DNS servers manage data in local files. These files are located on the primary server. They are then transferred through a polling and zone transfer mechanism to read only secondary servers. However, large zones will often require frequent record updating. This could lead to incorrect records located within the secondary server. To correct this situation, DNS uses a special notification process that notifies slave servers that an update is available, which then prompts a zone transfer to the read-only servers.
Domain DNS zone	This is the zone that contains the records for a particular domain, either a root or a child domain, within an AD DS forest structure.
Forest DNS zone	This is the zone that contains the records that pertain to an entire forest within an AD DS forest structure.
Forward lookup	DNS supports two types of lookups: forward or reverse. A forward lookup relies on a client providing an FQDN to the server. The server then matches this FQDN to the corresponding IP address.
Forward lookup zone	This comprises DNS containers—databases or text files—that include name resolutions for forward lookups.
Forwarders	DNS servers have two mechanisms for name resolution: forwarders or root hints. DNS servers that provide name resolution services to the internal network will often rely on forwarders to forward any request they cannot resolve on their own to a trusted external DNS server. Windows Server 2008 also includes the ability to rely on conditional forwarders or forwarders that are used only when specific conditions are met in a request. For example, if the name is for an internal domain, but not one managed by this server, it can automatically forward the request to the internal name server for that domain.
GlobalNames Zone (GNZ)	NetBIOS names are single-label names that do not use the FQDN format. For example, down-level computer names are single-label names. Traditionally, these names are managed by Windows Internet Name Service (WINS). In an effort to remove this prior service from a Windows-based network, Microsoft has implemented the GlobalNames Zone in DNS in Windows Server 2008. GNZs can contain single-label names and replace WINS in a Windows-based network.

Table 9-2 DNS Terms and Concepts

Term	Description
Legacy DNS	Nondynamic DNS servers that rely on manual updating of zone records are deemed legacy DNS servers. Because it complies with the set of request for comments (RFC) that define and standardize the DNS protocol on the Internet, Windows Server 2008 can support former DNS services as well as the dynamic DNS service required by AD DS. Legacy DNS servers host either primary or secondary zones.
Name recursion	Name resolutions can be either iterative or recursive. In an iterative request, each DNS server holds only part of the answer for a query and must rely on other DNS servers to complete the query. In a recursive query, the DNS server will hold the complete answer and provide it to the requester. Because of record aging, it is possible for a recursive query to respond with an erroneous IP address.
Primary zones	These are zones that contain read-write information for a particular domain. Primary zones are stored on nondynamic or dynamic DNS servers. When stored on nondynamic DNS servers, primary zones are contained within text files and are edited manually by an administrator. When stored on DDNS servers, primary zones are contained within Active Directory and are updated either automatically by each record holder or manually by an administrator.
Resource records	These are the name records contained within DNS databases. Resource records usually link an IP address with an FQDN.
Reverse lookup	DNS supports two types of lookups: forward or reverse. A reverse lookup relies on a client providing an IP address and requesting the FQDN that corresponds to the address.
Reverse lookup zone	This zone comprises DNS containers—databases or text files—that include name resolutions for reverse lookups for a particular domain.
Root hints	DNS servers have two mechanisms for name resolution: forwarders or root hints. DNS servers that provide name resolution services to the internal network but also have a direct connection to the Internet can rely on root hints to locate authoritative servers for root names such as .com, .org, .net, and so on in the Internet and provide resolution services to internal clients. By default, Windows Server 2008 DNS servers rely on root hints for external name resolution. These hints are regularly updated through Microsoft Windows Update. Root hints are contained within a special file named <code>Cache.dns</code> , which can also be used to reset root hints in the event of issues with the external name resolution process.

Table 9-2 DNS Terms and Concepts

Term	Description
Round robin	DNS services can be used to provide some form of high availability. This is done by creating multiple records for the same resource, each with a different IP address. When queried, the DNS server will provide the first address, then the second address, then the third, and so on, balancing the load between multiple servers that host the same service. For example, Microsoft Exchange Server 2007 Edge Transport Server—servers that face the Internet to accept and send internal e-mail—rely on the round robin process to provide e-mail load balancing.
Secondary zone	A secondary zone is a read-only zone obtained from a primary DNS server. Secondary zones provide local DNS resolution in highly distributed networks.
Server scavenging	A feature that was introduced with the dynamic DNS service released with Windows 2000 Server. Because records have an age or time to live, they can become stale when they extend beyond their expected duration. Server scavenging will scour the DNS database to locate records that have aged beyond their usefulness and automatically remove them.
Single-label names	NetBIOS names that do not use the FQDN format. For example, down-level computer names are single-label names. These names include 16 characters and do not support special characters such as dots. Only the first 15 characters of a single-label name can be used because the sixteenth character is reserved by the system to complete the name. Traditionally, these names are managed by WINS. In Windows Server 2008, you can rely on the GNZ in DNS to replace WINS.
Start of Authority (SOA) record	This is a special DNS record that outlines domain information such as the update schedule for the records it contains, the intervals other DNS servers should use to verify updates, and the date and time of the last update as well as other information such as contacts for the domain and so on. Only one SOA record can be contained within a specific zone file. Each zone file should contain a particular SOA record.
Stub zone	This is a special zone type that contains only the records of other DNS servers that maintain the actual zone itself. This can speed name resolution and reduce the likelihood of errors because stub zones are used as referrals to other, authoritative DNS servers for a zone.
TTL	Each DNS record is given a TTL value. This value determines the valid duration of the record. When it expires, the record can be deleted through scavenging. However, if the record is still valid before its TTL value expires, you can renew the record and, therefore, renew its TTL value.

Table 9-2 DNS Terms and Concepts

Term	Description
Zone delegations	Delegations are used to help you manage different namespace sections better. For example, Microsoft might want to delegate different sections of its namespace, notably the MSDN or TechNet sections of <i>Microsoft.com</i> , to have them administered by other divisions in the company. When managing DNS namespaces in AD DS, you must delegate new domain-based zones when you create the domain; otherwise, the zone will be managed at the forest level and not at the domain level as it should be. In Windows Server 2003, this delegation had to be created manually before creating the domain. In Windows Server 2008, the Active Directory Domain Services Installation Wizard will perform the delegation automatically when you create a child domain.
Zone scavenging	Scavenging scours the DNS server to remove stale or outdated records whose age has gone beyond their TTL value. Zone scavenging applies when scavenging is applied to a single zone as opposed to the entire server.
Zone transfers	These are the transactions DNS servers use to replicate information from one server to another. Full zone transfers transfer the entire content of a zone to one or more other DNS servers. Incremental transfers send only a subset of the data. Traditionally, full transfers are referred to as Asynchronous Full Transfer (AXFR) whereas incremental transfers are dubbed Incremental Zone Transfer (IXFR). Windows Server 2008 also supports secure zone transfers, which are performed through AD DS multimaster replication.

The Windows Server 2008 DNS service supports three zone types, as shown in Figure 9-6:

- **Primary Zone** Zones that can be integrated with AD DS or that can be of the former, standard type. These zones are authoritative for the namespace they contain. Primary zones are read-write zones except when located on RODCs.
- **Secondary Zone** Zones that are of the former, standard type and are only a replica of the data maintained by a primary or authoritative server for a namespace. When you create a secondary zone, you must tell DNS the address of the primary zone or source of the zone data.
- **Stub Zone** Zones that are nothing but pointers to other, authoritative servers for the namespace they maintain. Once again, when you create a stub zone, you must specify a list of server(s) that are authoritative for the namespace.

Each zone type can be stored either in a text file or within an Active Directory directory store partition.

Exam Tip Keep these zone types in mind for the exam. You can change from one zone type to another in DNS, but remember that the most useful zone type is the primary zone. This is the type used by AD DS when you integrate the DNS service with it.

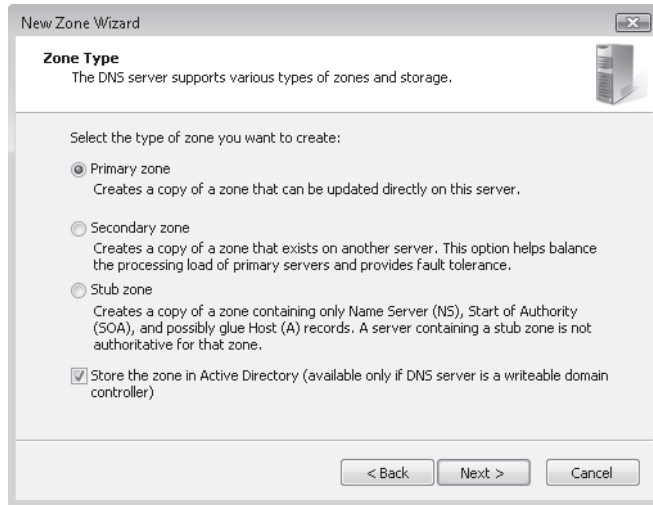


Figure 9-6 The New Zone Wizard enables you to create any of the three supported zones

Zones are containers that include information about the objects they manage. This information is in the form of records. DNS can contain several types of records. Table 9-3 outlines the most common record types used in DNS in Windows Server 2008.

Table 9-3 DNS Record Types in Windows Server 2008

Record Type	Usage
Alias (CNAME)	Used to create an alternate record or alias DNS name for a name that is already specified as another record type in a specific zone. This is also known as a canonical name (CNAME). For example, if you want to create a record such as <i>intranet.contoso.com</i> to point to a server or server farm hosting Microsoft Office SharePoint Server, you would create it as an alias record. This facilitates usage by providing a more functional name than the server name.
Host (A or AAAA) records	The most common record type in DNS. They represent computer objects in the namespace and are used to resolve a specific IP address to a device.

Table 9-3 DNS Record Types in Windows Server 2008

Record Type	Usage
Mail exchanger (MX)	Routes e-mail messages to a specific namespace. For example, the MX record for <i>contoso.com</i> would indicate that all e-mail directed to <i>contoso.com</i> should pass through the host or hosts identified by this record.
Pointer (PTR)	Points to a specific location within the namespace. PTR records are usually used to provide reverse lookup capabilities within the namespace.
Service location (SRV)	Indicates the location of a specific TCP/IP service. For example, if you want to use Microsoft Office Communications Server, you must create a session initiation protocol (SIP) service location record to indicate to all the devices that rely on this service where it is situated in your network. Similarly, AD DS creates several service location records in support of the logon or the Group Policy distribution processes. Service location records usually consist of the IP address for the server as well as the TCP/IP port on which the service is available.

The records in Table 9-3 provide the main functionality of DNS in a Windows Server 2008 implementation.

Exam Tip Table 9-3 lists the most common record types. However, review all the record types the Windows Server 2008 DNS server supports in preparation for the exam.

Windows Server DNS Features

The Windows Server 2008 DNS server complies fully with the RFC generated by the Internet Engineering Task Force (IETF, found at <http://www.ietf.org>) for Internet technology standards, but, in addition, it also includes a series of features that are designed to support the network operating system (NOS) features of AD DS. The DNS server in Windows Server 2008 can also operate with non-Windows-based DNS servers because it complies with all the RFCs related to the DNS service.

When the DNS service is integrated with AD DS, you can store DNS data in different locations within the directory database. DNS data can be stored within the domain partition of the directory. You choose this option for data that references the domain itself. For example, a child domain within a forest would normally have its data stored within its own domain partition to make the data available to all DNS servers in the domain. You can also store data within application directory partitions. Unlike domain partitions, application directory partitions have a controllable replication scope. For example, forest DNS data is stored in an application directory partition that spans the entire forest, making this data available to any DNS server within the forest. By default, Windows Server DNS creates two application directory partitions to host DNS data in each forest. These partitions are respectively named

ForestDnsZones and DomainDnsZones. In addition, DomainDnsZones is created in each child domain within a forest to host data for that domain.

Exam Tip DNS replication scopes are a key section of the exam. Examine them in your DNS server implementations and understand the contents of each scope type.

In addition, the DNS service in Windows Server 2008 has been improved to support background zone loading. When a DNS server hosts a large number of zones and records hosted in AD DS, it might take time for the server to boot because, traditionally, it needs to load all zone data before servicing requests. By using background loading, the DNS service can begin to respond to requests more quickly as it continues to load zone data in the background after the computer is started and logon is initiated.

To support the new read-only domain controller role, DNS has been updated to provide read-only DNS data for primary zones hosted on the RODC. This further secures the role and ensures that no one can create records from potentially unprotected servers to spoof the network.

Exam Tip Remember that DNS zones in RODCs are read-only *primary* DNS zones. Traditionally, read-only zones are secondary zones.

In an effort to support the removal of the WINS service from networks while still providing support for single-label names or names that do not include the parent name in their own (for example, SERVER10 instead of SERVER10.Contoso.com), DNS has been updated to include a GNZ. This zone can be used to host a small number of names with static IP addresses.

Exam Tip Keep in mind that you use GNZs to replace WINS implementations but only when you have a small number of single-label names to manage. Single-label or NetBIOS names are often required for previous applications that cannot work with the more complex FQDN structure. In fact, single-label names stem from older Windows NT-based networks or applications. In most cases, organizations should have been able to deprecate these applications and remove them from their networks, but some exceptions might remain. GNZs are designed to support these few remaining applications. However, if an organization needs to run a multitude of single-label names, you will need to implement the WINS service along with DNS.

Finally, in an effort to provide further protection against spoofing, DNS now supports the addition of global query block lists. When clients use protocols such as the Web Proxy Automatic Discovery Protocol (WPAD) or the Intra-site Automatic Tunnel Addressing Protocol (ISATAP) and rely on DNS to resolve host names, they can be vulnerable to malicious users who take advantage of dynamic updates to register computers that are not legitimate hosts.

WPAD is normally the protocol Web browsers rely on to discover network proxy server settings. Spoofing this address could lead users to malicious servers that impersonate legitimate proxy servers and potentially compromise a network. ISATAP is a transition protocol that enables IPv4 and IPv6 networks to work together. It does this by encapsulating IPv6 packets in IPv4 format to transmit them through routers. It does not support dynamic router discovery. Instead, it relies on a potential routers list to identify potential ISATAP routers. If this list is compromised, IPv6 packets could be routed to malicious routers and compromised in turn.

You can reduce the potential for these vulnerabilities by using global query block lists that contain specific blocked address ranges. Only the leftmost portion of an FQDN is included in global query block lists. When the DNS server receives a query that includes this name, it returns an answer as if no record existed. By default, the DNS server will generate this list at installation or during an upgrade of an existing DNS service. If either of the two protocols exists, the one that exists will not be blocked. If they do not exist, they will both be blocked. In addition, you can add your own names to this list to block names you do not want to be operational in your network.

MORE INFO Global query block lists

For more information on global query block lists, search for DNS global query block lists on the Microsoft Web site. You can download a document on the subject.

In short, the DNS service in Windows Server 2008 provides full support for all the standard features you would expect in a DNS server but also includes custom features that are available in Windows only.

Quick Check

1. What are the most common address types in IPv6 and which type is used by default on Windows Server 2008 and Windows Vista systems?
2. What is a major difference between the PNRP and DNS?
3. Which are the two types of read-only DNS servers supported by Windows Server 2008?
4. What is the first step in an AD DS logon process?
5. Which type of delegations can the Active Directory Domain Services Installation Wizard automatically remove?

Quick Check Answers

1. The most common IPv6 address types are link-local, site-local, and global unicast. By default, Windows Server 2008 and Windows Vista are designed to use dynamic IPv6 addresses. However, when no DHCPv6 servers are present in a network, IPv6 automatically assigns a link-local address to the interface.
2. One major difference between PNRP and DNS is the number of records each can contain. PNRP can scale to contain millions of name records; DNS is much more modest and relies on a hierarchy of servers to validate names.
3. The DNS server supports two read-only modes in Windows Server 2008. The first is the traditional, or legacy, secondary DNS. Secondary DNS servers are subordinate to one or more primary servers and contain only a copy of the information provided to them by a read-write source. The second type of read-only server is the one included in an RODC. This DNS server, however, includes *primary* read-only zones.
4. The first step in an AD DS logon process is a DNS request sent to locate the SRV for the closest domain controller. When this record is resolved, the logon process can begin through exchanges with the domain controller.
5. The Active Directory Domain Services Installation Wizard supports the removal of any delegation you have control over. This means it will properly remove child domain delegations, but it cannot remove top-level delegations because the root servers are on the Internet, and you do not have access to these servers.

Integration with AD DS

Because of its special Windows features, always deploy the Windows DNS server when you deploy AD DS. You can rely on a third-party DNS server to provide name resolution support for AD DS also, but it is significantly more work to set up and prepare this DNS server than to use the one built into Windows. When you use the Windows DNS server with AD DS, all DNS content is configured by default. This is why DNS installation is integrated with the domain controller promotion wizard. Installing DNS with AD DS performs several tasks that are usually completely transparent to the administrator running the wizard. These operations occur only during the creation of a forest, a domain tree within an existing forest, or a new domain within an existing forest.

If the AD DS deployment is for a forest root domain, DNS will create placeholders for the forward lookup zones (FLZ), the reverse lookup zones (RLZ) and conditional forwarders (CF). Then, it will generate two new zones within the FLZ. The first will be a container for the entire forest for the namespace you created during the installation of AD DS. This zone is usually named `_msdcs.domainname`. For example, for the `contoso.com` domain, this zone is called

_msdcs.contoso.com. In addition, it creates a zone within the FLZ for the root domain itself, as shown in Figure 9-7.

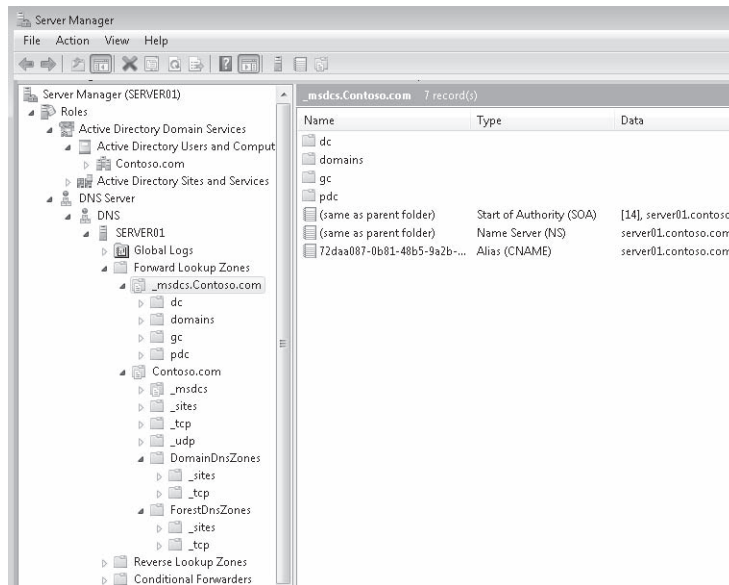


Figure 9-7 Forward lookup zones for the *contoso.com* forest

When the AD DS process creates a domain tree in an existing forest, it requires a manual delegation before the domain tree is created. Because the name of the domain tree is different from the root domain name—it must be different because that is the definition of a tree within a forest—the wizard cannot create the delegation on its own. When two DNS namespaces are different, neither has the authority to delegate information for the other, hence the need for a manual delegation. Then, after the delegation has been created, the AD DS Installation Wizard will create the DNS namespace and store it appropriately within the domain tree's new domain partition.

When the AD DS process creates a child domain in an existing forest, it automatically creates a delegation within the top-level root domain and then properly stores the DNS data for the child domain in the child domain's partition.

To remove a domain, you must run the Active Directory Domain Services Installation Wizard once again to remove the domain controller role, and then you can remove the AD DS role. However, because there is no interface to access the wizard anywhere, you must type **Dcpromo.exe** in the Search box of the Start menu to launch the wizard. When you remove the DC role, it will also remove DNS data created for a domain if this DC is the last DC in a domain. Also, if the DC is a global catalog (GC) server, it will give you a warning during the demotion because GCs support the search function in AD DS. During the removal of the DC role, you

will be prompted to remove DNS delegations, as shown in Figure 9-8. If this is a top-level domain such as a forest or tree root domain, make sure you clear this option; otherwise, you will receive an error because the wizard will ask you for credentials to delete the delegation. Because you do not have root-level credentials (for names such as .com, .net, .org, and so on), you cannot provide them and, therefore, cannot delete (or create, for that matter) root-level delegations. However, if it is a child domain, select to delete the DNS delegation and it will work properly.

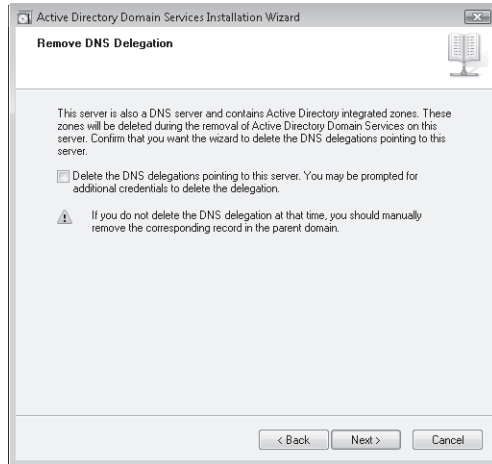


Figure 9-8 Removing DNS delegations with the AD DS Installation Wizard

Lesson 2: Configuring and Using Domain Name System

When you install the DNS Server role with AD DS, there is little configuration to be done. FLZs are created automatically; replication is configured automatically because it rides on the AD DS multimaster replication system, and you don't even need to add records because all computer systems running Windows 2000 or later can register and update their own records in the dynamic DNS AD DS requires.

However, some operations are not performed automatically. For example, the DNS server configuration does not, by default, include RLZs. It is a good idea to add them to support reverse lookups. In addition, the DNS server needs configuration finalization. For example, you must set it to support record scavenging, automatically deleting outdated records.

It is also a very good idea to review all the DNS server content to become familiar with it and ensure that all data and values correspond to your actual requirements.

After this lesson, you will be able to:

- Finalize the configuration of your DNS servers.
- Administer DNS servers and DNS replication.

Estimated lesson time: 40 minutes

Configuring DNS

The DNS configuration involves several activities. These include:

- Considering the security of your DNS servers to reduce their attack surface.
- Configuring scavenging settings for the server as a whole.
- Finalizing the configuration of your FLZs.
- Creating RLZs.
- Adding custom records to FLZs for specific services and resources.

It is also a good idea to make sure your DNS replication is working properly and that all DNS data is being replicated properly.

Security Considerations for the DNS Server Role

DNS servers that are exposed to the Internet are notorious targets for malicious users. The most common attack is a denial-of-service (DoS) attack that floods the DNS service with so many requests that the service cannot respond to valid requests. Another common attack form occurs when an attacker tries to obtain all the data contained within a DNS server, intending to use it to identify the object a network contains. This is called *footprinting the network*. Two more attack types attempt to modify data within the DNS server or redirect user queries from

a valid DNS server to another DNS server that would be under the control of the attacker. The latter usually occurs through the modification of DNS data contained within the DNS cache. Remember that DNS uses in-memory caching to increase the speed of responses to queries. When this data is corrupted, users can receive invalid responses to their queries.

This is why it is important to apply common security measures to your DNS installations. When you use a whole-brain approach to DNS configuration and you rely on DNS integration with AD DS in your internal network, your internal DNS servers are much less prone to attack because they do not share a namespace with the outside world and are, therefore, protected from external access by firewalls, which do not allow external users to access your internal DNS servers. This does not mean that internal servers do not need protection. Any time an untrusted user can connect to your network either through the wired connections or through wireless access, your infrastructure is at risk. This is why extensive screening is a good practice whenever you allow someone you are not familiar with to connect to your network. It is not because you are inside the firewall that everything is protected by default.

Consider a different security approach with internal vs. external DNS servers. When servers are in an external or perimeter network, they should be highly secured. One good protection method is to use a secondary or subordinate server only whenever the server is exposed to the outside world. Then, you configure the zone updates to occur only from known sources that are included within DNS itself.

In internal networks, tie the DNS Server role to the DC role and ensure that they support secure dynamic updates only. This will help protect them from obtaining or transmitting erroneous data. Verify DNS data on a regular basis to validate it and monitor your DNS event logs to identify potential security issues quickly if they arise.

Exam Tip The exam focuses on DNS usage with AD DS. Because of this, it does not cover external or standalone DNS servers. If you find you need to configure an external DNS server, you can look up more information on DNS security at <http://technet2.microsoft.com/windowsserver/en/library/fea46d0d-2de7-4da0-9c6f-2bb0ae9ca7e91033.mspx?mfr=true>.

Working with DNS Server Settings

DNS stores name records for a specific period of time. Each name record is assigned a TTL value. When this value expires, the record should be removed to avoid providing false positive results to users performing lookups on the name. Fortunately, the DNS server in Windows Server 2008 can perform this task automatically through server or zone scavenging. When applied to the server, scavenging cleans all active zones on the server. When applied to a particular zone, only the records for the zone are scavenged.

Configuring Scavenging for All Zones To set scavenging for an entire server, you must assign the setting through the server's action menu.

1. Right-click the server name in the DNS node of Server Manager and choose Set Aging/Scavenging For All Zones.
2. Select the Scavenge Stale Resource Records check box to enable the feature. No-Refresh Interval refers to the time between the most recent refresh of a record stamp and the moment when the system allows the timestamp to be refreshed again. Refresh Interval refers to the earliest moment when a record may be updated or when it may be scavenged if no updates have been applied. The default value of seven days is sufficient for most networks.
3. Leave the default values as is and click OK.
4. Because you set the values for existing zones, DNS also enables you to set it for any future zone you create, including Active Directory–integrated zones. Select the Apply These Settings To The Existing Active Directory-Integrated Zones check box and click OK.

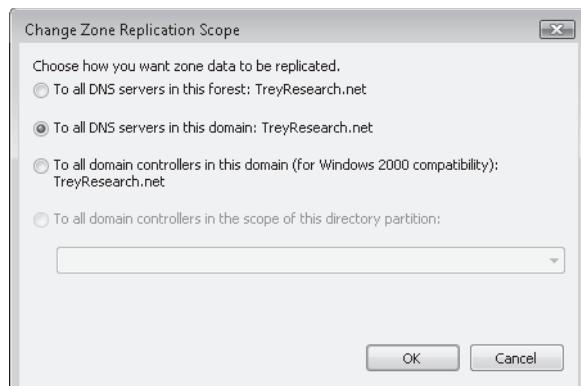
Your DNS zones are set to remove stale records. Make sure you apply these settings to every DNS server in your network. Make this part of your default configuration for DNS servers. If you need to modify the setting for a single zone, you must use the Properties dialog box for that zone. Zone scavenging is performed by using the General tab and clicking the Aging button.

You'll note that in the server's context menu, you can also manually initiate scavenging by clicking Scavenge Stale Resource Records. You use this operation when you discover that your servers are sending out stale data.

If you do discover that records are stale, you can also use the *Clear Cache* command from the same context menu. Because the DNS server relies heavily on the in-memory cache to improve performance, you might have scavenged records from the database but find they are still in the cache, which might still be providing false positives.

Finalizing FLZ Configuration When you examine the Properties dialog box for a FLZ, you'll find that there are several options you can set for each zone. Make a point of examining these options and configure the following settings for each production DNS zone as a best practice:

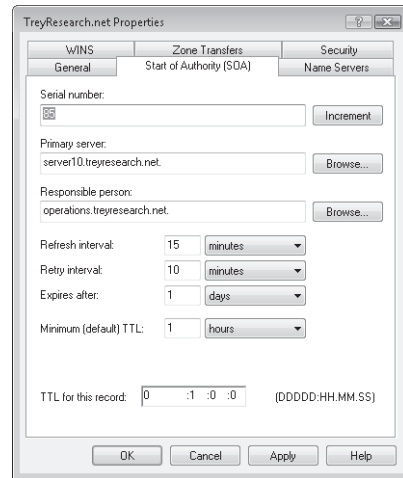
- On the General tab, make sure each internal DNS zone is Active Directory-Integrated, uses the proper replication scope, and supports Secure Only dynamic updates.
 - Domain-based DNS zones should replicate to all DNS servers in the domain. Each DC that also hosts the DNS role will include the zone.
 - Forest DNS zones should replicate to all DNS servers in the forest.
 - If you maintain Windows 2000 Server DCs in your network, you must use the To All Domain Controllers In This Domain (For Windows 2000 Compatibility) option because Windows 2000 Server does not support application directory partitions.
 - You can also set replication to custom application directory partitions, but you must create the partition first.



- On the Name Servers tab, ensure that each DNS zone includes at least two name servers. Just as you would create at least two DCs for each domain, create two DNS servers for each zone as a best practice.
- On the WINS tab, assign WINS lookups only if you cannot use GNZs and you must deploy WINS. This lesson will discuss single-label name management further in a later section.
- On the Zone Transfers tab, set the name servers to which you allow this zone to be transferred upon request. If this zone is integrated with Active Directory, zone transfers are not required. This tab is mostly used for previous DNS server installations.
- On the Security tab, review the default security settings. These settings use the appropriate configuration for most networks, but in highly secure implementations, they might need to be revised and modified.
- The final tab is the SOA. SOA records identify the zone and its related information such as owner, operator, update schedules, and so on. These records include the following information:
 - Serial Number, which is assigned when your zone is created. You can increment the serial number if you need to change its value.
 - Primary Server is the master server for this zone. This is usually the server where the zone was first created.
 - Responsible Person should list the operator name for this server. Normally, this is a standard term such as Hostmaster or Operations. By default, Windows Server 2008 assigns `hostmaster.dnszonename` where `dnszonename` is the FQDN of the zone. Responsible Person entries are based on Responsible Person records. These records are not created by default. Create a proper Responsible Person record for each zone or, at the very least, for each master DNS server and assign it to this value.
 - The SOA lists the various intervals and time-based settings for the record. These include the Refresh Interval, the Retry Interval, the Expires After setting, and the

Minimum (Default) TTL for the record. Default values are acceptable for most record types.

- ❑ The last value of the SOA is the TTL For This Record. Note that it is assigned to the same value as the Minimum (Default) TTL value listed above it in the dialog box.



Finalize these settings for each zone you manage on your DNS servers.

Creating a Responsible Person Record As mentioned earlier, each zone should be assigned a Responsible Person (RP) as a best practice. This means you need at least one RP record in your DNS configuration. Use the context menu for the zone you want to host this record in to create the record. Keep in mind that you will require different items for the creation of this record. These include:

- **A common group name** This name will be displayed in the record.
- **A group mailbox in the directory** It is best to use a group mailbox to make sure the messages sent to this address are treated in a timely fashion.
- **A text record to include with the Responsible Person record** The text record can indicate information about your organization and its DNS management policies.

Use the following procedure to create the RP record. Begin with the Text Record.

1. Right-click the zone name and choose Other New Records.
2. In the Select A Resource Record Type list, scroll down to select Text (TXT) and click Create Record.
3. In the New Resource Record dialog box, type the name of the record, for example, **Disclaimer**, and move to the Text entry box.
4. Type your message. Click OK to create the record. This returns you to the Resource Record Type dialog box.

This should include information about who you are and what your DNS management practices are. You might consider preparing the message in a word processor and then pasting it in this dialog box because the text box does not include any proofing capabilities.

5. In the Select A Resource Record Type list, scroll up to select Responsible Person (RP) and click Create Record.
6. In the New Resource Record dialog box, type the name of the record in the Host Or Domain text box, for example, **Operations**, and click Browse to locate the mailbox of the responsible person. You can also type the address if you know it.
7. Click Browse to locate your newly created text record. Navigate to the zone you are working with to locate the text record, select it, and click OK.
8. Click OK to create the record. Click Done to close the Resource Record Type dialog box.
9. Return to the zone Properties dialog box or double-click the Start Of Authority record to assign the RP record to the SOA record.

Perform these operations for each zone you manage. It is always better to configure the zone completely than to have to figure out what to do if issues arise and nothing is configured.

Creating Reverse Lookup Zones

Small networks with few computers, for example, fewer than 500, might not require RLZs. These zones are used to provide resolution from an IP address to a name instead of a name to an IP address. These are most often used by applications. For example, a secure Web application will use a reverse lookup to verify that the computer it is communicating with is actually the right computer and not another computer impersonating it. If you do not have any such application in your network, then you can safely do without RLZs.

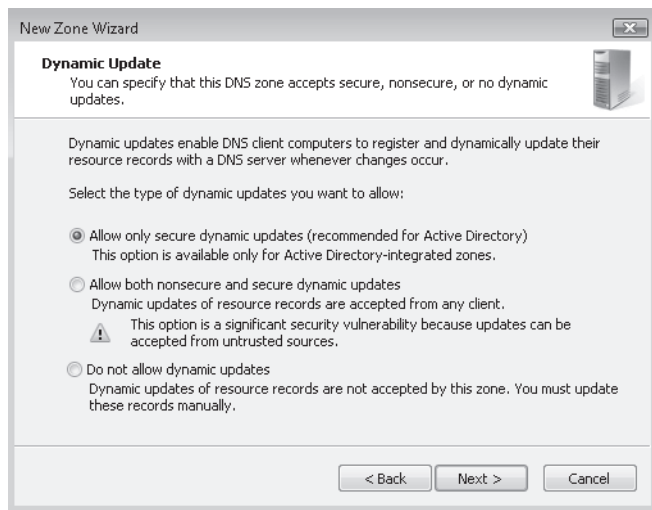
However, clients that have the ability to update their own DNS records dynamically will also create a PTR record—a reverse record that maps the IP address to the name—and try to store it within the RLZ that corresponds to the FLZ their name record is located in. If there is no RLZ, these records will never be generated.

MORE INFO How dynamic updates work

For more information on dynamic updates and how they work, go to <http://technet2.microsoft.com/windowsserver/en/library/e760737e-9e55-458d-b5ed-a1ae9e04819e1033.mspx?mfr=true>.

If you need RLZs, create them for the corresponding FLZs. Create a zone for each named FLZ. In an Active Directory–integrated DNS implementation, you would create a RLZ for each domain DNS zone. In a multidomain forest, this would include the root domain, any child domain, and any domain trees. Use the following procedure:

1. Move to the Reverse Lookup Zone section of the DNS node in Server Manager.
2. Right-click Reverse Lookup Zone and select New Zone.



As soon as the zones are created, they will begin supporting record hosting when the next dynamic update refresh occurs on your client systems.

Exam Tip Practice working with zones and zone properties because they are an important part of the exam topics.

Quick Check

1. Why should you configure scavenging and aging on a DNS server?
2. When should you create reverse lookup zones?

Quick Check Answers

1. Every DNS name record is assigned a TTL value when created. This value determines when the information in the record is no longer valid. If the record is not renewed, then it becomes a stale record. Aging and scavenging on the DNS server will automatically remove stale records to limit the possibility of false positives when users request data from the DNS server.
2. Reverse lookup zones are mostly useful for secure Web applications, which must validate the IP address provided by the systems they communicate with. If a network does not include any such application, then reverse lookup zones are not required.

Creating Custom Records

The last step in a DNS server configuration is the creation of custom records for the FLZs. Custom records are created manually and will provide a variety of services in your network. For example, you might need to create the following:

- An MX record to point to your e-mail servers.
- An alias record such as *intranet.domainname* to point to an Office SharePoint Server server farm to support collaboration in your network.
- SRV records for various services in your network. For example, you must create SIP records for Microsoft Office Communication Server deployments.

Time will tell which custom records you need. In an internal network, manually created records should be infrequent because of the dynamic update process initiated by client systems.

Exam Tip Practice record creation because it is also an important topic on the exam.

Forwarders vs. Root Hints

Name resolution is performed by two main methods. DNS servers will either contain root hints that enable them to identify and locate authoritative DNS servers for root names or rely on forwarders to link them to another server that will perform the lookup for them.

By default, Windows DNS Server relies on root hints to perform lookups. This means that if your users need to perform a lookup on the Internet, your DNS servers will communicate with the name servers. In smaller organizations, this is quite acceptable because even if your DNS servers expose themselves by communicating directly with the Internet, they are the ones who initiate the communication. External systems reply to the initiated communication only and cannot initiate the communication themselves.

However, in highly secure networks, you might prefer to rely on forwarders instead of root hints. For example, you might place two standalone DNS servers in your perimeter network and link the internal DNS servers to these servers through forwarders. Each time the internal servers need to resolve an Internet name, they link to one of the servers in the perimeter and ask it to perform the lookup for them. This way, the only servers to communicate outside the network are the more secure standalone servers in the perimeter.

Forwarders are configured as part of the properties of the DNS server and are accessed from the Forwarders tab in the DNS server Properties dialog box. (See Figure 9-15.) If you are configuring forwarders for security purposes, make sure you uncheck the Use Root Hints option if no forwarders are available; otherwise, your internal DNS servers will communicate directly with the Internet if your servers in the perimeter do not respond.

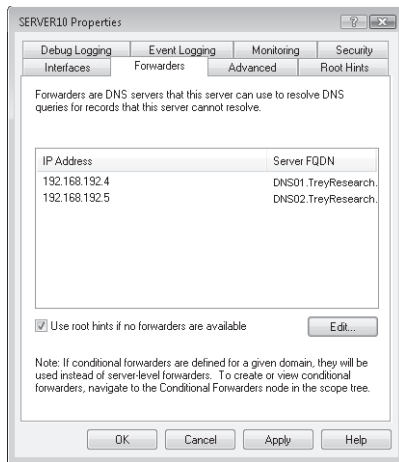


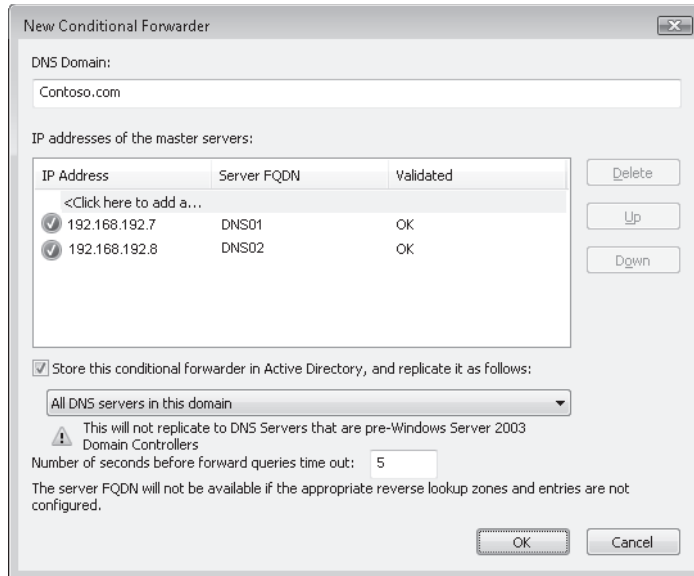
Figure 9-15 Configuring forwarders in DNS

You can also use conditional forwarders in your DNS configuration. Conditional forwarders are used to forward DN requests when specific conditions are met. For example, if you want to link two namespaces but only when users request a particular name, you would use conditional forwarders.

For example, consider the following scenario. Your network includes two forests. The first is the production forest, the one that contains all the accounts users use to work together in your organization. The second is a special forest that was created to test the AD DS integration of third-party applications with the AD DS forest schema before they are deployed in your production forest. Because of the schema changes, you do not want to link your forests together through a forest trust. Therefore, you create conditional forwarders in each forest so that users in the production domain, mostly developers and IT professionals, can link from the production to the staging domain.

Conditional forwarders include their own container in DNS Server.

1. To create a new conditional forwarder, right-click the Conditional Forwarders node and select New Conditional Forwarder.
2. Type the name of the DNS domain you want to forward to.
3. Click <Click Here To Add An IP Address Or DNS Name> and type the server's IP address.
4. Add at least two servers to the list.
5. As a best practice, store the conditional forwarder in Active Directory and determine which replication scope to apply to the forwarder. Click OK.



In the preceding example, you would replicate the data only to the production domain because you do not need to replicate it to the entire forest. In other cases, you might need to replicate it to the entire forest.

Note that when you create a conditional forwarder, it creates a new container for the domain you will forward to under the Conditional Forwarders node. From now on, each time your users request a name resolution that contains this domain name, your DNS servers will automatically forward it to the DNS servers you provided in the list.

Single-Label Name Management

When you want to manage single-label names, you will need to create a GNZ manually. A single GNZ is required for each forest. The basic process of creating a GNZ requires five steps, but it involves an operation on each DNS server in the forest. If you are using AD DS–integrated DNS servers and each of your DCs is also running the DNS service, you must perform this operation on each DC. This means using domain administrator credentials to complete the operation.

- Create the GlobalNames FLZ.
- Set its replication scope to all DNS servers in the forest.
- Do not enable dynamic updates for this zone.
- Enable GNZ support on each DNS server in the forest.
- Add single-label names to the DNS zone.

Configuration is performed through the command line because there is no graphical interface to access this feature. However, you can create the GNZ through Server Manager, but enabling

GNZ support in a DNS server requires a modification of the Windows Registry. This modification is performed with the *Dnscmd.exe* command and uses the following format:

```
dnscmd /config /enableglobalnamesupport 1
```

This command needs to be run on each DNS server in the forest. If you need to support single-label names and you do not want to use WINS, you might want to make this command part of your standard DNS server installation and configuration process. You need to restart the DNS service when the command has been run.

After you have enabled GNZ support, you can begin to add records. GNZ names are aliases because each object in your network already has a host name in DNS. You create an alias and point it to the corresponding FQDN for the object. GNZ aliases, like WINS names, cannot have more than 15 characters—they actually use 16 characters, but the system reserves the last character. If you want to create the names through a command file, use the following command format for each name:

```
dnscmd dnsservername /recordadd globalnames singlelabelname cname  
correspondingdnsname
```

Where *dnsservername* is the name of the DNS server that you are adding the name to, the *singlelabelname* is the 15-character name you want to add, and *correspondingdnsname* is the FQDN of the object whose GNZ name you are adding.

MORE INFO GlobalNames zones

For more information on GNZs, view the DNS GlobalNames Zone Deployment document at <http://www.microsoft.com/downloads/details.aspx?FamilyID=1c6b31cd-3dd9-4c3f-8acd-3201a57194f1&displaylang=en>.

DNS and WINS

If you are in a network that requires a multitude of single-label names and you cannot provide support for them through a GNZ because there are simply too many names to manage, install the WINS service on at least two servers in your network. WINS will then automatically generate and manage names for each object in the network. Remember that WINS services are a feature of Windows Server 2008, not a role, and that it is an outdated technology that has not been updated since the original release of Windows Server 2003.

If you do deploy WINS, remember that:

- WINS does not appear in Server Manager. To administer WINS, you must use the WINS console in the Administrative Tools program group.
- WINS supports IPv4 addresses only and will not be updated to support IPv6.

- You need at least two WINS servers to provide fault tolerance for single-label names in your network. These two servers should be configured to use push/pull synchronization to make sure both name databases are synchronized at all times.
- You need to ensure that the values for WINS are specified in the DHCP settings you send to computers requiring dynamic IPv4 addresses. Two settings are required. The first lists the name servers and the second identifies which type of node each client will work with.
 - 044 WINS/NBNS Servers identifies which servers host the WINS service.
 - 046 WINS/NBT Node Type identifies how nodes interact with WINS. The most commonly used node type is 0x8 or the Hybrid node type. This minimizes the amount of broadcasting required in single-label name networks.
- You can also integrate WINS and DNS by modifying the properties of an FLZ. This property sheet includes a WINS tab that is unused unless you have WINS servers in your network. (See Figure 9-16.) This feature is useful in networks in which many clients rely on WINS and it is possible that some of the client device names will not be present in DNS. However, all Windows operating systems since Windows 2000 can participate in a dynamic DNS infrastructure. Networks that run earlier clients than Windows 2000 are becoming very rare.

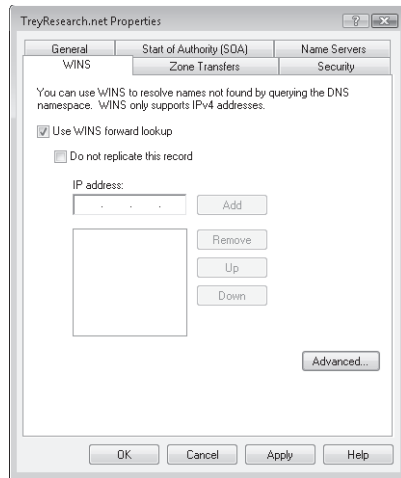


Figure 9-16 Linking DNS with WINS to provide both FQDN and single-label name resolution

DNS and DHCP Considerations

When you work with dynamic DNS and you integrate the DNS service with the AD DS directory store, you must change the traditional approach network administrators use to configure DHCP settings that are provided to each client device that relies on dynamic IP addresses, whether they are IPv4 or IPv6.

Traditionally, network administrators provide as few as two central DNS servers in the server options of the DHCP settings. This provides all client devices with the DNS addresses they need to resolve both internal and external FQDNs, but because the servers were centrally located, any client that was in a remote site would need to perform a DNS lookup over the WAN.

However, with the integration of DNS and especially DNS data with the directory store, DNS data is now available wherever there is a DC and, to provide authentication services wherever clients are located, DCs are distributed throughout a network. In fact, some organizations have DCs available wherever there are at least 20 clients. With the advent of server virtualization through Hyper-V as well as DC hardening through the RODC, DCs can be even more prevalent in networks. This means that DNS data, even read-only DNS data, will be available in each remote site or branch office. Clients can use servers in their local site to perform FQDN lookups.

However, for clients to perform the lookup locally, they must know about the presence of local DNS servers. Imagine the following scenario:

- A client in a remote site uses a dynamic IP address allocated through DHCP.
- There are two DCs in the remote site.
- DNS is integrated with the directory and is replicated with the domain partition.
- DHCP sends out values for only two DNS servers in a central site.
- When the client boots in the morning, it performs a DNS lookup to locate its closest DC to log on.
- The DNS lookup occurs over the WAN to request the name resolution from one of the two central DCs.
- The central DNS servers look up the client's site and find that there are two local DCs to support logon.
- The DNS server returns the location of the closest DC to the client, once again over the WAN.
- The client contacts its local DC to log on.

In this scenario, the client cannot log on if there is no WAN connectivity even though the DNS data is stored locally within the two DCs!

Because of this, DHCP options must be modified as follows:

- The server scope should continue to provide at least two addresses for centrally located DNS servers for redundancy purposes. If the local DCs is down, clients will still be able to log on, albeit over the WAN.
- Each individual address scope should include options for name resolution servers, and these records should point to the DCs that are local to the site the scope is assigned to. This means adding the *006 DNS Servers* value to each individual scope in DHCP.

- All DCs should also be running the DNS Server role. That's all you have to do. If a DNS zone is stored in the AD DS directory store, it will be made available to the DNS service as soon as you install the DNS Server role on the DC. There is nothing more to configure except global DNS server settings.

Keep this in mind when planning to integrate DNS with DHCP.

Working with Application Directory Partitions

In certain circumstances, you will want to create custom application directory partitions in support of DNS data replication. Remember that application directory partitions control the replication scope of the data they contain. The DNS server creates two application directory partitions when it is installed with AD DS in a forest, one for forest data and one for domain data in each domain, but under certain conditions, these two scopes might not be appropriate, especially in complex forests.

Consider this scenario. Your forest includes three domains: the forest root, a global child production domain, and a development domain. You created the development domain because your developers have special access right requirements and you do not want to grant them these access rights in the production domain. All production domain users except for system administrators have standard user access rights. In the development domain, however, you can grant developers higher access rights—rights to create, modify, or delete objects—because this domain does not affect production operations.

In addition, you created only a single account for each developer. This account is located in the global child production domain and has standard user rights, but through the transitive trusts inherent in each forest, developers can use their accounts from the production domain to access objects in the development domain where their production domain accounts have higher access rights.

By default, name resolution between the two child domains passes through the forest root domain. Developers access this domain on a constant basis every day, so to provide them with faster name resolution, you create a custom application directory partition to share the DNS records between the development and the production domains. This means that because the data is available in the partition, production DNS servers will not need to pass through the forest root domain to resolve development domain names. (See Figure 9-17.)

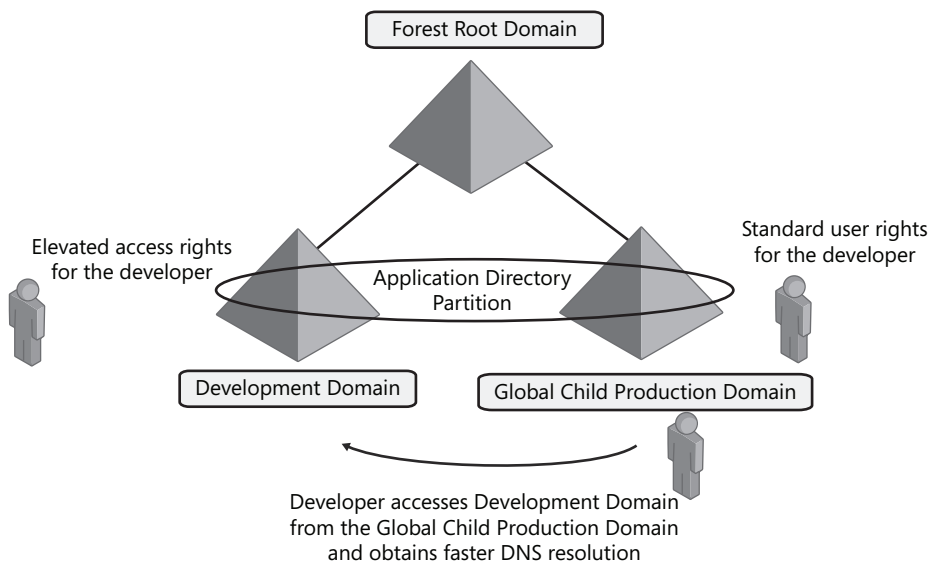


Figure 9-17 Relying on custom application directory partitions to share DNS data between two child domains

Creating and Assigning Custom Application Directory Partitions

Custom application directory partitions are created through the command line with the *Dnscmd.exe* command. There is no interface for creating these partitions. However, after they are created, the partitions can be assigned through the graphical interface. You can perform each operation through the command line if you prefer. You must perform three tasks:

- Create the partition.
- Enlist DNS servers into the partition.
- Assign the zones whose replication scope you want to change to the newly created partition.

To create an application directory partition, you must be a member of the Enterprise Admins groups because you must have full access to the forest.

1. Log on to a DNS server with an account that is a member of the Enterprise Admins group for the forest.
2. Launch an elevated command prompt through the context menu and the *Run as administrator* command.
3. Type the following command:

```
dnscmd dnsservername /createdirectorypartition partitionfqdn
```

where the *dnsservername* is the FQDN of your DNS server or its IP address, and *partitionfqdn* is the FQDN of the partition you want to create.

For example, if you want to create a new partition on SERVER10 and name it partition01.treyresearch.net, you would use the following command:

```
dnscmd server10.treyresearch.net /createdirectorypartition  
partition01.treyresearch.net
```

4. Enlist the server into the partition. Once again, you use the *Dnscmd.exe* command. Type the following command:

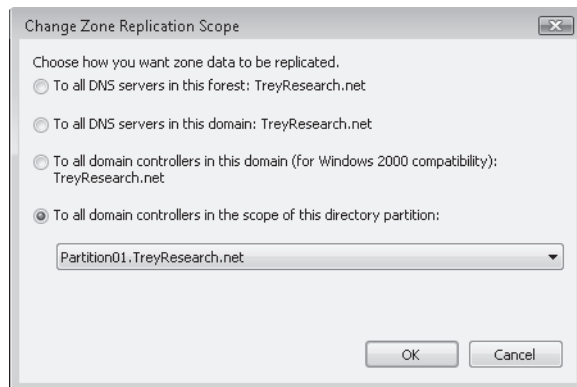
```
dnscmd dnsservername /enlistdirectorypartition partitionfqdn
```

You need to repeat this command for each DNS server you want to enlist into the partition. Note that the server you use to create the partition is enlisted by default. In the preceding scenario, you would need to enlist all the DNS servers for the production domain as well as all the DNS servers for the development domain into the partition. For example, if you want to enlist SERVER30—a server of the child domain—as an additional server into the new partition named partition01.treyresearch.net, you would use the following command:

```
dnscmd server30.intranet.treyresearch.net /enlistdirectorypartition  
partition01.treyresearch.net
```

Now you can change the replication scope of the zones you want to make available to the members of the new application directory partition.

5. Return to the DNS node in Server Manager, right-click the name of the zone you want to change, and select Properties.
6. On the General tab, click the Change button to change the replication scope.
7. In the Change Zone Replication Scope dialog box, select To All Domain Controllers In The Scope Of This Directory Partition and click the drop-down list to select your new partition. Click OK twice.



Be careful when you work with application directory partitions because many of the commands are manually entered. If you make a mistake, you could damage the replication scope of your servers and, therefore, disable name resolution.

MORE INFO Application directory partitions

For more information on application directory partitions, go to <http://technet2.microsoft.com/windowsserver2008/en/library/2e2e0678-1775-4cdd-8779-32d5c281540f1033.msp?mfr=true>.

Exam Tip Replication scopes and application directory partitions are an important part of the exam. Be sure you understand them fully.

Administering DNS Servers

You've already seen several tools in operation through practices and detailed step-by-step lists. However, you might need to work with other tools when working with your DNS servers. Table 9-4 outlines the different tools you can use to support DNS operations and management.

Table 9-4 Common DNS Administration Tools

Tool	Task	Location
DNS Manager	Perform initial configuration of a new server. Connect to and manage a local DNS server. Add and remove forward and reverse lookup zones. Add, remove, and update resource records in zones. Modify how zones are stored and replicated between servers. Modify how server processes queries and handles dynamic update. Modify security for specific zones or resource records. Perform maintenance. Monitor contents of the server cache. Tune advanced server options. Configure and perform aging and scavenging of stale resource records.	Administrative Tools program group or Server Manager
<i>Dnscmd</i>	Manage all aspects of DNS servers. This is the most powerful command-line tool for DNS administration. Common switches include: <ul style="list-style-type: none"> ■ <i>/info</i> to obtain server information. ■ <i>/config</i> to modify server configuration parameters. ■ <i>/statistics</i> to obtain operational statistics from a server. ■ <i>/clearcache</i> to clear and reset the cache. ■ <i>/startscavenging</i> to initiate a scavenging operation. ■ <i>/directorypartitioninfo</i> for information about partitions. ■ <i>/exportsettings</i> to create a backup file of your server's settings. 	Command line

Table 9-4 Common DNS Administration Tools

Tool	Task	Location
<i>Dnslint</i>	Diagnose common DNS name resolution issues. Common switches include: <ul style="list-style-type: none"> ■ <i>/d</i> to request domain name resolution tests. ■ <i>/ql</i> to verify DNS query tests from a list. ■ <i>/ad</i> to verify records specifically related to Active Directory. 	
Event Viewer	There are two options for monitoring DNS servers: <ul style="list-style-type: none"> ■ Default logging of DNS server event messages to the DNS server log. ■ Debug options for trace logging to a text file on the DNS server. This option is enabled through the DNS server's Properties dialog box and is disabled by default. Use it only for debugging purposes. 	Server Manager
<i>Ipconfig</i>	Display and modify IP configuration details. Common switches include: <ul style="list-style-type: none"> ■ <i>/all</i> to display all network configuration settings on a system. ■ <i>/renew</i> to request a dynamic IPv4 address renewal from DHCP. ■ <i>/renew6</i> to request a dynamic IPv6 address renewal from DHCP. ■ <i>/release</i> to release a dynamic IPv4 address. ■ <i>/release6</i> to release a dynamic IPv6 address. ■ <i>/flushdns</i> to clear the DNS resolver cache on a system. ■ <i>/registerdns</i> to renew a dynamic DNS registration for a system. 	Command line
<i>Nslookup</i>	Perform query testing of the DNS domain namespace. <i>Nslookup</i> is also a command interpreter that is entered by simply typing nslookup at the command line. Type exit to return to the command line. However, it can also be used directly. To do so, type nslookup followed by the hostname or the IP address of the computer you are looking for.	Command line
System Monitor	Create charts and graphs of server performance trends. Determine performance benchmarks.	Server Manager, Diagnostics, Reliability, and Performance

Exam Tip Run through each of these tools. DNS operation is an important part of the exam.

Little will go wrong with your internal DNS implementations if you follow the guidelines outlined here. However, there is always the possibility of uncontrolled issues. This is why you should become familiar with the tools listed in Table 9-4. Examine DNS events and understand them.

MORE INFO DNS troubleshooting and potential resolutions

For more information on DNS troubleshooting and potential resolutions to DNS issues, go to <http://technet2.microsoft.com/windowsserver2008/en/library/8e3f7e44-91dd-44c4-81cf-158cea7089021033.mspx?mfr=true>. To obtain *Dnslint.exe*, go to <http://support.microsoft.com/kb/321045>.

Chapter 10

Domain Controllers

Domain controllers (DCs) host the directory service and perform the services that support identity and access management in a Microsoft Windows enterprise. To this point in the training kit, you have learned to support the logical and management components of an Active Directory Domain Services (AD DS) infrastructure: users, groups, computers, and Group Policy. Each of these components is contained in the directory database and in SYSVOL on domain controllers. In this chapter, you will begin your exploration of the service-level components of Active Directory, starting with the domain controllers themselves. You will learn how to add Windows Server 2008 domain controllers to a forest or domain, how to prepare a Microsoft Windows Server 2003 forest or domain for its first Windows Server 2008 DC, how to manage the roles performed by DCs, and how to migrate the replication of SYSVOL from the File Replication Service (FRS) used in previous versions of Windows to the Distributed File System Replication (DFS-R) mechanism that provides more robust and manageable replication.

Exam objectives in this chapter:

- Configure a forest or a domain.
- Configure Active Directory replication.
- Configure operations masters.

Before You Begin

To complete the practices in this chapter, you must have created a domain controller named SERVER01 in a domain named *contoso.com* and a member server, with a full installation, joined to the domain named SERVER02. See Chapter 1, “Installation,” for detailed steps for this task.

Real World*Dan Holme*

Active Directory enables you to configure a domain and a forest with a single domain controller. But that's not enough. Domain controllers provide functionality critical to the identity and access management requirements of an enterprise, and if a domain controller fails, you must have a way to provide continuity of service. That's why it's very important to have at least two DCs in a domain. As soon as you start adding DCs to a domain, you start needing to consider replication, and in this chapter, you'll learn about one of the exciting new features of Windows Server 2008: DFS-R or SYSVOL. FRS, used by previous versions of Windows and supported by Windows Server 2008 for backward compatibility, has been a notorious weak spot prone to problems and difficult to troubleshoot. To take advantage of this feature, all domain controllers must be running Windows Server 2008, so you'll need to know how to prepare an existing forest for its first Windows Server 2008 DC—another objective of this chapter. Finally, as you add domain controllers to an enterprise, you need to consider the placement of single master operations, which are special roles assigned to one DC in a forest or domain. By the time you're through with this chapter, you'll have the skills to improve the redundancy, performance, and manageability of multiple domain controllers in your enterprise.

Lesson 1: Installing Domain Controllers

In Chapter 1, you used the Add Roles Wizard in Server Manager to install Active Directory Domain Services (AD DS). Then you used the Active Directory Domain Services Installation Wizard to create the first DC in the *contoso.com* forest. Because DCs are critical to authentication, it is highly recommended to maintain at least two domain controllers in each domain in your forest to provide a level of fault tolerance in the event that one DC fails. You might also need to add domain controllers to remote sites or create new domains or trees in your Active Directory forest. In this lesson, you will learn user-interface, command-line, and unattended methods for installing domain controllers in a variety of scenarios.

After this lesson, you will be able to:

- Install a DC, using the Windows interface, *Dcpromo.exe* command-line parameters, or an answer file for unattended installation.
- Add Windows Server 2008 DCs to a domain or forest with Windows Server 2003 and Windows 2000 Server DCs.
- Create new domains and trees.
- Perform a staged installation of a read-only domain controller.
- Install a DC from installation media to reduce network replication.
- Remove a domain controller.

Estimated lesson time: 60 minutes

Installing a Domain Controller with the Windows Interface

If you want to use the Windows interface to install a domain controller, there are two major steps. First, you must install the AD DS role, which, as you learned in Chapter 1, can be accomplished using the Add Roles Wizard in Server Manager. After the AD DS role installation has copied the binaries required for the role to the server, you must install and configure AD DS by launching the Active Directory Domain Services Installation Wizard, using one of these methods:

- Click Start and, in the Start Search box, type **dcpromo** and click OK.
- When you complete the Add Roles Wizard, click the link to launch the Active Directory Domain Services Installation Wizard.
- After adding the AD DS role, links will appear in Server Manager that remind you to run the Active Directory Domain Services Installation Wizard. Click any of those links.

The Active Directory Domain Services Installation Wizard is shown in Figure 10-1.

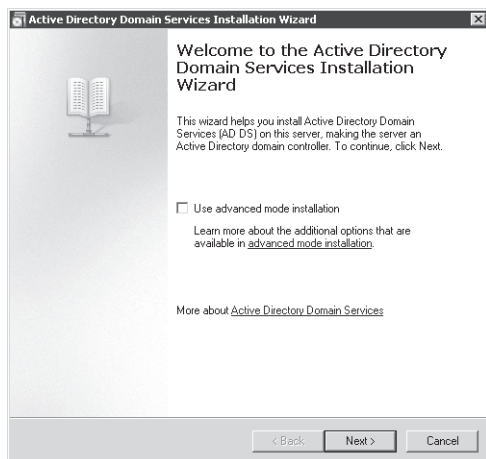


Figure 10-1 The Active Directory Domain Services Installation Wizard

NOTE All-in-one wizard

Microsoft documentation for Windows Server 2008 emphasizes the role-based model, so it recommends you add the AD DS role and then run *Dcpromo.exe* (the Active Directory Domain Services Installation Wizard). However, you can simply run *Dcpromo.exe* and, as a first step, the wizard detects that the AD DS binaries are not installed and adds the AD DS role automatically.

Unattended Installation Options and Answer Files

You can also add or remove a domain controller at the command line, using unattended installation supported by the Windows Server 2008 version of *Dcpromo.exe*. Unattended installation options provide values to the Active Directory Domain Services Installation Wizard. For example, the `NewDomainDNSName` option specifies a fully qualified domain name (FQDN) for a new domain.

These options can be provided at the command line by typing `dcpromo /unattendOption:value`, for example, `dcpromo /newdomaindnsname:contoso.com`. Alternatively, you can provide the options in an unattended installation answer file. The answer file is a text file that contains a section heading, `[DCINSTALL]`, followed by options and their values in the `option=value` form. For example, the following file provides the `NewDomainDNSName` option:

```
[DCINSTALL]
NewDomainDNSName=contoso.com
```

The answer file is called by adding its path to the *unattend* parameter, for example:

```
dcpromo /unattend:"path to answer file"
```

The options in the answer file can be overridden by parameters on the command line. For example, if the *NewDomainDNSName* option is specified in the answer file and the */NewDomainDNSName* parameter is used on the command line, the value on the command line takes precedence. If any required values are neither in the answer file nor on the command line, the Active Directory Domain Services Installation Wizard will prompt for the answers, so you can use the answer file to partially automate an installation, providing a subset of configuration values to be used during an interactive installation.

The wizard is not available when running *Dcpromo.exe* from the command line in Server Core. In that case, the *Dcpromo.exe* command will return with an error code.

For a complete list of parameters that you can specify as part of an unattended installation of AD DS, open an elevated command prompt and type the following command:

```
dcpromo /?[:operation]
```

where *operation* is one of the following:

- **Promotion** returns all parameters you can use when creating a domain controller.
- **CreateDCAccount** returns all parameters you can use when creating a prestaged account for a read-only domain controller (RODC).
- **UseExistingAccount** returns all parameters you can use to attach a new DC to a prestaged RODC account.
- **Demotion** returns all parameters you can use when removing a domain controller.

MORE INFO *Dcpromo* parameters and unattended installation

For a complete reference of *Dcpromo* parameters and unattended installation options, see <http://go.microsoft.com/fwlink/?LinkID=101181>.

NOTE Generate an answer file

When you use the Windows interface to create a domain controller, the Active Directory Domain Services Installation Wizard gives you the option, on the Summary page, to export your settings to an answer file. If you need to create an answer file for use from the command line, for example, on a Server Core installation, you can use this shortcut to create an answer file with the correct options and values.

Installing a New Windows Server 2008 Forest

Chapter 1 discussed the installation of the first Windows Server 2008 DC in a new forest, using the Windows interface. Exercise 3, “Install a New Windows Server 2008 Forest with the Windows Interface,” and Exercise 4, “Install a New Windows Server 2008 Forest,” of Lesson 1, “Installing Active Directory Domain Services,” in that chapter detailed the steps to add the AD DS role to a server by using Server Manager and then to run *Dcpromo.exe* to promote the server to a domain controller. When creating a new forest root domain, you must specify the forest root Domain Name System (DNS) name, its NetBIOS name, and the forest and domain functional levels. The first domain controller cannot be a read-only domain controller and must be a global catalog (GC) server. If the Active Directory Domain Services Installation Wizard detects that it is necessary to install or configure DNS, it does it automatically.

You can also use an answer file by typing **dcpromo /unattend:***“path to answer file”*, where the answer file contains unattended installation options and values. The following example contains the minimum parameters for an unattended installation of a new Windows Server 2008 domain controller in a new forest:

```
[DCINSTALL]
ReplicaOrNewDomain=domain
NewDomain=forest
NewDomainDNSName=fully qualified DNS name
DomainNetBiosName=domain NetBIOS name
ForestLevel={0=Windows 2000 Server Native;
             2=Windows Server 2003 Native;
             3=Windows Server 2008}
DomainLevel={0=Windows Server 2000 Native;
             2=Windows Server 2003 Native;
             3=Windows Server 2008}
InstallDNS=yes
DatabasePath="path to folder on a local volume"
LogPath="path to folder on a local volume"
SYSVOLPath="path to folder on a local volume"
SafeModeAdminPassword=password
RebootOnCompletion=yes
```

You can also specify one or more unattended installation parameters and values at the command line. For example, if you don't want the Directory Services Restore Mode password in the answer file, leave the entry blank and specify the */SafeModeAdminPassword:password* parameter when you run *Dcpromo.exe*.

You can also include all options on the command line itself. The following example creates the first domain controller in a new forest in which you don't expect to install any Windows Server 2003 domain controllers:

```
dcpromo /unattend /installDNS:yes /dnsOnNetwork:yes
        /replicaOrNewDomain:domain /newDomain:forest
        /newDomainDnsName:contoso.com /DomainNetbiosName:contoso
```

```
/databasePath:"e:\ntds" /logPath:"f:\ntdslogs" /sysvolpath:"g:\sysvol"  
/safeModeAdminPassword:password /forestLevel:3 /domainLevel:3  
/rebootOnCompletion:yes
```

Installing Additional Domain Controllers in a Domain

If you have a domain with at least one domain controller running Windows 2000 Server, Windows Server 2003, or Windows Server 2008, you can create additional domain controllers to distribute authentication, create a level of fault tolerance in the event any one DC fails, or provide authentication in remote sites.

Installing the First Windows Server 2008 Domain Controller in an Existing Forest or Domain

If you have an existing forest with domain controllers running Windows Server 2003 or Windows 2000 Server, you must prepare them prior to creating your first Windows Server 2008 domain controller. That's because there are objects and attributes that Windows Server 2008 adds to the directory that previous versions of Windows don't understand. Therefore, the schema must be updated. The schema is the definition of the attributes and object classes that can exist within a domain. It is like the catalog for what can be created in other directory partitions. To prepare the forest schema for Windows Server 2008, follow these steps:

1. Log on to the schema master as a member of the Enterprise Admins, Schema Admins, and Domain Admins groups.
Lesson 2, "Configuring Operations Masters," discusses operations masters and provides steps for identifying which domain controller is the schema master.
2. Copy the contents of the \Sources\Adprep folder from the Windows Server 2008 DVD to a folder on the schema master.
3. Open a command prompt and change directories to the Adprep folder.
4. Type **adprep /forestprep** and press Enter.
5. If you plan to install an RODC in any domain in the forest, type **adprep /rodcprep** and press Enter.

NOTE RODCPREP, anytime

You can also run *Adprep /rodcprep* at any time in a Windows 2000 Server or Windows Server 2003 forest. It does not have to be run in conjunction with */forestprep*; however, you must run it and allow its changes to replicate throughout the forest prior to installing the first RODC. You can run *Adprep /rodcprep* from any DC as long as you are logged on as a member of the Enterprise Admins group.

Exam Tip The *Adprep /rodcprep* command is required before installing an RODC into any domain in an existing forest with Windows Server 2003 or Windows 2000 Server domain controllers. It is not necessary if the forest is a new forest consisting only of Windows Server 2008 domain controllers.

You must allow time for the operation to complete. After the changes have replicated throughout the forest, you can continue to prepare the domains for Windows Server 2008. To prepare a Windows 2000 Server or Windows Server 2003 domain for Windows Server 2008, perform these steps:

1. Log on to the domain infrastructure operations master as a member of Domain Admins. Lesson 2 provides steps for identifying which domain controller is the infrastructure operations master.
2. Copy the contents of the `\Sources\Adprep` folder from the Windows Server 2008 DVD to a folder on the infrastructure master.
3. Open a command prompt and change directories to the Adprep folder.
4. Type **adprep /domainprep /gpprep** and press Enter.
On Windows Server 2003, you might receive an error message stating that updates were unnecessary. You can ignore this message.

Allow the change to replicate throughout the forest before you install a domain controller that runs Windows Server 2008.

Installing an Additional Domain Controller

Additional domain controllers can be added by installing AD DS and launching the Active Directory Domain Services Installation Wizard. You are prompted to choose the deployment configuration; to enter network credentials; to select a domain and site for the new DC; and to configure the DC with additional options such as DNS Server, Global Catalog, or Read-Only Domain Controller. The remaining steps are the same as for the first domain controller: configuring file locations and the Directory Services Restore Mode Administrator password.

If you have one domain controller in a domain, and if you select the Use Advanced Mode Installation check box on the Welcome To The Active Directory Domain Services Installation Wizard page, you are able to configure advanced options, which are:

- **Install From Media** By default, a new domain controller replicates all data for all directory partitions it will host from other domain controllers during the Active Directory Domain Services Installation Wizard. To improve the performance of installation, particularly over slow links, you can use installation media created by existing domain controllers. Installation media is a form of backup. The new DC is able to read data from the installation media directly and then replicate only updates from other

domain controllers. Install From Media (IFM) is discussed in the “Installing AD DS from Media” section.

- **Source Domain Controller** If you want to specify the domain controller from which the new DC replicates its data, you can click Use This Specific Domain Controller.

NOTE *Dcpromo /adv* is still supported

In Windows Server 2003, *Dcpromo /adv* was used to specify advanced installation options. The *adv* parameter is still supported; it simply pre-selects the Use Advanced Mode Installation check box on the Welcome page.

To use *Dcpromo.exe* with command-line parameters to specify unattended installation options, you can use the minimal parameters shown in the following example:

```
dcpromo /unattend /replicaOrNewDomain:replica
  /replicaDomainDNSName:contoso.com /installDNS:yes /confirmGC:yes
  /databasePath:"e:\ntds" /logPath:"f:\ntdslogs" /sysvolpath:"g:\sysvol"
  /safeModeAdminPassword:password /rebootOnCompletion:yes
```

If you are not logged on to the server with domain credentials, specify the *userdomain* and *username* parameters as well. A minimal answer file for an additional domain controller in an existing domain is as follows:

```
[DCINSTALL]
ReplicaOrNewDomain=replica
ReplicaDomainDNSName=FQDN of domain to join
UserDomain=FQDN of domain of user account
UserName=DOMAIN\username (in Administrators group of the domain)
Password=password for user specified by UserName (* to prompt)
InstallDNS=yes
ConfirmGC=yes
DatabasePath="path to folder on a local volume"
LogPath="path to folder on a local volume"
SYSVOLPath="path to folder on a local volume"
SafeModeAdminPassword=password
RebootOnCompletion=yes
```

Installing a New Windows Server 2008 Child Domain

If you have an existing domain, you can create a new child domain by creating a Windows Server 2008 domain controller. Before you do, however, you must run *Adprep /forestprep*, as described in the “Installing the First Windows Server 2008 Domain Controller in an Existing Forest or Domain” section.

Then install AD DS and launch the Active Directory Domain Services Installation Wizard and, on the Choose A Deployment Configuration page, click Existing Forest and Create A New Domain In An Existing Forest. You are prompted to select the domain functional level.

Because it is the first DC in the domain, it cannot be an RODC, and it cannot be installed from media. If you select the Use Advanced Mode Installation check box on the Welcome page, the wizard presents you with a Source Domain Controller page on which you specify a domain controller from which to replicate the configuration and schema partitions.

Using *Dcpromo.exe*, you can create a child domain with the minimal options shown in the following command:

```
dcpromo /unattend /installDNS:yes
  /replicaOrNewDomain:domain /newDomain:child
  /ParentDomainDNSName:contoso.com
  /newDomainDnsName:subsidiary.contoso.com /childName:subsidiary
  /DomainNetbiosName:subsidiary
  /databasePath:"e:\ntds" /logPath:"f:\ntdslogs" /sysvolpath:"g:\sysvol"
  /safeModeAdminPassword:password /forestLevel:3 /domainLevel:3
  /rebootOnCompletion:yes
```

The following answer file reflects the same minimal parameters:

```
[DCINSTALL]
ReplicaOrNewDomain=domain
NewDomain=child
ParentDomainDNSName=FQDN of parent domain
UserDomain=FQDN of user specified by UserName
UserName= DOMAIN\username (in Administrators group of ParentDomainDNSName)
Password=password for user specified by UserName or * for prompt
ChildName=single-label prefix for domain
      (Child domain FQDN will be ChildName.ParentDomainDNSName)
DomainNetBiosName=Domain NetBIOS name
DomainLevel=domain functional level (not lower than current forest level)
InstallDNS=yes
CreateDNSDelegation=yes
DNSDelegationUserName=DOMAIN\username with permissions to create
      DNS delegation, if different than UserName, above
DNSDelegationPassword=password for DNSDelegationUserName or * for prompt
DatabasePath="path to folder on a local volume"
LogPath="path to folder on a local volume"
SYSVOLPath="path to folder on a local volume"
SafeModeAdminPassword=password
RebootOnCompletion=yes
```

Installing a New Domain Tree

You learned in Chapter 1 that in an Active Directory forest, a tree is composed of one or more domains that share contiguous DNS namespace. So, for example, the *contoso.com* and *subsidiary.contoso.com* domains would be in a single tree. Additional trees are simply additional domains that are not in the same namespace. For example, if Contoso, Ltd., bought Tailspin Toys, the *tailspintoys.com* domain would be in a separate tree in the domain. There is very little functional difference between a child domain and a domain in another tree, and the process for creating a new tree is, therefore, very similar to creating a child domain.

First, you must run *Adprep /forestprep*, as described in the “Installing the First Windows Server 2008 Domain Controller in an Existing Forest or Domain” section. Then you can install AD DS and run the Active Directory Domain Services Installation Wizard. You must select Use Advanced Mode Installation on the Welcome page of the wizard. On the Choose A Deployment Configuration page, click Existing Forest, select Create A New Domain In An Existing Forest, and select Create A New Domain Tree Root Instead Of A New Child Domain. The rest of the process is identical to creating a new child domain.

The following options provided as parameters to *Dcpromo.exe* create a new tree for the *tailspintoys.com* domain within the *contoso.com* forest:

```
dcpromo /unattend /installDNS=yes
        /replicaOrNewDomain:domain /newDomain:tree
        /newDomainDnsName:tailspintoys.com /DomainNetbiosName:tailspintoys
        /databasePath:"e:\ntds" /logPath:"f:\ntdslogs" /sysvolpath:"g:\sysvol"
        /safeModeAdminPassword:password /domainLevel:2
        /rebootOnCompletion:yes
```

The domain functional level is configured at 2—Windows Server 2003 Native—so the domain could include Windows Server 2003 domain controllers. An unattended installation answer file that creates the same new tree would look similar to the following:

```
[DCINSTALL]
ReplicaOrNewDomain=domain
NewDomain=tree
NewDomainDNSName=FQDN of new domain
DomainNetBiosName=NetBIOS name of new domain
UserDomain=FQDN of user specified by UserName
UserName= DOMAIN\username (in Administrators group of ParentDomainDNSName)
Password=password for user specified by UserName or * for prompt
DomainLevel=domain functional level (not lower than current forest level)
InstallDNS=yes
ConfirmGC=yes
CreatedNSDNSDelegation=yes
DNSDelegationUserName=account with permissions to create DNS delegation
                        required only if different than UserName, above
DNSDelegationPassword=password for DNSDelegationUserName or * for prompt
DatabasePath="path to folder on a local volume"
LogPath="path to folder on a local volume"
SYSVOLPath="path to folder on a local volume"
SafeModeAdminPassword=password
RebootOnCompletion=yes
```

Staging the Installation of an RODC

As you remember from Chapter 8, “Authentication,” RODCs are designed to support branch office scenarios by providing authentication local to the site while mitigating the security and data integrity risks associated with placing a DC in a less well-controlled environment. Many

times, there are few or no IT support personnel in a branch office. How, then, should a domain controller be created in a branch office?

To answer this question, Windows Server 2008 enables you to create a staged, or delegated, installation of an RODC. The process includes two stages:

- **Create the account for the RODC** A member of Domain Admins creates an account for the RODC in Active Directory. The parameters related to the RODC are specified at this time: the name, the Active Directory site in which the RODC will be created, and, optionally, the user or group that can complete the next stage of the installation.
- **Attach the server to the RODC account** After the account has been created, AD DS is installed, and the RODC is attached to the domain. These steps can be the users or groups specified when the RODC account was prestaged; these users do not require any privileged group membership. A server can also be attached by a member of Domain Admins or Enterprise Admins, but the ability to delegate this stage to a nonprivileged user makes it much easier to deploy RODCs in branches without IT support. The domain controller will replicate its data from another writable DC in the domain, or you can use the IFM method discussed in the “Installing AD DS from Media” section.

NOTE Promote from a workgroup

When you create an RODC by using the staged approach—when you attach an RODC to a pre-staged account—the server must be a member of a workgroup, not of the domain, when you launch *Dcpromo.exe* or the Active Directory Domain Services Installation Wizard. The wizard will look in the domain for the existing account with its name and will attach to that account.

Creating the Prestaged Account for the RODC

To create the account for the RODC, using the Active Directory Users and Computers snap-in, right-click the Domain Controllers OU and choose Pre-Create Read-Only Domain Controller Account. A wizard appears that is very similar to the Active Directory Domain Services Installation Wizard. You are asked to specify the RODC name and site. You are also able to configure the password replication policy, as detailed in Chapter 8.

On the Delegation Of RODC Installation And Administration page, you can specify one security principal—user or group—that can attach the server to the RODC account you create. The user or group will also have local administrative rights on the RODC after the installation. It is recommended that you delegate to a group rather than to a user. If you do not specify a user or group, only members of the Domain Admins or Enterprise Admins groups can attach the server to the account.

MORE INFO Creating prestaged RODC accounts

You can create prestaged RODC accounts by using *Dcpromo.exe* with numerous parameters or by creating an answer file for *Dcpromo.exe*. The steps for doing so are detailed at <http://technet2.microsoft.com/windowsserver2008/en/library/f349e1e7-c3ce-4850-9e50-d8886c866b521033.aspx?mfr=true>.

Attaching a Server to the RODC Account

After you have prestaged the account, the server can be attached to it. You cannot simply launch the Active Directory Domain Services Installation Wizard. You must do so by typing **dcpromo /useexistingaccount:attach**. The wizard prompts for network credentials and then finds the RODC account in the domain indicated by the credentials. Remaining steps are similar to other domain controller promotion operations.

To use an answer file, provide the following options and values:

```
[DCINSTALL]
ReplicaDomainDNSName=FQDN of domain to join
UserDomain=FQDN of user specified by UserName
UserName=DOMAIN\username (in Administrators group of the domain)
Password=password for user specified by UserName
InstallDNS=yes
ConfirmGC=yes
DatabasePath="path to folder on a local volume"
LogPath="path to folder on a local volume"
SYSVOLPath="path to folder on a local volume"
SafeModeAdminPassword=password
RebootOnCompletion=yes
```

Run *Dcpromo* with the unattend:“answer file path” and the UseExistingAccount:Attach options, as in the following example:

```
dcpromo /useexistingaccount:attache /unattend:"c:\rodanswer.txt"
```

All the options just shown in the answer file can also be specified or overridden directly on the command line. Just type a command similar to the following:

```
dcpromo /unattend /UseExistingAccount:Attach /ReplicaDomainDNSName:contoso.com
  /UserDomain:contoso.com /UserName:contoso\dan /password:*
  /databasePath:"e:\ntds" /logPath:"f:\ntdslogs" /sysvolpath:"g:\sysvol"
  /safeModeAdminPassword:password /rebootOnCompletion:yes
```

Quick Check

- You administer a domain containing Windows Server 2003 domain controllers. You want to allow a manager at a remote site to promote a member server at a remote site to an RODC. You do not want to give the manager administrative credentials in the domain. What steps must you and the manager take?

Quick Check Answer

- You must run *Adprep /rodcrep* to prepare the domain for the RODC. You must then prestage the RODC account, delegating to the manager the ability to attach the domain controller to the account. The manager will run *Dcpromo.exe* with the *UseExistingAccount* option to attach the server, but first, the server must be removed from the domain and placed in a workgroup.

Installing AD DS from Media

When you add domain controllers to a forest, data from existing directory partitions are replicated to the new DC. In an environment with a large directory or where bandwidth is constrained between a new DC and a writable DC from which to replicate, you can install AD DS more efficiently by using the IFM option. Installing from media involves creating *installation media*—a specialized backup of Active Directory that can be used by the Active Directory Domain Services Installation Wizard as a data source for populating the directory on a new DC. Then the new DC will replicate only updates from another writable DC, so if the installation media is recent, you can minimize the impact of replication to a new DC.

Remember that it is not only the directory that must be replicated to a new DC but SYSVOL as well. When you create your installation media, you can specify whether to include SYSVOL on the installation media.

Using IFM also enables you to control the timing of impact to your network bandwidth. You can, for example, create installation media and transfer it to a remote site during off hours, then create the domain controller during normal business hours. Because the installation media is from the local site, impact to the network is reduced, and only updates will be replicated over the link to the remote site.

To create installation media, open a command prompt on a writable domain controller, running Windows Server 2008. The installation media is compatible across platforms. Run *Ntdsutil.exe* and then, at the *ntdsutil* prompt, type the **activate instance ntds** command and then the **ifm** command. At the *ifm:* prompt, type one of the following commands, based on the type of installation media you want to create:

- **create sysvol full path** Creates installation media with SYSVOL for a writable domain controller in the folder specified by *Path*

- **create full path** Creates installation media without SYSVOL for a writable domain controller or an Active Directory Lightweight Directory Services (AD LDS) instance in the folder specified by *Path*
- **create sysvol rodc path** Creates installation media with SYSVOL for a read-only domain controller in the folder specified by *Path*
- **create rodc path** Creates installation media without SYSVOL for a read-only domain controller in the folder specified by *Path*

When you run the Active Directory Domain Services Installation Wizard, select the Use Advanced Mode Installation check box, and you will be presented the Install From Media page later in the wizard. Choose Replicate Data From Media At The Following Location. You can use the ReplicationSourcePath installation option in an answer file or on the *Dcpromo.exe* command line.

Practice It Exercise 3, "Create Installation Media," in the practice at the end of this lesson, steps you through the process of creating installation media with *Ntdsutil.exe*.

Removing a Domain Controller

You can remove a domain controller by using *Dcpromo.exe*, either to launch the Active Directory Domain Services Installation Wizard or from a command prompt, specifying options at the command line or in an answer file. When a domain controller is removed while it has connectivity to the domain, it updates the forest metadata about the domain controller so that the directory knows the DC has been removed.

MORE INFO Removing a domain controller

For detailed steps for removing a domain controller, see <http://technet2.microsoft.com/windowsserver2008/en/library/9260bb40-a808-422f-b33b-c3d2330f5eb81033.mspx>.

If a domain controller must be demoted while it cannot contact the domain, you must use the *forceremoval* option of *Dcpromo.exe*. Type **dcpromo /forceremoval**, and the Active Directory Domain Services Installation Wizard steps you through the process. You are presented warnings related to any roles the domain controller hosts. Read each warning and, after you have mitigated or accepted the impact of the warning, click Yes. You can suppress warnings, using the *demotefsмо:yes* option of *Dcpromo.exe*. After the DC has been removed, you must manually clean up the forest metadata.

MORE INFO Performing metadata cleanup

See article 216498 in the Microsoft Knowledge Base for information about performing metadata cleanup. The article is located at <http://go.microsoft.com/fwlink/?LinkId=80481>.

Lesson 2: Configuring Operations Masters

In an Active Directory domain, all domain controllers are equivalent. They are all capable of writing to the database and replicating changes to other domain controllers. However, in any multimaster replication topology, certain operations must be performed by one and only one system. In an Active Directory domain, *operations masters* are domain controllers that play a specific role. Other domain controllers are capable of playing the role but do not. This lesson will introduce you to the five operations masters found in Active Directory forests and domains. You will learn their purposes, how to identify the operations masters in your enterprise, and the nuances of administering and transferring roles.

After this lesson, you will be able to:

- Define the purpose of the five single master operations in Active Directory forests.
- Identify the domain controllers performing operations master roles.
- Plan the placement of operations master roles.
- Transfer and seize operations master roles.

Estimated lesson time: 45 minutes

Understanding Single Master Operations

In any replicated database, some changes must be performed by one and only one replica because they are impractical to perform in a multimaster fashion. Active Directory is no exception. A limited number of operations are not permitted to occur at different places at the same time and must be the responsibility of only one domain controller in a domain or forest. These operations, and the domain controllers that perform them, are referred to by a variety of terms:

- *Operations masters*
- *Operations master roles*
- *Single master roles*
- *Operations tokens*
- *Flexible single master operations (FSMOs)*

Regardless of the term used, the idea is the same. One domain controller performs a function, and while it does, no other domain controller performs that function.

Not Déjà Vu

If you were an administrator in the days of Microsoft Windows NT 4.0, the concept of operations masters might sound similar to Windows NT primary domain controllers (PDCs). However, single master operations are characteristic of any replicated database, and Active Directory single master operations bear striking differences to Windows NT 4.0 PDCs:

- All Active Directory domain controllers are capable of performing single master operations. The domain controller that actually does perform an operation is the domain controller that currently holds the operation's token.
- An operation token, and thus the role, can be transferred easily to another domain controller without a reboot.
- To reduce the risk of single points of failure, the operations tokens can be distributed among multiple DCs.

AD DS contains five operations master roles. Two roles are performed for the entire forest:

- Domain naming
- Schema

Three roles are performed in each domain:

- Relative identifier (RID)
- Infrastructure
- PDC Emulator

Each of these roles is detailed in the following sections. In a forest with a single domain, there are, therefore, five operations masters. In a forest with two domains, there are eight operations masters because the three domain master roles are implemented separately in each of the two domains.

Exam Tip Commit to memory the list of forest-wide and domain single master operations. You are likely to encounter questions that test your knowledge of which roles apply to the entire forest and which are domain specific. Exam questions are cast in scenarios and, often, the scenarios provide so much detail that you can lose sight of what is really being asked. When you read items on the certification exam, always ask yourself, "What is really being tested?" Sometimes what is being tested is different from, and simpler than, what the scenario in the question would lead you to believe.

Forest-Wide Operations Master Roles

The schema master and the domain naming master must be unique in the forest. Each role is performed by only one domain controller in the entire forest.

Domain Naming Master Role

The domain naming role is used when adding or removing domains in the forest. When you add or remove a domain, the domain naming master must be accessible, or the operation will fail.

Schema Master Role

The domain controller holding the schema master role is responsible for making any changes to the forest's schema. All other DCs hold read-only replicas of the schema. If you want to modify the schema or install an application that modifies the schema, it is recommended you do so on the domain controller holding the schema master role. Otherwise, changes you request must be sent to the schema master to be written into the schema.

Domain-Wide Operations Master Roles

Each domain maintains three single master operations: RID, Infrastructure, and PDC Emulator. Each role is performed by only one domain controller in the domain.

RID Master Role

The RID master plays an integral part in the generation of security identifiers (SIDs) for security principals such as users, groups, and computers. The SID of a security principal must be unique. Because any domain controller can create accounts and, therefore, SIDs, a mechanism is necessary to ensure that the SIDs generated by a DC are unique. Active Directory domain controllers generate SIDs by assigning a unique RID to the domain SID. The RID master for the domain allocates pools of unique RIDs to each domain controller in the domain. Thus, each domain controller can be confident that the SIDs it generates are unique.

NOTE The RID master role is like DHCP for SIDs

If you are familiar with the concept that you allocate a scope of IP addresses for the Dynamic Host Configuration Protocol (DHCP) server to assign to clients, you can draw a parallel to the RID master, which allocates pools of RIDs to domain controllers for the creation of SIDs.

Infrastructure Master Role

In a multidomain environment, it is common for an object to reference objects in other domains. For example, a group can include members from another domain. Its multivalued

member attribute contains the distinguished names of each member. If the member in the other domain is moved or renamed, the infrastructure master of the group's domain updates the group's *member* attribute accordingly.

NOTE The infrastructure master

You can think of the infrastructure master as a tracking device for group members from other domains. When those members are renamed or moved in the other domain, the infrastructure master identifies the change and makes appropriate changes to group memberships so that the memberships are kept up to date.

Phantoms of the Directory

Although you are not expected to understand the internals of the infrastructure master role for the certification exam, such understanding can be helpful in the production environment. When you add a member from another domain into a group in your domain, the group's *member* attribute is appended with the distinguished name of the new member. However, your domain might not always have access to a domain controller from the member's domain, so Active Directory creates a phantom object to represent the member. The phantom object includes only the member's SID, distinguished name (DN), and globally unique identifier (GUID). If the member is moved or renamed in its domain, its GUID does not change, but its DN changes. If the object is moved between domains, its SID also changes. The infrastructure master periodically—every two days by default—contacts a GC or a DC in the member domain. At that time, the infrastructure master looks for each phantom object, using the GUID of the phantom object. It updates the DN of the phantom objects with the current DN of the object. Any change is then propagated to the *member* attribute of groups.

After a member is moved or renamed in another domain, and until the infrastructure master has updated DNs, you might look at the membership of a group using the Active Directory Users and Computers snap-in, for example, and the group might appear not to include that member. However, the member continues to belong to the group. The member's *memberOf* attribute still refers to the group, so the *memberOf* attribute and the *tokenGroups* constructed attribute are unchanged. There is no compromise to security; it is only an administrator looking at that particular group membership that would notice the temporary inconsistency.

PDC Emulator Role

The PDC Emulator role performs multiple, crucial functions for a domain:

- **Emulates a Primary Domain Controller (PDC) for backward compatibility** In the days of Windows NT 4.0 domains, only the PDC could make changes to the directory. Previous tools, utilities, and clients written to support Windows NT 4.0 are unaware that all Active Directory domain controllers can write to the directory, so such tools request a connection to the PDC. The domain controller with the PDC Emulator role registers itself as a PDC so that down-level applications can locate a writable domain controller. Such applications are less common now that Active Directory is nearly 10 years old, and if your enterprise includes such applications, work to upgrade them for full Active Directory compatibility.
- **Participates in special password update handling for the domain** When a user's password is reset or changed, the domain controller that makes the change replicates the change immediately to the PDC emulator. This special replication ensures that the domain controllers know about the new password as quickly as possible. If a user attempts to log on immediately after changing passwords, the domain controller responding to the user's logon request might not know about the new password. Before it rejects the logon attempt, that domain controller forwards the authentication request to a PDC emulator, which verifies that the new password is correct and instructs the domain controller to accept the logon request. This function means that any time a user enters an incorrect password, the authentication is forwarded to the PDC emulator for a second opinion. The PDC emulator, therefore, should be highly accessible to all clients in the domain. It should be a well-connected, high-performance DC.
- **Manages Group Policy updates within a domain** If a Group Policy object (GPO) is modified on two DCs at approximately the same time, there could be conflicts between the two versions that could not be reconciled as the GPO replicates. To avoid this situation, the PDC emulator acts as the focal point for all Group Policy changes. When you open a GPO in Group Policy Management Editor (GPME), the GPME binds to the domain controller performing the PDC emulator role. Therefore, all changes to GPOs are made on the PDC emulator by default.
- **Provides a master time source for the domain** Active Directory, Kerberos, File Replication Service (FRS), and DFS-R each rely on timestamps, so synchronizing the time across all systems in a domain is crucial. The PDC emulator in the forest root domain is the time master for the entire forest, by default. The PDC emulator in each domain synchronizes its time with the forest root PDC emulator. Other domain controllers in the domain synchronize their clocks against that domain's PDC emulator. All other domain members synchronize their time with their preferred domain controller. This hierarchical structure of time synchronization, all implemented through the Win32Time service, ensures consistency of time. Universal Time Coordinate (UTC) is synchronized, and the time displayed to users is adjusted based on the time zone setting of the computer.

MORE INFO Change the time service only one way

It is highly recommended to allow Windows to maintain its native, default time synchronization mechanisms. The only change you should make is to configure the PDC emulator of the forest root domain to synchronize with an extra time source. If you do not specify a time source for the PDC emulator, the System event log will contain errors reminding you to do so. See <http://go.microsoft.com/fwlink/?LinkId=91969>, and the articles it refers to, for more information.

- **Acts as the domain master browser** When you open Network in Windows, you see a list of workgroups and domains, and when you open a workgroup or domain, you see a list of computers. These two lists, called *browse lists*, are created by the Browser service. In each network segment, a master browser creates the browse list: the lists of workgroups, domains, and servers in that segment. The domain master browser serves to merge the lists of each master browser so that browse clients can retrieve a comprehensive browse list.

Placing Operations Masters

When you create the forest root domain with its first domain controller, all five operations master roles are performed by the domain controller. As you add domain controllers to the domain, you can transfer the operations master role assignments to other domain controllers to balance the load among domain controllers or to optimize placement of a single master operation. The best practices for the placement of operations master roles are as follows:

- **Co-locate the schema master and domain naming master** The schema master and domain naming master roles should be placed on a single domain controller that is a GC server. These roles are rarely used, and the domain controller hosting them should be tightly secured. The domain naming master must be hosted on a GC server because when a new domain is added, the master must ensure that there is no object of any type with the same name as the new domain. The GC's partial replica contains the name of every object in the forest. The load of these operations master roles is very light unless schema modifications are being made.
- **Co-locate the RID master and PDC Emulator roles** Place the RID and PDC Emulator roles on a single domain controller. If the load mandates that the roles be placed on two separate domain controllers, those two systems should be physically well connected and have explicit connection objects created in Active Directory so that they are direct replication partners. They should also be direct replication partners with domain controllers that you have selected as standby operations masters.
- **Place the infrastructure master on a DC that is not a GC** The infrastructure master should be placed on a domain controller that is not a GC server but is physically well connected to a GC server. The infrastructure master should have explicit connection objects in Active Directory to that GC server so that they are direct replication partners.

The infrastructure master can be placed on the same domain controller that acts as the RID master and PDC emulator.

NOTE It doesn't matter if they're all GCs

If all DCs in a domain are GC servers—which indeed is a best practices recommendation that will be discussed in Chapter 11, “Sites and Replication”—you do not need to worry about which DC is the infrastructure master. When all DCs are GCs, all DCs have up-to-date information about every object in the forest, which eliminates the need for the infrastructure master role.

- **Have a failover plan** In following sections, you will learn to transfer single operations master roles between domain controllers, which is necessary if there is lengthy planned or unplanned downtime of an operations master. Determine, in advance, a plan for transferring operations roles to other DCs in the event that one operations master is offline.

Identifying Operations Masters

To implement your role placement plan, you must know which DCs are currently performing single master operations roles. Each role is exposed in an Active Directory administrative tool as well as in other user interface and command-line tools. To identify the current master for each role, use the following tools:

- **PDC Emulator: The Active Directory Users And Computers snap-in** Right-click the domain and choose Operations Masters. Click the PDC tab. An example is shown in Figure 10-2, which indicates that SERVER01.contoso.com is currently the PDC operations master.

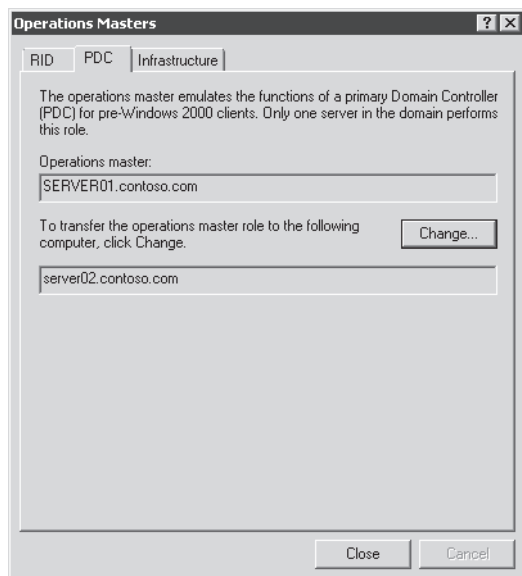


Figure 10-2 PDC Operations Master

- **RID Master: The Active Directory Users And Computers snap-in** Right-click the domain and choose Operations Masters. Click the RID tab.
- **Infrastructure Master: The Active Directory Users And Computers snap-in** Right-click the domain and choose Operations Masters. Click the Infrastructure tab.
- **Domain Naming: The Active Directory Domains And Trusts snap-in** Right-click the root node of the snap-in (Active Directory Domains And Trusts) and choose Operations Master.
- **Schema Master: The Active Directory Schema snap-in** Right-click the root node of the snap-in (Active Directory Schema) and choose Operations Master.

NOTE Registering the Active Directory Schema snap-in

You must register the Active Directory Schema snap-in before you can create a custom Microsoft Management Console (MMC) with the snap-in. At a command prompt, type **regsvr32 schmmgmt.dll**.

You can also use several other tools to identify operations masters, including the following commands:

```
ntdsutil
roles
connections
connect to server DomainControllerFQDN:portnumber
quit
select operation target
list roles for connected server"
quit
quit
quit

dcdiag /test:knowsofroleholders /v

netdom query fsmo
```

Practice It Exercise 1, "Identify Operations Masters," in the practice at the end of this lesson, steps you through the identification of operations masters.

Transferring Operations Master Roles

You can transfer a single operations master role easily. You will transfer roles in the following scenarios:

- When you establish your forest, all five roles are performed by the first domain controller you install. When you add a domain to the forest, all three domain roles are performed by the first domain controller in that domain. As you add domain controllers, you can distribute the roles to reduce single-point-of-failure instances and improve performance.

- If you plan to take a domain controller offline that is currently holding an operations master role, transfer that role to another domain controller prior to taking it offline.
- If you are decommissioning a domain controller that currently holds an operations master role, transfer that role to another domain controller prior to decommissioning. The Active Directory Domain Services Installation Wizard will attempt to do so automatically, but you should prepare for demoting a domain controller by transferring its roles.

To transfer an operations master role, follow these steps:

1. Open the administrative tool that exposes the current master.
For example, open the Active Directory Users And Computers snap-in to transfer any of the three domain master roles.
2. Connect to the domain controller to which you are transferring the role.
This is accomplished by right-clicking the root node of the snap-in and choosing Change Domain Controller or Change Active Directory Domain Controller. (The command differs between snap-ins.)
3. Open the Operations Master dialog box, which will show you the domain controller currently holding the role token for the operation. Click the Change button to transfer the role to the domain controller to which you are connected.

Practice It Exercise 2, “Transfer an Operations Master Role,” in the practice at the end of this lesson, steps you through the transfer of an operations master role.

When you transfer an operations master role, both the current master and the new master are online. The token is transferred, the new master immediately begins to perform the role, and the former master immediately ceases to perform the role. This is the preferred method of moving operations master roles.

It is recommended to make sure that the new role holder is up to date with replication from the former role holder before transferring the role. You can use skills introduced in Chapter 11 to force replication between the two systems.

Recognizing Operations Master Failures

Several operations master roles can be unavailable for quite some time before their absence becomes a problem. Other master roles play a crucial role in the day-to-day operation of your enterprise. You can identify problems with operations masters by examining the Directory Service event log.

However, you will often discover that an operations master has failed when you attempt to perform a function managed by the master, and the function fails. For example, if the RID master fails, eventually you will be prevented from creating new security principals.

Seizing Operations Master Roles

If a domain controller performing a single master operation fails, and you cannot bring the system back to service, you have the option of seizing the operations token. When you seize a role, you designate a new master without gracefully removing the role from the failed master.

Seizing a role is a drastic action, so before seizing a role, think carefully about whether it is necessary. Determine the cause and expected duration of the offline operations master. If the operations master can be brought online in sufficient time, wait. What is sufficient time? It depends on the impact of the role that has failed:

- **PDC emulator failure** The PDC emulator is the operations master that will have the most immediate impact on normal operations and on users if it becomes unavailable. Fortunately, the PDC Emulator role can be seized to another domain controller and then transferred back to the original role holder when the system comes back online.
- **Infrastructure master failure** A failure of the infrastructure master will be noticeable to administrators but not to users. Because the master is responsible for updating the names of group members from other domains, it can appear as if group membership is incorrect although, as mentioned earlier in this lesson, membership is not actually affected. You can seize the infrastructure master role to another domain controller and then transfer it back to the previous role holder when that system comes online.
- **RID master failure** A failed RID master will eventually prevent domain controllers from creating new SIDs and, therefore, will prevent you from creating new accounts for users, groups, or computers. However, domain controllers receive a sizable pool of RIDs from the RID master, so unless you are generating numerous new accounts, you can often go for some time without the RID master online while it is being repaired. Seizing this role to another domain controller is a significant action. After the RID master role has been seized, the domain controller that had been performing the role cannot be brought back online.
- **Schema master failure** The schema master role is necessary only when schema modifications are being made, either directly by an administrator or by installing an Active Directory integrated application that changes the schema. At other times, the role is not necessary. It can remain offline indefinitely until schema changes are necessary. Seizing this role to another domain controller is a significant action. After the schema master role has been seized, the domain controller that had been performing the role cannot be brought back online.
- **Domain naming master failure** The domain naming master role is necessary only when you add a domain to the forest or remove a domain from a forest. Until such changes are required to your domain infrastructure, the domain naming master role can remain offline for an indefinite period of time. Seizing this role to another domain controller is a significant action. After the domain naming master role has been seized, the domain controller that had been performing the role cannot be brought back online.

Although you can transfer roles by using the administrative tools, you must use *Ntdsutil.exe* to seize a role. To seize an operations master role, perform the following steps:

1. From the command prompt, type **ntdsutil** and press Enter.
2. At the ntdsutil prompt, type **roles** and press Enter.
The next steps establish a connection to the domain controller you want to perform the single master operation role.
3. At the fsmo maintenance prompt, type **connections** and press Enter.
4. At the server connections prompt, type **connect to server *DomainControllerFQDN*** and press Enter.
DomainControllerFQDN is the FQDN of the domain controller you want to perform the role.
Ntdsutil responds that it has connected to the server.
5. At the server connections prompt, type **quit** and press Enter.
6. At the fsmo maintenance prompt, type **seize *role*** and press Enter.
Role is one of the following:
 - a. schema master
 - b. domain naming master
 - c. RID master
 - d. PDC
 - e. infrastructure master
7. At the fsmo maintenance prompt, type **quit** and press Enter.
8. At the ntdsutil prompt, type **quit** and press Enter.

Returning a Role to Its Original Holder

To provide for planned downtime of a domain controller if a role has been transferred, not seized, the role can be transferred back to the original domain controller.

If, however, a role has been seized and the former master is able to be brought back online, you must be very careful. The PDC emulator and infrastructure master are the only operations master roles that can be transferred back to the original master after having been seized.

NOTE Do not return a seized schema, domain naming, or RID master to service

After seizing the schema, domain naming, or RID roles, you must completely decommission the original domain controller.

If you have seized the schema, domain naming, or RID roles to another domain controller, you must not bring the original domain controller back online without first completely

decommissioning it. That means you must keep the original role holder physically disconnected from the network, and you must remove AD DS by using the *Dcpromo /forceremoval* command. You must also clean the metadata for that domain controller as described in <http://go.microsoft.com/fwlink/?LinkId=80481>.

After the domain controller has been completely removed from Active Directory, if you want the server to rejoin the domain, you can connect it to the network and join the domain. If you want it to be a domain controller, you can promote it. If you want it to resume performing the operations master role, you can transfer the role back to the DC.

NOTE Better to rebuild

Because of the critical nature of domain controllers, it is recommended that you completely reinstall the former domain controller in this scenario.

Quick Check

- You need to upgrade the power supply and motherboard of SERVER01, the domain controller performing the PDC Emulator operations master role. You want to ensure continuity of services provided by the PDC emulator. Describe the process of transferring the role to SERVER02, another domain controller, and transferring it back after SERVER01 has been upgraded. Which tools will you use, and which steps will you perform?

Quick Check Answer

- Prior to performing the upgrade, make sure the standby operations master is up to date with replication from the PDC emulator. Then open the Active Directory Users And Computers snap-in, right-click the domain, and choose Change Domain Controller. Select SERVER02. Right-click the domain and choose Operations Masters. Click the PDC tab and click Change. The role is transferred. When SERVER01 comes back online, right-click the domain, choose Change Domain Controller, and select SERVER01. Right-click the domain, choose Operations Masters, click the PDC tab, and click Change.

Lesson 3: Configuring DFS Replication of SYSVOL

SYSVOL, a folder located at %SystemRoot%\SYSVOL by default, contains logon scripts, group policy templates (GPTs), and other resources critical to the health and management of an Active Directory domain. Ideally, SYSVOL should be consistent on each domain controller. However, changes to Group Policy objects and to logon scripts are made from time to time, so you must ensure that those changes are replicated effectively and efficiently to all domain controllers. In previous versions of Windows, the FRS was used to replicate the contents of SYSVOL between domain controllers. FRS has limitations in both capacity and performance that cause it to break occasionally. Unfortunately, troubleshooting and configuring FRS is quite difficult. In Windows Server 2008 domains, you have the option to use DFS-R to replicate the contents of SYSVOL. In this lesson, you will learn how to migrate SYSVOL from FRS to DFS-R.

After this lesson, you will be able to:

- Raise the domain functional level.
- Migrate SYSVOL replication from FRS to DFS-R.

Estimated lesson time: 60 minutes

Raising the Domain Functional Level

In Chapter 12, “Domains and Forests,” you will learn about forest and domain functional levels. A domain’s functional level is a setting that both restricts the operating systems that are supported as domain controllers in a domain and enables additional functionality in Active Directory. A domain with a Windows Server 2008 domain controller can be at one of three functional levels: Windows 2000 Native, Windows Server 2003 Native, and Windows Server 2008. At Windows 2000 Native domain functional level, domain controllers can be running Windows 2000 Server or Windows Server 2003. At Windows Server 2003 Native domain functional level, domain controllers can be running Windows Server 2003. At Windows Server 2008 domain functional level, all domain controllers must be running Windows Server 2008.

As you raise functional levels, new capabilities of Active Directory are enabled. At Windows Server 2008 domain functional level, for example, you can use DFS-R to replicate SYSVOL. Simply upgrading all domain controllers to Windows Server 2008 is not enough: You must specifically raise the domain functional level. You do this by using Active Directory Domains and Trusts. Right-click the domain and choose Raise Domain Functional Level. Then select Windows Server 2008 as the desired functional level and click Raise. After you’ve set the domain functional level to Windows Server 2008, you cannot add domain controllers running Windows Server 2003 or Windows 2000 Server. The functional level is associated only with domain controller operating systems; member servers and workstations

can be running Windows Server 2003, Windows 2000 Server, Windows Vista, Windows XP, or Windows 2000 Workstation.

Quick Check

- You are the administrator of Northwind Traders. The domain consists of three domain controllers. You have upgraded two of them to Windows Server 2008. The third is still running Windows Server 2003. You want to establish DFS-R as the replication mechanism for SYSVOL. What must you do?

Quick Check Answer

- You must upgrade the third domain controller to Windows Server 2008 and then raise the domain functional level to Windows Server 2008.

Understanding Migration Stages

Because SYSVOL is critical to the health and functionality of your domain, Windows does not provide a mechanism with which to convert replication of SYSVOL from FRS to DFS-R instantly. In fact, migration to DFS-R involves creating a parallel SYSVOL structure. When the parallel structure is successfully in place, clients are redirected to the new structure as the domain's system volume. When the operation has proven successful, you can eliminate FRS.

Migration to DFS-R thus consists of four stages or *states*:

- **0 (start)** The default state of a domain controller. Only FRS is used to replicate SYSVOL.
- **1 (prepared)** A copy of SYSVOL is created in a folder called SYSVOL_DFSR and is added to a replication set. DFS-R begins to replicate the contents of the SYSVOL_DFSR folders on all domain controllers. However, FRS continues to replicate the original SYSVOL folders and clients continue to use SYSVOL.
- **2 (redirected)** The SYSVOL share, which originally refers to SYSVOL\sysvol, is changed to refer to SYSVOL_DFSR\sysvol. Clients now use the SYSVOL_DFSR folder to obtain logon scripts and Group Policy templates.
- **3 (eliminated)** Replication of the old SYSVOL folder by FRS is stopped. The original SYSVOL folder is not deleted, however, so if you want to remove it entirely, you must do so manually.

You move your domain controllers through these stages, using the *Dfsrmig.exe* command. You will use three options with *Dfsrmig.exe*:

- **setglobalstate state** The *setglobalstate* option configures the current global DFSR migration state, which applies to all domain controllers. The state is specified by the *state* parameter, which is 0–3. Each domain controller will be notified of the new DFSR migration state and will migrate to that state automatically.

- **getglobalstate** The `getglobalstate` option reports the current global DFSR migration state.
- **getmigrationstate** The `getmigrationstate` option reports the current migration state of each domain controller. Because it might take time for domain controllers to be notified of the new global DFSR migration state, and because it might take even more time for a DC to make the changes required by that state, DCs will not be synchronized with the global state instantly. The `getmigrationstate` option enables you to monitor the progress of DCs toward the current global DFSR migration state.

If there is a problem moving from one state to the next higher state, you can revert to previous states by using the `setglobalstate` option. However, after you have used the `setglobalstate` option to specify state 3 (eliminated), you cannot revert to earlier states.

Migrating SYSVOL Replication to DFS-R

To migrate SYSVOL replication from FRS to DFS-R, perform the following steps:

1. Open the Active Directory Domains And Trusts snap-in.
2. Right-click the domain and choose Raise Domain Functional Level.
3. If the Current Domain Functional Level box does not indicate Windows Server 2008, choose Windows Server 2008 from the Select An Available Domain Functional Level list.
4. Click Raise. Click OK twice in response to the dialog boxes that appear.
5. Log on to a domain controller and open a command prompt.
6. Type **`dfsrmig /setglobalstate 1`**.
7. Type **`dfsrmig /getmigrationstate`** to query the progress of DCs toward the Prepared global state. Repeat this step until the state has been attained by all DCs.
This can take 15 minutes to an hour or longer.
8. Type **`dfsrmig /setglobalstate 2`**.
9. Type **`dfsrmig /getmigrationstate`** to query the progress of DCs toward the Redirected global state. Repeat this step until the state has been attained by all DCs.
This can take 15 minutes to an hour or longer.
10. Type **`dfsrmig /setglobalstate 3`**.
After you begin migration from state 2 (prepared) to state 3 (replicated), any changes made to the SYSVOL folder will have to be replicated manually to the SYSVOL_DFSR folder.
11. Type **`dfsrmig /getmigrationstate`** to query the progress of DCs toward the Eliminated global state. Repeat this step until the state has been attained by all DCs. This can take 15 minutes to an hour or longer.

For more information about the `Dfsrmig.exe` command, type **`dfsrmig.exe /?`**.

Chapter 11

Sites and Replication

You've learned in previous chapters that domain controllers (DCs) in a Windows Server 2008 domain are peers. Each maintains a copy of the directory, each performs similar services to support authentication of security principals, and changes made on any one domain controller will be replicated to all other domain controllers. As an administrator of a Microsoft Windows enterprise, one of your tasks is to ensure that authentication is provided as efficiently as possible, and that replication between domain controllers is optimized. Active Directory Domain Services (AD DS) sites are the core component of the directory service that supports the goals of service localization and replication. In this chapter, you will learn how to create a distributed directory service that supports domain controllers in portions of your network that are separated by expensive, slow, or unreliable links. You'll learn where domain controllers should be placed and how to manage replication and service usage. You'll also learn how to control which data is replicated to each domain controller by configuring global catalogs (GCs) and application partitions.

Exam objectives in this chapter:

- Configuring the Active Directory Infrastructure
 - Configure the global catalog.
 - Configure sites.
 - Configure Active Directory replication.
- Maintaining the Active Directory Environment
 - Monitor Active Directory.

Before You Begin

To complete the practices in this chapter, you must have created two domain controllers named SERVER01 and SERVER02 in a domain named *contoso.com*. See Chapter 1, “Installation,” and Chapter 10, “Domain Controllers,” for detailed steps for this task.

Real World

Dan Holme

As you learned in the previous chapter, it is important to have more than one domain controller in a domain to provide continuity of service in the event that one domain controller fails. That rule of thumb—at least two DCs per domain—assumes that all servers and clients in your environment are well connected to the DCs. But what happens if you have several network locations separated by links that are not LAN speed? And what must you do if those intersite links are unreliable? Well, then, you must make a determination whether to place domain controllers in remote locations and how to manage replication of the directory to those domain controllers. On the 70-640 exam, the focus on Active Directory sites is their relationship to replication, and you will certainly learn how to manage replication in this chapter. But sites are also important in highly connected environments because they enable you to manage *service localization*—that is, ensure that when a service is available from multiple servers, a client uses the most efficient server. Throughout this chapter, keep your eye on the relationship between sites and service localization as well, because although it might not be as important on the exam, it is certainly important in your production environment.

Lesson 1: Configuring Sites and Subnets

Active Directory represents human beings by user objects in the directory service. It represents machines by computer objects. It represents network topology with objects called *sites* and *subnets*. Active Directory site objects are used to manage replication and service localization and, fortunately, in many environments, the configuration of sites and subnets can be quite straightforward. In this lesson, you will learn the fundamental concepts and techniques required to configure and manage sites and subnets.

After this lesson, you will be able to:

- Identify the roles of sites and subnets.
- Describe the process with which a client locates a domain controller.
- Configure sites and subnets.
- Manage domain controller server objects in sites.

Estimated lesson time: 45 minutes

Understanding Sites

When administrators describe their network infrastructure, they often mention how many sites comprise their enterprise. To most administrators, a site is a physical location, an office or city, for example. Sites are connected by links—network links that might be as basic as dial-up connections or as sophisticated as fiber links. Together, the physical locations and links make up the network infrastructure.

Active Directory represents the network infrastructure with objects called *sites* and *site links*, and although the words are similar, these objects are not identical to the sites and links described by administrators. This lesson focuses on sites, and Lesson 3, “Configuring Replication,” discusses site links.

It’s important to understand the properties and roles of sites in Active Directory to understand the subtle distinction between Active Directory sites and network sites. Active Directory sites are objects in the directory, specifically in the Configuration container (CN=Configuration,DC=*forest root domain*). These objects are used to achieve two service management tasks:

- To manage replication traffic
- To facilitate service localization

Replication Traffic

Replication is the transfer of changes between domain controllers. When you add a user or change a user's password, for example, the change you make is committed to the directory by one domain controller. That change must be communicated to all other domain controllers in the domain.

Active Directory assumes there are two types of networks within your enterprise: highly connected and less highly connected. Conceptually, a change made to Active Directory should replicate immediately to other domain controllers within the highly connected network in which the change was made. However, you might not want the change to replicate immediately over a slower, more expensive, or less reliable link to another site. Instead, you might want to manage replication over less highly connected segments of your enterprise to optimize performance, reduce costs, or manage bandwidth.

An Active Directory site represents a highly connected portion of your enterprise. When you define a site, the domain controllers within the site replicate changes almost instantly. Replication between sites can be scheduled and managed.

Service Localization

Active Directory is a distributed service. That is, assuming you have at least two domain controllers, there are multiple servers (domain controllers) providing the same services of authentication and directory access. If you have more than one network site, and if you place a domain controller in each, you want to encourage clients to authenticate against the domain controller in their site. This is an example of service localization.

Active Directory sites help you localize services, including those provided by domain controllers. During logon, Windows clients are automatically directed to a domain controller in their site. If a domain controller is not available in their site, they are directed to a DC in another site that will be able to authenticate the client efficiently.

Other services can be localized as well. Distributed File System Namespaces (DFS Namespaces), for example, is a localized service. DFS clients will obtain replicated resources from the most efficient server, based on their Active Directory site. In fact, because clients know what site they are in, any distributed service could be written to take advantage of the Active Directory site structure to provide intelligent localization of service usage.

Planning Sites

Because sites are used to optimize replication and to enable service localization, you must spend time designing your Active Directory site structure. Active Directory sites might not map one to one with your network's sites. Consider two scenarios:

- You have offices in two distinct locations. You place one domain controller in each location. The locations are highly connected, and to improve performance, you decide to configure a single Active Directory site that includes both locations.
- You have an enterprise on a large, highly connected campus. From a replication perspective, the enterprise could be considered a single site. However, you want to encourage clients to use distributed services in their location, so you configure multiple sites to support service localization.

Therefore, an Active Directory site can include more than one network site or be a subset of a single network site. The key is to remember that sites serve both replication management and service localization roles. Several characteristics of your enterprise can be used to help you determine which sites are necessary:

Connection Speed

An Active Directory site represents a unit of the network that is characterized by fast, reliable, inexpensive connectivity. Much documentation suggests that the slowest link speed within a site should be no less than 512 kilobits per second (kbps). However, this guidance is not immutable. Some organizations have links as slow as 56 or even 28 kbps within a site.

Service Placement

Because Active Directory sites manage Active Directory replication and service localization, it is not useful to create a site for a network location that does not host a domain controller or other Active Directory-aware service such as a replicated DFS resource.

NOTE Sites where there are no domain controllers

Domain controllers are only one distributed service in a Windows enterprise. Other services, such as replicated DFS resources, are site-aware as well. You might configure sites to localize services other than authentication, in which case you will have sites without domain controllers.

User Population

Concentrations of users can also influence your site design, although indirectly. If a network location has a sufficient number of users for whom the inability to authenticate would be problematic, place a domain controller in the location to support authentication within the location. After a domain controller or other distributed service is placed in the location to support those users, you might want to manage Active Directory replication to the location or localize service use by configuring an Active Directory site to represent the location.

Summarizing Site Planning Criteria

Every Active Directory forest includes at least one site. The default site created when you instantiate a forest with the first domain controller is creatively named *Default-First-Site-Name*. You should create additional sites when:

- A part of the network is separated by a slow link.
- A part of the network has enough users to warrant hosting domain controllers or other services in that location.
- Directory query traffic warrants a local domain controller.
- You want to control service localization.
- You want to control replication between domain controllers.

Server Placement

Network administrators often want to know when placing a domain controller in a remote site is recommended. The answer is, “It depends.” Specifically, it depends on the resources required by users in the site and the tolerance for downtime. If users in a remote site perform all work tasks by accessing resources in the data center, for example, then if the link to the remote site fails, the users cannot access the resources they require, and a local domain controller would not improve the situation. However, if users access resources in the remote site and the link fails, a local domain controller can continue to provide authentication for users and they can continue to work with their local resources.

In most branch office scenarios, there are resources in the branch office that users require to perform their work tasks. Those resources, if not stored on the user’s own computer, require domain authentication of the user. Therefore, a domain controller is generally recommended. The introduction of read-only domain controllers (RODCs) in Windows Server 2008 reduces the risk and management burden of domain controllers in branch offices, so it will be easier for most organizations to deploy DCs in each network location.

Defining Sites

Sites and replication are managed using the Active Directory Sites and Services snap-in. To define an Active Directory site, you will create an object of class *site*. The site object is a container that manages replication for domain controllers in the site. You will also create one or

more subnet objects. A subnet object defines a range of IP addresses and is linked to one site. Service localization is attained when a client's IP address can be associated with a site through the relationship between the subnet object and the site object.

You can create a site object by right-clicking the Sites node in Active Directory Sites And Services and choosing New Site. In the New Object – Site dialog box that appears, shown in Figure 11-1, enter a site name and select a site link. The default site link, DEFAULTIPSITELINK, will be the only site link available to you until you create additional site links as discussed in Lesson 2, “Configuring the Global Catalog and Application Directory Partitions.”

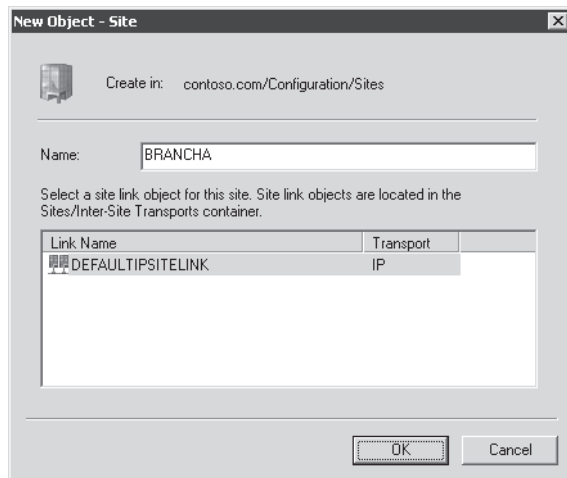


Figure 11-1 The New Object – Site dialog box

After creating a site, you can right-click it and choose Rename to rename it. It is recommended that you rename the Default-First-Site-Name site to reflect a site name that is aligned with your business and network topology.

Sites are useful only when a client or server knows the site to which it belongs. This is typically achieved by associating the system's IP address with a site, and subnet objects achieve this association. To create a subnet object, right-click the Subnets node in the Active Directory Sites And Services snap-in and choose New Subnet. The New Object – Subnet dialog box shown in Figure 11-2 appears. The subnet object is defined as a range of addresses using network prefix notation. For example, to enter a subnet representing the addresses 10.1.1.1 to 10.1.1.254 with a 24-bit subnet mask, the prefix would be 10.1.1.0/24. For more information about entering addresses, click the Learn More About Entering Address Prefixes link in the New Object – Subnet dialog box.

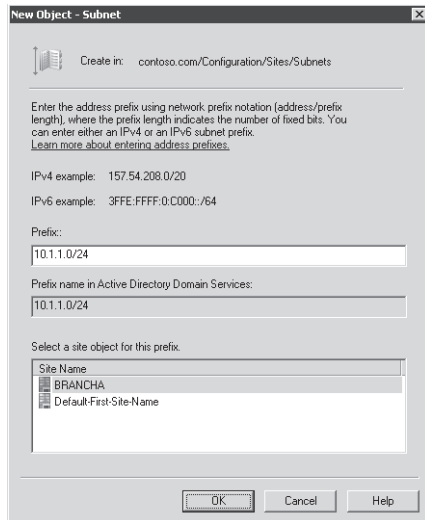


Figure 11-2 The New Object – Subnet dialog box

After entering the network prefix, select the site object with which the subnet is associated. A subnet can be associated with only one site; however, a site can have more than one subnet linked to it. The Properties dialog box of a site, shown in Figure 11-3, shows the subnets associated with the site. You cannot change the subnets in this dialog box, however; instead, you must open the properties of the subnet, shown in Figure 11-4, to change the site to which the subnet is linked.

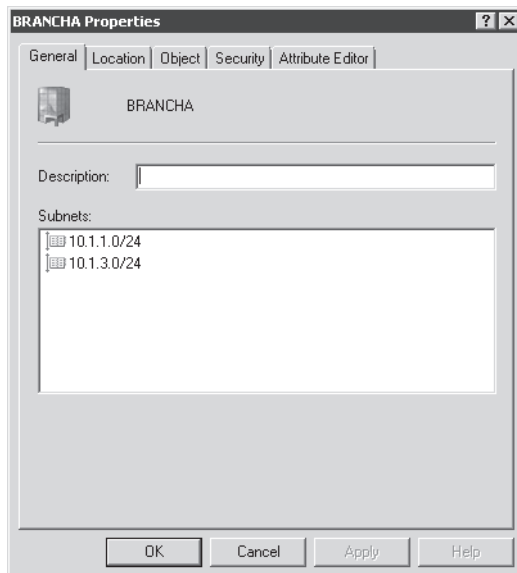


Figure 11-3 The Properties dialog box for a site

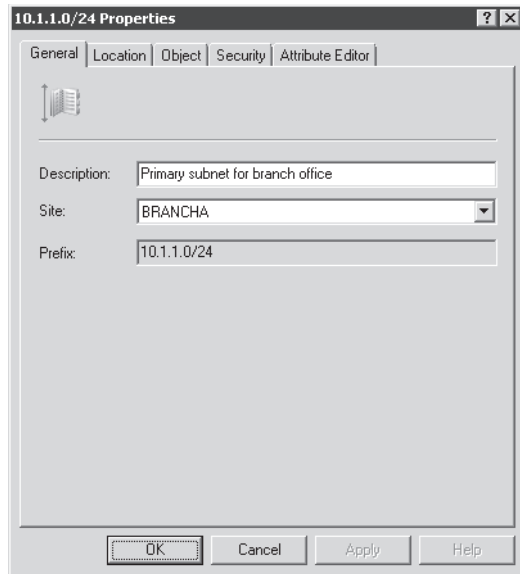


Figure 11-4 The Properties dialog box for a subnet

NOTE Defining every IP subnet

In your production environment, be certain to define every IP subnet as an Active Directory subnet object. If a client's IP address is not included within a subnet range, the client is unable to determine which Active Directory site it belongs to, which can lead to performance and functionality problems. Don't forget backbone subnets and subnets used for remote access such as virtual private network address ranges.

Managing Domain Controllers in Sites

There are times when you might need to manage domain controllers in Active Directory sites:

- You create a new site and move an existing domain controller to it.
- You demote a domain controller.
- You promote a new domain controller.

When you create your Active Directory forest, the first domain controller is automatically placed under the site object named *Default-First-Site-Name*. You can see the domain controller SERVER01.contoso.com in Figure 11-5. Additional domain controllers will be added to sites based on their IP addresses. For example, if a server with IP address 10.1.1.17 is promoted to a domain controller in the network shown in Figure 11-4, the server will automatically be added to the BRANCHA site. Figure 11-5 shows SERVER02 in the BRANCHA site.

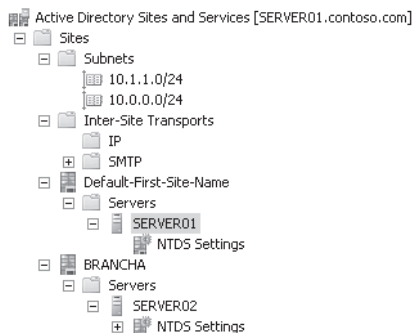


Figure 11-5 A domain controller in a site

Each site contains a Servers container, which itself contains an object for each domain controller in the site. The Servers container in a site should show only domain controllers, not all servers. When you promote a new domain controller, the domain controller will, by default, be placed in the site associated with its IP address. However, the Active Directory Domain Services Installation Wizard will enable you to specify another site. You can also pre-create the server object for the domain controller in the correct site by right-clicking the Servers container in the appropriate site and choosing Server from the New menu.

Finally, you can move the domain controller to the correct site after installation by right-clicking the server and choosing Move. In the Move Server dialog box, select the new site and click OK. The domain controller is moved. It is a best practice to place a domain controller in the site object that is associated with the DC's IP address. If a DC is multihomed, it can belong to only one site. If a site has no domain controllers, users will still be able to log on to the domain; their logon requests will be handled by a domain controller in an adjacent site or another domain controller in the domain.

To remove a domain controller object, right-click it and choose Delete.

Understanding Domain Controller Location

You started this lesson by examining AD DS as a distributed service, providing authentication and directory access on more than one domain controller. You learned to identify where, in your network topology, to define sites and place domain controllers. Now you are ready to examine how, exactly, service localization works—how Active Directory clients become site aware and locate the domain controller in their site. Although this level of detail is unlikely to appear on the certification examination, it can be extremely helpful when you need to troubleshoot authentication of a computer or of a user.

Service Locator Records

When a domain controller is added to the domain, it advertises its services by creating Service Locator (SRV) records, also called locator records, in DNS. Unlike host records (A records), which map host names to IP addresses, SRV records map services to host names. The domain controller advertises its ability to provide authentication and directory access by registering Kerberos and LDAP SRV records. These SRV records are added to several folders within the DNS zones for the forest. The first folder is within the domain zone. It is called `_tcp` and it contains the SRV records for all domain controllers in the domain. The second folder is specific to this site, in which the domain controller is located, with the path `_sites\sitename_tcp`, where *sitename* is the name of the site. In Figure 11-6, you can see the Kerberos and LDAP SRV records for SERVER02.contoso.com in its site, `_sites\BRANCHA_tcp`. You can also see the `_tcp` folder at the first level beneath the zone.

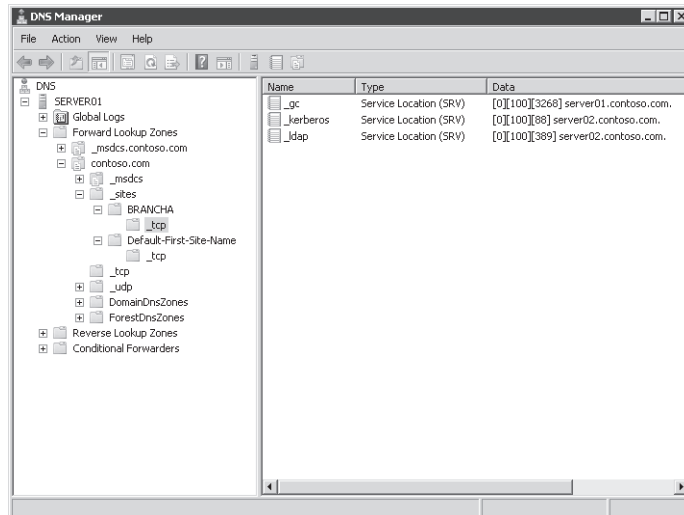


Figure 11-6 The SRV records for SERVER02 in the BRANCHA site

The same records are registered in several places in the `_msdcs.domainName` zone, for example, `_msdcs.contoso.com` in Figure 11-6. This zone contains records for Microsoft Domain Controller Services. The underscore characters are a requirement of RFC 2052.

Locator records contain:

- **Service name and port** This portion of the SRV record indicates a service with a fixed port. It does not have to be a well-known port. SRV records in Windows Server 2008 include LDAP (port 389), Kerberos (port 88), Kerberos Password protocol (KPASSWD, port 464), and GC services (port 3268).

- **Protocol** TCP or UDP will be indicated as a transport protocol for the service. The same service can use both protocols, in separate SRV records. Kerberos records, for example, are registered for both TCP and UDP. Microsoft clients use only TCP, but UNIX clients can use TCP.
- **Host name** The name corresponds to the A (Host) record for the server hosting the service. When a client queries for a service, the DNS server returns the SRV record and associated A records, so the client does not need to submit a separate query to resolve the IP address of a service.

The service name in the SRV record follows the standard DNS hierarchy, with components separated by dots. For example, the Kerberos service of a domain controller is registered as:

```
kerberos._tcp.siteName._sites.domainName
```

Reading this SRV record name right to left like other DNS records, it translates to:

- *domainName*: the domain or zone, for example, *contoso.com*
- *_sites*: all sites registered with DNS
- *siteName*: the site of the domain controller registering the service
- *_tcp*: any TCP-based services in the site
- *kerberos*: a Kerberos Key Distribution Center (KDC) using TCP as its transport protocol

Domain Controller Location

Imagine a Windows client has just been joined to the domain. It restarts, receives an IP address from a DHCP server, and is ready to authenticate to the domain. How does the client know where to find a domain controller? It does not. Therefore, the client queries the domain for a domain controller by querying the *_tcp* folder which, you'll remember, contains the SRV records for all domain controllers in the domain. DNS returns a list of all matching DCs, and the client attempts to contact all of them on this, its first startup. The first domain controller that responds to the client examines the client's IP address, cross-references that address with subnet objects, and informs the client of the site to which the client belongs. The client stores the site name in its registry, then queries for domain controllers in the site-specific *_tcp* folder. DNS returns a list of all DCs in the site. The client attempts to bind with all, and the DC that responds first authenticates the client.

The client forms an affinity for this DC and will attempt to authenticate with the same DC in the future. If the DC is unavailable, the client queries the site's *_tcp* folder again and attempts to bind with all DCs in the site. But what happens if the client is a mobile computer—a laptop? Imagine that the computer has been authenticating in the BRANCHA site and then the user brings the computer to the BRANCHB site. When the computer starts up, it actually attempts to authenticate with its preferred DC into BRANCHA site. That DC notices the client's IP address is associated with BRANCHB and informs the client of its new site. The client then queries DNS for domain controllers in BRANCHB.

You can see how, by storing subnet and site information in Active Directory and by registering services in DNS, a client is encouraged to use services in its site—the definition of service localization.

MORE INFO Domain controller location

For more information about domain controller location, see http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/distrib/dsbc_nar_jevl.msp?mfr=true.

Site Coverage

What happens if a site has no domain controller? Sites can be used to direct users to local copies of replicated resources such as shared folders replicated within a DFS namespace, so you might have sites without a DC. In this case, a nearby domain controller will register its SRV records in the site in a process called *site coverage*. To be precise, a site without a DC will generally be covered by a domain controller in a site with the lowest cost to the site requiring coverage. You'll learn more about site link costs in the next lesson. You can also manually configure site coverage and SRV record priority if you want to implement strict control over authentication in sites without DCs. The URL just listed contains details about the algorithm that determines which DC automatically covers a site without a DC.

Lesson 2: Configuring the Global Catalog and Application Directory Partitions

As soon as you have more than one domain controller in your domain, you must consider replication of the directory database between domain controllers. In this lesson, you will learn which directory partitions are replicated to each domain controller in a forest and how to manage the replication of the GC and of application partitions.

After this lesson, you will be able to:

- Define the purpose of the global catalog.
- Configure domain controllers as global catalog servers.
- Implement universal group membership caching.
- Understand the role of application directory partitions.

Estimated lesson time: 45 minutes

Reviewing Active Directory Partitions

In Chapter 1, you learned that AD DS includes a data store for identity and management, specifically the directory database, *Ntds.dit*. Within that single file are directory partitions. Each directory partition, also called a naming context, contains objects of a particular scope and purpose. Three major naming contexts have been discussed in this training kit:

- **Domain** The domain naming context (NC) contains all the objects stored in a domain, including users, groups, computers, and Group Policy containers (GPCs).
- **Configuration** The configuration partition contains objects that represent the logical structure of the forest, including domains, as well as the physical topology, including sites, subnets, and services.
- **Schema** The schema defines the object classes and their attributes for the entire directory.

Each domain controller maintains a copy, or *replica*, of several naming contexts. The configuration is replicated to every domain controller in the forest, as is the schema. The domain naming context for a domain is replicated to all domain controllers within a domain but not to domain controllers in other domains, so each domain controller has at least three replicas: the domain NC for its domain, configuration, and schema.

Traditionally, replicas have been complete replicas, containing every object of an attribute, and replicas have been writable on all DCs. Beginning with Windows Server 2008, RODCs change the picture slightly. An RODC maintains a read-only replica of all objects in the configuration, schema, and domain NCs of its domain. However, certain attributes are not replicated to an RODC—specifically, secrets such as user passwords—unless the password policy of the RODC allows such replication. There are also attributes that are domain and forest secrets that are never replicated to an RODC.

Understanding the Global Catalog

Imagine a forest with two domains. Each domain has two domain controllers. All four domain controllers will maintain a replica of the schema and configuration for the forest. The domain controllers in Domain A have replicas of the domain NC for Domain A, and the domain controllers in Domain B have replicas of the domain NC for Domain B.

What happens if a user in Domain B is searching for a user, computer, or group in Domain A? The Domain B domain controllers do not maintain any information about objects in Domain A, so a domain controller in Domain B could not answer a query about objects in the domain NC of Domain A.

That's where the global catalog comes in. The *global catalog* (GC) is a partition that stores information about every object in the forest. When a user in Domain B looks for an object in Domain A, the GC provides the results of the query. To optimize efficiency of the GC, it does not contain every attribute of every object in the forest. Instead, it contains a subset of attributes that are useful for searching across domains. That is why the GC is also called the *partial attribute set* (PAS). In terms of its role supporting search, you can think of the GC as a kind of index for the AD DS data store.

Placing GC Servers

The GC improves efficiency of the directory service tremendously and is required for applications such as Microsoft Exchange Server and Microsoft Office Outlook. Therefore, you want a GC to be available to these and other applications. The GC can be served only by a domain controller and, in an ideal world, every domain controller would be a GC server. In fact, many organizations are now configuring their domain controllers as GC servers.

The potential downside to such a configuration relates to replication. The GC is another partition that must be replicated. In a single domain forest, very little overhead is actually added by configuring all domain controllers as GC servers because all domain controllers already maintain a full set of attributes for all domain and forest objects. In a large, multidomain forest, there will be overhead related to replication of changes to the partial attribute set of objects in other domains. However, many organizations are finding that Active Directory replication is efficient enough to replicate the GC without significant impact to their networks and that the benefits far outweigh such impact. If you choose to configure all DCs as GC servers, you no longer need to worry about the placement of the infrastructure operations master; its role is no longer necessary in a domain where all DCs are GC servers.

It is particularly recommended to configure a GC server on a domain controller in a site where one or more of the following is true:

- A commonly used application performs directory queries, using port 3268, the GC.
- The connection to a GC server is slow or unreliable.
- The site contains a computer running Exchange Server.

Configuring a Global Catalog Server

When you create the first domain in the forest, the first domain controller is configured as a GC. You must decide for each additional DC whether it should be a GC server. The Active Directory Domain Services Installation Wizard and the *Dcpromo.exe* command each enable you to configure a GC server when promoting a domain controller. You can also add or remove the GC from a domain controller by using Active Directory Sites And Services. Expand the site, the Servers container within the site, and the domain controller's server object. Right-click the NTDS Settings node and choose Properties. On the General tab, shown in Figure 11-7, select the Global Catalog check box. To remove the GC from a domain controller, perform the same steps, clearing the Global Catalog check box.

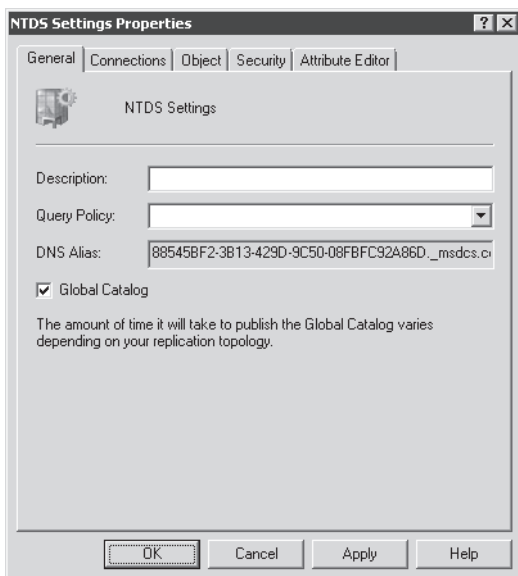


Figure 11-7 The NTDS Settings Properties dialog box, showing the Global Catalog check box

Universal Group Membership Caching

In Chapter 4, “Groups,” you learned that Active Directory supports groups of universal scope. Universal groups are designed to include users and groups from multiple domains in a forest. The membership of universal groups is replicated in the GC. When a user logs on, the user's universal group membership is obtained from a GC server. If a GC is not available, universal group membership is not available. It's possible that a universal group is used to deny the user access to resources, so Windows prevents a security incident by denying domain authentication to the user. If the user has logged on to his or her computer before, he or she can log on using cached credentials, but as soon as the user attempts to access network resources, access will be denied. To summarize: if a GC server is not available, users will effectively be unable to log on and access network resources.

If every domain controller is a GC server, this problem will not arise. However, if replication is a concern, and if you have, therefore, chosen not to configure a domain controller as a GC server, you can facilitate successful logon by enabling universal group membership caching (UGMC). When you configure universal group membership caching on a domain controller in a branch office, for example, that domain controller will obtain universal group membership information from a GC for a user when the user first logs on in the site, and the domain controller will cache that information indefinitely, updating universal group membership information every eight hours. That way, if the user later logs on and a GC server is not accessible, the domain controller can use its cached membership information to permit logon by the user.

It is recommended, therefore, that in sites with unreliable connectivity to a GC server, you should configure UGMC on the site's domain controllers. To configure UGMC, open the Active Directory Sites And Services snap-in and select the site in the console tree. In the details pane, right-click NTDS Site Settings and choose Properties. The NTDS Site Settings Properties dialog box, shown in Figure 11-8, exposes the Enable Universal Group Membership Caching option. You can select the check box and specify the GC from which to refresh the membership cache.

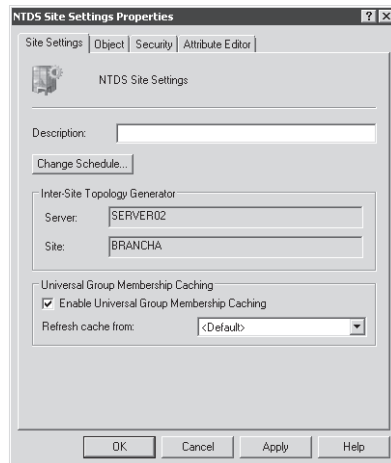


Figure 11-8 The NTDS Site Settings Properties dialog box with the option to enable Universal Group Membership Caching

Understanding Application Directory Partitions

Whereas the domain, configuration, and schema partitions of the directory are replicated to all DCs in a domain, and the configuration and schema are further replicated to all DCs in the forest, Active Directory also supports *application directory partitions*. An application directory partition is a portion of the data store that contains objects required by an application or service

that is outside of the core AD DS service. Unlike other partitions, application partitions can be targeted to replicate to specific domain controllers; they are not, by default, replicated to all DCs.

Application directory partitions are designed to support directory-enabled applications and services. They can contain any type of object except security principals such as users, computers, or security groups. Because these partitions are replicated only as needed, application directory partitions provide the benefits of fault tolerance, availability, and performance while optimizing replication traffic.

The easiest way to understand application directory partitions is to examine the application directory partitions maintained by Microsoft DNS Server. When you create an Active Directory–integrated zone, DNS records are replicated between DNS servers by using an application directory partition. The partition and its DNS record objects are not replicated to every domain controller, only to those acting as DNS servers.

You can expose the application directory partitions in your forest by opening ADSI Edit. Right-click the root of the snap-in, ADSI Edit, and choose Connect To. In the Select A Well Known Naming Context drop-down list, choose Configuration and click OK. Expand Configuration and the folder representing the configuration partition, and then select the Partitions folder, CN=Partitions, in the console tree. In the details pane, you will see the partitions in your AD DS data store, as shown in Figure 11-9.

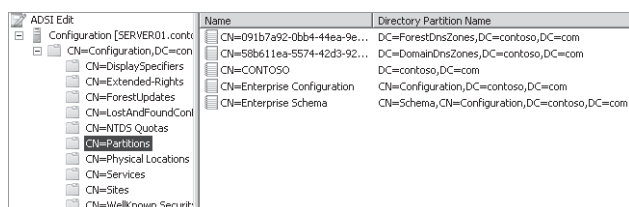


Figure 11-9 Partitions in the *contoso.com* forest

Note the two application partitions in Figure 11-9, *ForestDnsZones* and *DomainDnsZones*. Most application partitions are created by applications that require them. DNS is one example, and Telephony Application Programming Interface (TAPI) is another. Members of the Enterprise Admins group can also create application directory partitions manually by using *Ntdsutil.exe*.

An application partition can appear anywhere in the forest namespace that a domain partition can appear. The DNS partitions distinguished names—*DC=DomainDnsZones,DC=contoso,DC=com*, for example—place the partitions as children of the *DC=contoso,DC=com* domain partition. An application partition can also be a child of another application partition or a new tree in the forest.

MORE INFO About application directory partitions

For more information about application directory partitions, visit <http://technet2.microsoft.com/WindowsServer/en/library/ed363e83-c043-4a50-9233-763e6f4af1f21033.mspx>.

Generally speaking, you will use tools specific to the application to manage the application directory partition, its data, and its replication. For example, simply adding an Active Directory-integrated zone to a DNS server will automatically configure the domain controller to receive a replica of the DomainDns partition. With tools such as *Ntdsutil.exe* and *Ldp.exe*, you can manage application directory partitions directly.

MORE INFO Managing application directory partitions

To learn how to manage application directory partitions, see <http://technet2.microsoft.com/WindowsServer/en/library/920d6995-9ee9-46a7-9d1b-320e65c02d1a1033.mspx>.

It is important that you consider application partitions prior to demoting a domain controller. If a domain controller is hosting an application directory partition, you must evaluate the purpose of the partition, whether it is required by any applications, and whether the domain controller holds the last remaining replica of the partition, in which case, demoting the domain controller will result in permanently losing all information in the partition. Although the Active Directory Domain Services Installation Wizard will prompt you to remove application directory partitions, it is recommended that you manually remove application directory partitions prior to demoting a domain controller.

MORE INFO Application directory partitions and domain controller demotion

For more information about application directory partitions and domain controller demotion, see <http://technet2.microsoft.com/WindowsServer/en/library/1572d8a2-622c-4879-bb0b-76e26c4001291033.mspx>.

Lesson 3: Configuring Replication

In Lesson 1, you learned how to create site and subnet objects that enable Active Directory and its clients to localize authentication and directory access; you decided *where* domain controllers should be placed. In Lesson 2, you configured GC servers and application directory partitions; you managed *what* will replicate between domain controllers. In this lesson, you will learn *how* and *when* replication occurs. You'll discover why the default configuration of Active Directory supports effective replication and why you might modify that configuration so that replication is equally effective but more efficient, based on your network topology.

After this lesson, you will be able to:

- Create connection objects to configure replication between two domain controllers.
- Implement site links and site link costs to manage replication between sites.
- Designate preferred bridgehead servers.
- Understand notification and polling.
- Report and analyze replication with *Repadmin.exe*.
- Perform Active Directory replication health checks with *Dcdiag.exe*.

Estimated lesson time: 90 minutes

Understanding Active Directory Replication

In previous lessons, you learned how to place domain controllers in network locations and how to represent those locations with site and subnet objects. You also learned about the replication of directory partitions (schema, configuration, and domain), the partial attribute set (GC), and application partitions. The most important thing to remember as you learn about Active Directory replication is that it is designed so that, in the end, each replica on a domain controller is consistent with the replicas of that partition hosted on other domain controllers. It is not likely that all domain controllers will have exactly the same information in their replicas at any one moment in time because changes are constantly being made to the directory. However, Active Directory replication ensures that all changes to a partition are transferred to all replicas of the partition. Active Directory replication balances accuracy (or *integrity*) and consistency (called *convergence*) with performance (keeping replication traffic to a reasonable level). This balancing act is described as *loose coupling*.

Key features of Active Directory replication are:

- Partitioning of the data store. Domain controllers in a domain host only the domain naming context for their domain, which helps keep replication to a minimum, particularly in multidomain forests. Other data, including application directory partitions and the partial attribute set (GC), are not replicated to every domain controller in the forest, by default.

- Automatic generation of an efficient and robust replication topology. By default, Active Directory will configure an effective, two-way replication topology so that the loss of any one domain controller does not impede replication. This topology is automatically updated as domain controllers are added, removed, or moved between sites.
- Attribute-level replication. When an attribute of an object is modified, only that attribute, and minimal metadata that describes that attribute, is replicated. The entire object is not replicated except when the object is created.
- Distinct control of intrasite replication (within a single site) and intersite replication (between sites).
- Collision detection and management. It is possible, although rare, that an attribute will have been modified on two different domain controllers during a single replication window. In such an event, the two changes will have to be reconciled. Active Directory has resolution algorithms that satisfy almost every such situation.

It is easier to understand Active Directory replication by examining each of its components. The following sections examine the components of Active Directory replication.

Connection Objects

A domain controller replicates changes from another domain controller because of AD DS connection objects, also called simply *connection objects*. Connection objects appear in the administrative tools in the Active Directory Sites and Services snap-in as objects contained in the NTDS Settings container of a domain controller's server object. Figure 11-10 shows an example: a connection object in SERVER02 configures replication from SERVER01 to SERVER02. A connection object represents a replication path from one domain controller to another.

Connection objects are one-way, representing inbound-only replication. Replication in Active Directory is always a pull technology. In the domain illustrated in Figure 11-10, SERVER02 pulls changes from SERVER01. SERVER02 is considered, in this example, a downstream replication partner of SERVER01. SERVER01 is the upstream partner. Changes from SERVER01 flow to SERVER02.

NOTE Force replication

You can force replication between two domain controllers by right-clicking the connection object and choosing Replicate Now. Remember replication is inbound only, so to replicate both domain controllers, you will need to replicate the inbound connection object of each domain controller.

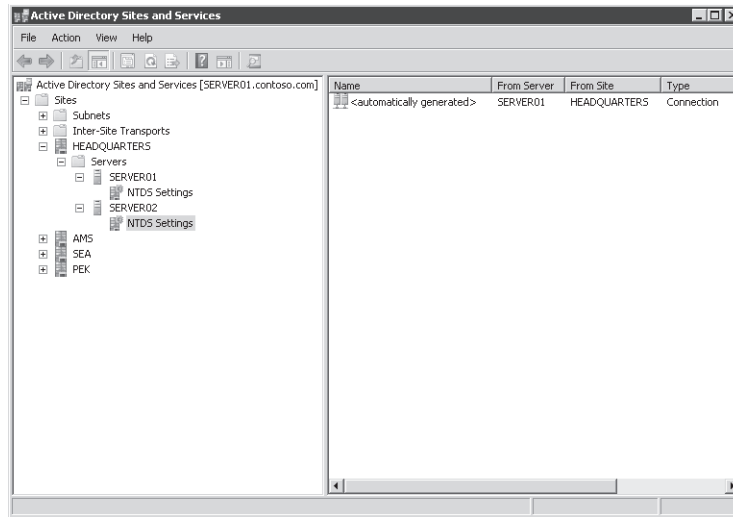


Figure 11-10 A connection object in the Active Directory Sites and Services snap-in

The Knowledge Consistency Checker

The replication paths built between domain controllers by connection objects create the replication topology for the forest. Luckily, you do not have to create the replication topology manually. By default, Active Directory creates a topology that ensures effective replication. The topology is two-way so that if any one domain controller fails, replication will continue uninterrupted. The topology also ensures that there are no more than three hops between any two domain controllers.

You'll notice in Figure 11-10 that the connection object indicates it was automatically generated. On each domain controller, a component of Active Directory called the knowledge consistency checker (KCC) helps generate and optimize the replication automatically between domain controllers within a site. The KCC evaluates the domain controllers in a site and creates connection objects to build the two-way, three-hop topology described earlier. If a domain controller is added to or removed from the site, or if a domain controller is not responsive, the KCC rearranges the topology dynamically, adding and deleting connection objects to rebuild an effective replication topology.

You can manually create connection objects to specify replication paths that should persist. Manually created connection objects are not deleted by the KCC. To create a connection object, locate the server object for the downstream replication partner—the DC that will receive changes from a source DC. Right-click the NTDS Settings container in the server object and choose New Active Directory Domain Services Connection. In the Find Active Directory Domain Controllers dialog box, select the upstream replication partner and click OK. Give the new connection object a name and click OK. Then open the properties of the connection

object; use the Description field to indicate the purpose of any manually created connection object.

Within a site, there are very few scenarios that would require creating a connection object. One such scenario is standby operations masters. Operations masters are discussed in Chapter 10. It is recommended that you select domain controllers as standby operations masters to be used in the event that the operations master role must be transferred or seized. A standby operations master should be a direct replication partner with the current operations master. Thus, if a domain controller named DC01 is the RID master, and DC02 is the system that will take the RID master role if DC01 is taken offline, then a connection object should be created in DC02 so that it replicates directly from DC01.

Intrasite Replication

After connection objects between the domain controllers in a site have been established—automatically by the KCC or manually—replication can take place. Intrasite replication involves the replication of changes within a single site.

Notification

Consider the site shown in Figure 11-10. When SERVER01 makes a change to a partition, it queues the change for replication to its partners. SERVER01 waits 15 seconds, by default, to notify its first replication partner, SERVER02, of the change. *Notification* is the process by which an upstream partner informs its downstream partners that a change is available. SERVER01 waits three seconds, by default, between notifications to additional partners. These delays, called the *initial notification delay* and the *subsequent notification delay*, are designed to stagger network traffic caused by intrasite replication.

Upon receiving the notification, the downstream partner, SERVER02, requests the changes from SERVER01, and the directory replication agent (DRA) performs the transfer of the attribute from SERVER01 to SERVER02. In this example, SERVER01 made the initial change to Active Directory. It is the originating domain controller, and the change it made originates the change. When SERVER02 receives the change from SERVER01, it makes the change to its directory. The change is not called a replicated change, but it is a change nonetheless. SERVER02 queues the change for replication to its own downstream partners.

SERVER03 is a downstream replication partner of SERVER02. After 15 seconds, SERVER02 notifies SERVER03 that it has a change. SERVER03 makes the replicated change to its directory and then notifies its downstream partners. The change has made two hops, from SERVER01 to SERVER02 and from SERVER02 to SERVER03. The replication topology will ensure that there are no more than three hops before all domain controllers in the site have received the change. At approximately 15 seconds per hop, that means the change will have fully replicated in the site within one minute.

Polling

It is possible that SERVER01 might not make any changes to its replicas for quite a long time, particularly during off hours. In this case, SERVER02, its downstream replication partner, will not receive notifications from SERVER01. It is also possible that SERVER01 might be offline, which would also prevent it from sending notifications to SERVER02, so it's important for SERVER02 to know that its upstream partner is online and simply does not have any changes.

This is achieved through a process called *polling*. Polling involves the downstream replication partner contacting the upstream replication partner with a query as to whether any changes are queued for replication. By default, the polling interval for intrasite replication is once per hour. It is possible, although not recommended, to configure the polling frequency from the properties of a connection object by clicking Change Schedule.

If an upstream partner fails to respond to repeated polling queries, the downstream partner launches the KCC to check the replication topology. If the upstream server is indeed offline, the site's replication topology is rebuilt to accommodate the change.

Site Links

The KCC assumes that within a site, all domain controllers can reach each other. It builds an intrasite replication topology that is agnostic to the underlying network connectivity. Between sites, however, you can represent the network paths over which replication should occur by creating *site link* objects. A site link contains two or more sites. The intersite topology generator (ISTG), a component of the KCC, builds connection objects between servers in each of the sites to enable intersite replication—replication between sites.

Site links are greatly misunderstood, and the important thing to remember about a site link is that it represents an available path for replication. A single site link does not control the network routes that are used. When you create a site link and add sites to it, you are telling Active Directory that it can replicate between any of those sites. The ISTG will create connection objects, and those objects will determine the actual path of replication. Although the replication topology built by the ISTG will effectively replicate Active Directory, it might not be efficient, given your network topology.

An example will illustrate this concept. When you create a forest, one site link object is created: DEFAULTIPSITELINK. By default, each new site that you add is associated with the DEFAULTIPSITELINK. Consider an organization with a data center at the headquarters and three branch offices. The three branch offices are each connected to the data center with a dedicated link. You create sites for each branch office, Seattle (SEA), Amsterdam (AMS), and Beijing (PEK). The network and site topology is shown in Figure 11-11.

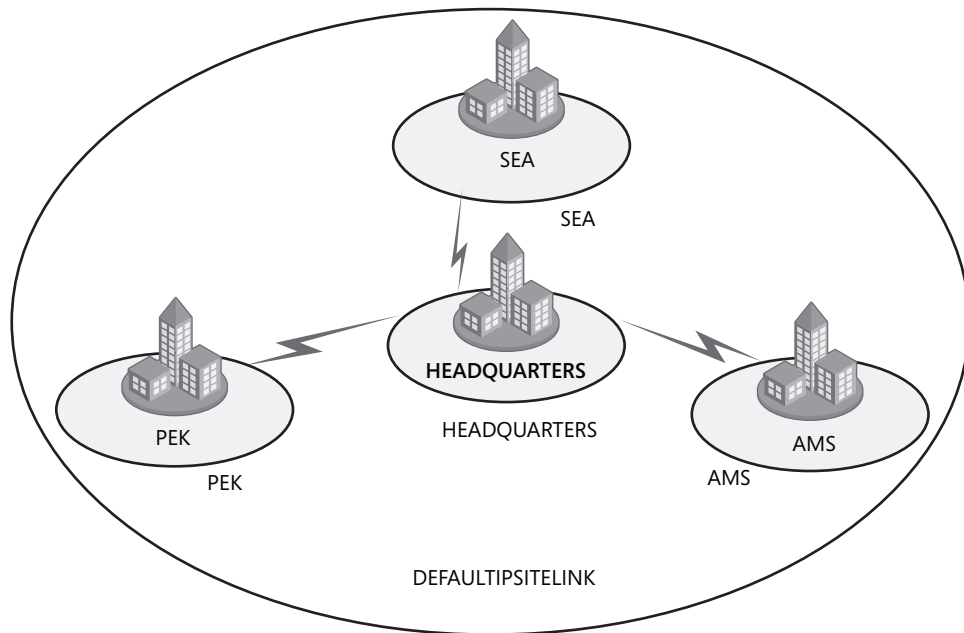


Figure 11-11 Network topology and a single site link

Because all four sites are on the same site link, you are instructing Active Directory that all four sites can replicate with each other. That means it is possible that Seattle will replicate changes from Amsterdam, Amsterdam will replicate changes from Beijing, and Beijing will replicate changes from Headquarters, which in turn replicates changes from Seattle. In several of these replication paths, the replication traffic on the network flows from one branch through the headquarters on its way to another branch. With a single site link, you have not created a hub-and-spoke replication topology even though your network topology is hub-and-spoke.

Therefore, it is recommended that you manually create site links that reflect your physical network topology. Continuing the preceding example, you would create three site links:

- HQ-AMS, including the Headquarters and Amsterdam sites
- HQ-SEA, including the Headquarters and Seattle sites
- HQ-PEK, including the Headquarters and Beijing sites

You would then delete the DEFAULTIPSITELINK. The resulting topology is shown in Figure 11-12.

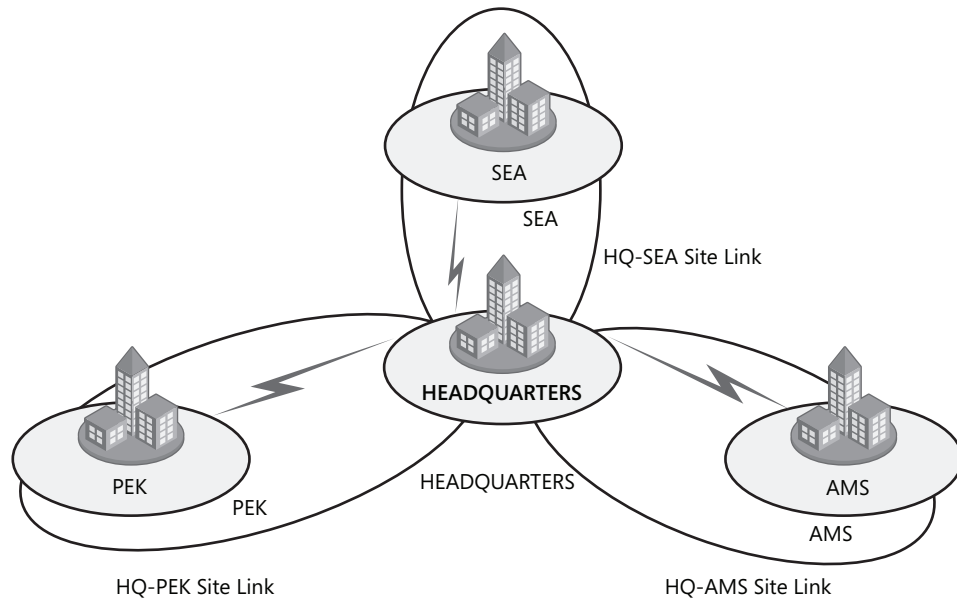


Figure 11-12 Network topology and a three-site link

After you have created site links, the ISTG will use the topology to build an intersite replication topology connecting each site. Connection objects will be built to configure the intersite replication paths. These connection objects are created automatically, and although you can create connection objects manually, few scenarios require manually creating intersite connection objects.

Replication Transport Protocols

You'll notice, in the Active Directory Sites and Services snap-in, that site links are contained within a container named IP that itself is inside the Inter-Site Transports container. Changes are replicated between domain controllers, using one of two protocols:

- **Directory Service Remote Procedure Call (DS-RPC)** DS-RPC appears in the Active Directory Sites and Services snap-in as IP. IP is used for all intrasite replication and is the default, and preferred, protocol for intersite replication.
- **Inter-Site Messaging—Simple Mail Transport Protocol (ISM-SMTP)** Also known simply as SMTP, this protocol is used only when network connections between sites are unreliable or are not always available.

In general, you can assume you will use IP for all intersite replication. Very few organizations use SMTP for replication because of the administrative overhead required to configure and manage a certificate authority (CA) and because SMTP replication is not supported for the

domain naming context, meaning that if a site uses SMTP to replicate to the rest of the enterprise, that site must be its own domain.

Exam Tip Although, in the production environment, you are highly unlikely to use SMTP for replication, it is possible you will encounter SMTP replication on the exam. The most important thing to remember is that if two sites can replicate only with SMTP—if IP is not an option—then those two sites must be separate domains in the forest. SMTP cannot be used to replicate the domain naming context.

Bridgehead Servers

The ISTG creates a replication topology between sites on a site link. To make replication more efficient, one domain controller is selected to be the *bridgehead server*. The bridgehead server is responsible for all replication into and out of the site for a partition. For example, if a data center site contains five DCs, one of the DCs will be the bridgehead server for the domain naming context. All changes made to the domain partition within the data center will replicate to all DCs in the site. When the changes reach the bridgehead server, those changes will be replicated to bridgehead servers in branch offices, which in turn replicate the changes to DCs in their sites. Similarly, any changes to the domain naming context in branch offices will be replicated from the branches' bridgehead servers to the bridgehead server in the data center, which in turn replicates the changes to other DCs in the data center. Figure 11-13 illustrates intrasite replication within two sites and the intersite replication using connection objects between the bridgehead servers in the sites.

To summarize, the bridgehead server is the server responsible for replicating changes to a partition from other bridgehead servers in other sites. It is also polled by bridgehead servers in other sites to determine when it has changes that they should replicate.

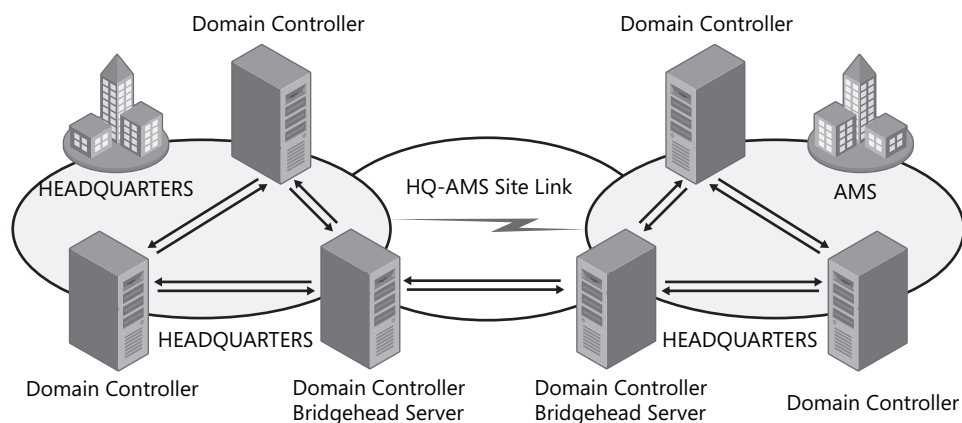


Figure 11-13 Sites, intrasite replication, bridgehead servers, and intersite replication

Bridgehead servers are selected automatically, and the ISTG creates the intersite replication topology to ensure that changes are replicated effectively between bridgeheads sharing a site link. Bridgeheads are selected per partition, so it is possible that one DC in a site might be the bridgehead server for the schema and another might be for the configuration. However, you will usually find that one domain controller is the bridgehead server for all partitions in a site unless there are domain controllers from other domains or application directory partitions, in which case bridgeheads will be chosen for those partitions.

Preferred Bridgehead Servers

You can also designate one or more *preferred bridgehead servers*. To designate a domain controller as a preferred bridgehead server, open the properties of the server object in the Active Directory Sites And Services snap-in, select the transport protocol, which will almost always be IP, and click Add.

You can configure more than one preferred bridgehead server for a site, but only one will be selected and used as the bridgehead. If that bridgehead fails, one of the other preferred bridgehead servers will be used.

It's important to understand that if you have specified one or more bridgehead servers and none of the bridgeheads is available, no other server is automatically selected, and replication does not occur for the site even if there are servers that could act as bridgehead servers. In an ideal world, you should not configure preferred bridgehead servers. However, performance considerations might suggest that you assign the bridgehead server role to domain controllers with greater system resources. Firewall considerations might also require that you assign a single server to act as a bridgehead instead of allowing Active Directory to select and possibly reassign bridgehead servers over time.

Configuring Intersite Replication

After you have created site links and the ISTG has generated connection objects to replicate partitions between bridgehead servers that share a site link, your work might be complete. In many environments, particularly those with straightforward network topologies, site links might be sufficient to manage intersite replication. In more complex networks, however, you can configure additional components and properties of replication.

Site Link Transitivity

By default, site links are transitive. That means, continuing the example from earlier, that if Amsterdam and Headquarters sites are linked, and Headquarters and Seattle sites are linked, then Amsterdam and Seattle are transitively linked. This means, theoretically, that the ISTG could create a connection object directly between a bridgehead in Seattle and a bridgehead in Amsterdam, again working around the hub-and-spoke network topology.

You can disable site link transitivity by opening the properties of the IP transport in the Inter-Site Transports container and deselecting Bridge All Site Links. Before you do this in a production environment, be sure to spend time reading the technical resources about replication in the Windows Server technical libraries on Microsoft TechNet at <http://technet.microsoft.com>.

Exam Tip For the certification exam, you need to know that site links are transitive by default, that transitivity can be disabled, and that when transitivity is disabled, you might want to build site link bridges.

Site Link Bridges

A site link bridge connects two or more site links in a way that creates a transitive link. Site link bridges are necessary only when you have cleared the Bridge All Site Links option for the transport protocol. Remember that site link transitivity is enabled by default, in which case site link bridges have no effect.

Figure 11-14 illustrates the use of a site link bridge in a forest in which site link transitivity has been disabled. By creating a site link bridge, AMS-HQ-SEA, that includes the HQ-AMS and HQ-SEA site links, those two site links become transitive, so a replication connection can be made between a domain controller in Amsterdam and a domain controller in Seattle.

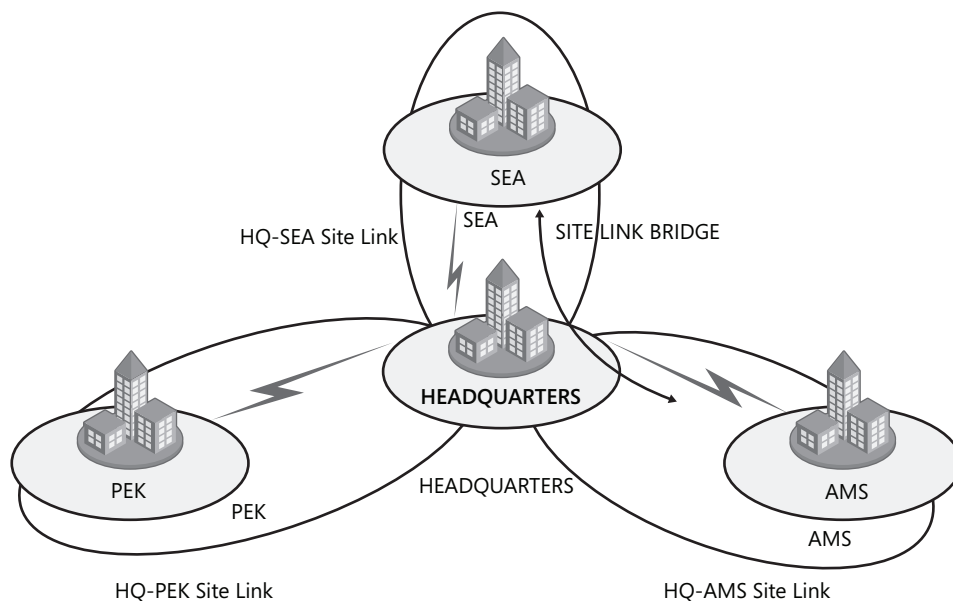


Figure 11-14 A site link bridge that includes the HQ-AMS and HQ-SEA site links

Site Link Costs

Site link costs are used to manage the flow of replication traffic when there is more than one route for replication traffic. You can configure site link cost to indicate that a link is faster, more reliable, or is preferred. Higher costs are used for slow links, and lower costs are used for fast links. Active Directory replicates using the connection with the lowest cost.

By default, all site links are configured with a cost of 100. To change the site link cost, open the properties of a site link and change the value in the Cost spin-box, shown in Figure 11-15.

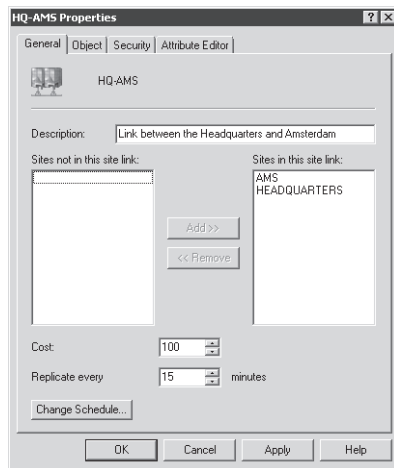


Figure 11-15 The properties of a site link

Returning to the example used earlier in the lesson, imagine if a site link was created between the Amsterdam and Beijing sites, as shown in Figure 11-16. Such a site link could be configured to allow replication between domain controllers in those two sites in the event that the links to the headquarters became unavailable. You might want to configure such a topology as part of a disaster recovery plan, for example.

With the default site link cost of 100 assigned to the AMS-PEK site link, Active Directory will replicate changes directly between Amsterdam and Beijing. If you configure the site link cost to 300, changes will replicate between Amsterdam and the Headquarters, then between the Headquarters and Beijing at a cost of 200 rather than directly over the AMS-PEK site link at a cost of 300.

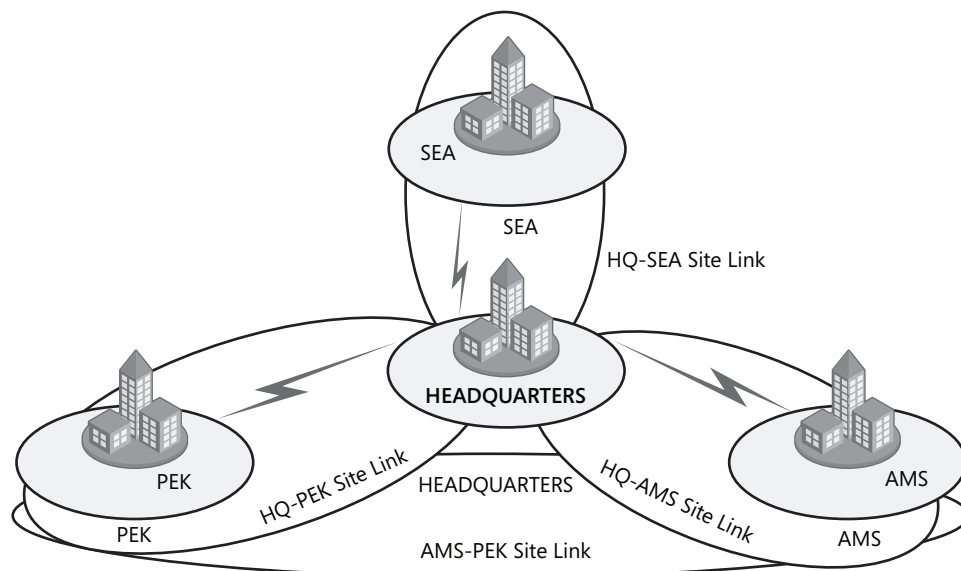


Figure 11-16 Site links and costs

Replication Frequency

Intersite replication is based only on polling; there is no notification. Every three hours, by default, a bridgehead server will poll its upstream replication partners to determine whether changes are available. This replication interval is too long for organizations that want changes to the directory to replicate more quickly. You can change the polling interval for each site link. Open the site link's properties and change the value in the Replicate Every spin-box, shown in Figure 11-15.

The minimum polling interval is 15 minutes. That means, using Active Directory's default replication configuration, a change made to the directory in one site will take several minutes before it is replicated to domain controllers in another site.

Replication Schedules

By default, replication occurs 24 hours a day. However, you can restrict intersite replication to specific times by changing the schedule attributes of a site link. Open the properties of a site link and click the Change Schedule button. Using the Schedule For dialog box shown in Figure 11-17, you can select the times during which the link is available for replication. The link shown in the figure does not replicate from 8:00 A.M. to 6:00 P.M. Monday through Friday.

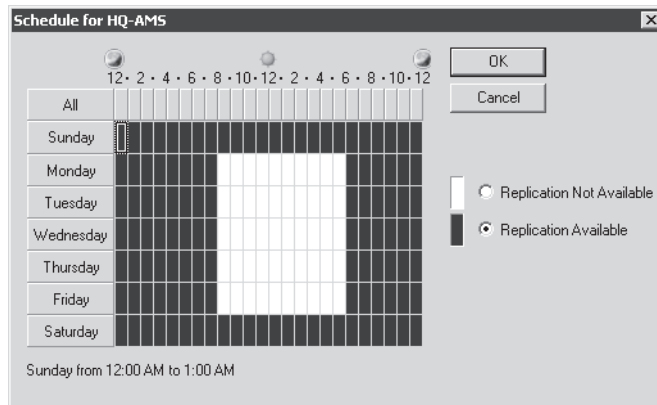


Figure 11-17 Site link schedule

You must be careful when scheduling site link availability. It is possible to schedule windows of availability that do not overlap, at which point replication will not happen. It's generally not recommended to configure link availability. If you do not require link scheduling, select the Ignore Schedules option in the properties of the IP transport protocol. This option causes any schedules for site link availability to be ignored, ensuring replication 24 hours a day over all site links.

Monitoring Replication

After you have implemented your replication configuration, you must be able to monitor replication for ongoing support, optimization, and troubleshooting. Two tools are particularly useful for reporting and analyzing replication: the Replication Diagnostics tool (*Repadmin.exe*) and Directory Server Diagnosis (*Dcdiag.exe*). This lesson introduces you to these powerful tools.

Repadmin.exe

The Replication Diagnostics tool, *Repadmin.exe*, is a command-line tool that enables you to report the status of replication on each domain controller. The information produced by *Repadmin.exe* can help you spot a potential problem before it gets out of control and troubleshoot problems with replication in the forest. You can view levels of detail down to the replication metadata for specific objects and attributes, enabling you to identify where and when a problematic change was made to Active Directory. You can even use *Repadmin.exe* to create the replication topology and force replication between domain controllers.

Like other command-line tools, you can type **repadmin /?** to see the usage information for the tool. Its basic syntax is as follows:

```
repadmin command arguments...
```

Repadmin.exe supports a number of commands that perform specific tasks. You can learn about each command by typing **repadmin /?:command**. Most commands require arguments. Many commands take a *DSA_LIST* parameter, which is simply a network label (DNS or Net-BIOS name or IP address) of a domain controller. Some of the replication monitoring tasks you can perform with *Repadmin* are:

- **Displaying the replication partners for a domain controller** To display the replication connections of a domain controller, type **repadmin /showrepl DSA_LIST**. By default, *Repadmin.exe* shows only intersite connections. Add the */repsto* argument to see intersite connections as well.
- **Displaying connection objects for a domain controller** Type **repadmin /showconn DSA_LIST** to show the connection objects for a domain controller.
- **Displaying metadata about an object, its attributes, and replication** You can learn a lot about replication by examining an object on two different domain controllers to find out which attributes have or have not replicated. Type **repadmin /showobjmeta DSA_LIST Object**, where *DSA_LIST* indicates the domain controller(s) to query. (You can use an asterisk [*] to indicate all domain controllers.) *Object* is a unique identifier for the object, its DN, or its GUID, for example.

You can also make changes to your replication infrastructure by using *Repadmin*. Some of the management tasks you can perform are:

- **Launching the KCC** Type **repadmin /kcc** to force the KCC to recalculate the inbound replication topology for the server.
- **Forcing replication between two partners** You can use *Repadmin* to force replication of a partition between a source and a target domain controller. Type **repadmin /replicate Destination_DSA_LIST Source_DSA_Name Naming_Context**.
- **Synchronizing a domain controller with all replication partners** Type **repadmin /syncall DSA /A /e** to synchronize a domain controller with all its partners, including those in other sites.

Dcdiag.exe

The Directory Service Diagnosis tool, *Dcdiag.exe*, performs a number of tests and reports on the overall health of replication and security for AD DS. Run by itself, *Dcdiag.exe* performs summary tests and reports the results. On the other extreme, *Dcdiag.exe /c* performs almost every test. The output of tests can be redirected to files of various types, including XML. Type **dcdiag /?** for full usage information.

You can also specify one or more tests to perform using the */test:Test Name* parameter. Tests that are directly related to replication include:

Chapter 12

Domains and Forests

In Chapter 1, “Installation,” you learned that Active Directory Domain Services (AD DS) provides the foundation for an identity and access management solution, and you explored the creation of a simple AD DS infrastructure consisting of a single forest and a single domain. In subsequent chapters, you mastered the details of managing an AD DS environment. Now, you are ready to return to the highest level of an AD DS infrastructure and consider the model and functionality of your domains and forests. In this chapter, you will learn how to raise the domain and forest functionality levels within your environment, how to design the optimal AD DS infrastructure for your enterprise, how to migrate objects between domains and forests, and how to enable authentication and resource access across multiple domains and forests.

Exam objectives in this chapter:

- Configuring the Active Directory Infrastructure
 - Configure a forest or a domain.
 - Configure trusts.

Before You Begin

To complete the practices in this chapter, you must have created two domain controllers, named SERVER01 and SERVER02, in a domain named *contoso.com*. See Chapter 1 and Chapter 10, “Domain Controllers,” for detailed steps for this task.

Real World

Dan Holme

In some organizations, there is a perception that domain controllers should be the last systems to be upgraded. My experience, however, has been that domain controllers (DCs) should be among the first systems that you should upgrade (after testing the upgrade in a lab, of course). Domain controllers are the cornerstone of identity and access management in your enterprise AD DS forest. Because of that, you should ensure that, wherever possible, DCs are dedicated—serving only the AD DS role and related core services, such as DNS. If your DCs are dedicated, the risk associated with upgrading them diminishes significantly—there are far fewer moving parts that could cause problems during an upgrade. Additionally, the sooner you upgrade your DCs, the sooner you can raise the domain and forest functional levels.

Functional levels enable the newer capabilities added by Microsoft Windows Server 2003 and Windows Server 2008. In return for added functionality, you are restricted as to the versions of Microsoft Windows that are supported for the domain controllers in the domain. (Member servers and workstations can run any version of Windows.) Some of the functionality, such as linked-value replication, last logon information, read-only domain controllers, fine-grained password policies, and Distributed File System Replication (DFS-R) of System Volume (SYSVOL), have a profound impact on the day-to-day security, management, and flexibility of AD DS. I encourage you to move with a reasonable but quick pace toward upgrading your domain controllers to Windows Server 2008 so you can raise the domain and forest functional levels to take advantage of these capabilities. They make a big difference.

Lesson 1: Understanding Domain and Forest Functional Levels

As you introduce Windows Server 2008 domain controllers into your domains and forest, you can begin to take advantage of new capabilities in Active Directory directory service. Domain and forest functional levels are operating modes of domains and forests, respectively. Functional levels determine the versions of Windows that you can use as domain controllers and the availability of Active Directory features.

After this lesson, you will be able to:

- Understand domain and forest functional levels.
- Raise domain and forest functional levels.
- Identify capabilities added by each functional level.

Estimated lesson time: 45 minutes

Understanding Functional Levels

Functional levels are like switches that enable new functionality offered by each version of Windows. Windows Server 2003 added several features to Active Directory, and Windows Server 2008 continues the evolution of AD DS. These features are not backward compatible, so if you have DCs running Windows 2000 Server, you cannot enable the functionality offered by later versions of Windows, so the newer functionality is disabled. Similarly, until all DCs are running Windows Server 2008, you cannot implement its enhancements to AD DS. Raising the functional level entails two major tasks:

- All domain controllers must be running the correct version of Windows Server.
- You must manually raise the functional level. It does not happen automatically.

NOTE Functional levels, operating system versions, and domain controllers

Remember that only domain controllers determine your ability to set a functional level. You can have member servers and workstations running any version of Windows within a domain or forest at any functional level.

Domain Functional Levels

The domain functional level affects the Active Directory features available within the domain and determines the versions of Windows that are supported for domain controllers within the domain. In previous versions of Windows, domain functional levels and modes, as they were called in Windows 2000 Server, supported domain controllers running Microsoft Windows NT 4.0. That support has ended with Windows Server 2008. All domain controllers must be

running Windows 2000 Server or later before you can add the first Windows Server 2008 domain controller to the domain. Windows Server 2008 Active Directory supports three domain functional levels:

- Windows 2000 Native
- Windows Server 2003
- Windows Server 2008

Windows 2000 Native

The Windows 2000 Native domain functional level is the lowest functional level that supports a Windows Server 2008 domain controller. The following operating systems are supported for domain controllers:

- Windows 2000 Server
- Windows Server 2003
- Windows Server 2008

If you have domain controllers running Windows 2000 Server or Windows Server 2003, or if you expect that you might add one or more domain controllers running those previous versions of Windows, you should leave the domain at Windows 2000 Native functional level.

Windows Server 2003

After you have removed or upgraded all domain controllers running Windows 2000 Server, the domain functional level can be raised to Windows Server 2003. At this functional level, the domain can no longer support domain controllers running Windows 2000 Server, so all domain controllers must be running one of the following two operating systems:

- Windows Server 2003
- Windows Server 2008

Windows Server 2003 domain functional level adds a number of new features offered at the Windows 2000 Native domain functional level. These features include the following:

- **Domain controller rename** The domain management tool, *Netdom.exe*, can be used to prepare for domain controller rename.
- **The *lastLogonTimestamp* attribute** When a user or computer logs on to the domain, the *lastLogonTimestamp* attribute is updated with the logon time. This attribute is replicated within the domain.
- **The *userPassword* attribute** Security principals in Active Directory include users, computers, and groups. A fourth object class, *inetOrgPerson*, is similar to a user and is used to integrate with several non-Microsoft directory services. At the Windows Server 2003 domain functional level, you can set the *userPassword* attribute as the effective password on both *inetOrgPerson* and *user* objects.

- **Default user and computer container redirection** In Chapter 5, “Computers,” you learned that you can use the *Redirusr.exe* and *Redircmp.exe* commands to redirect the default user and computer containers. Doing so causes new accounts to be created in specific organizational units rather than in the Users and Computers containers.
- **Authorization Manager policies** Authorization Manager, a tool that can be used to provide authorization by applications, can store its authorization policies in AD DS.
- **Constrained delegation** Applications can take advantage of the secure delegation of user credentials by means of the Kerberos authentication protocol. Delegation can be configured to be allowed only to specific destination services.
- **Selective authentication** In Lesson 2, “Managing Multiple Domains and Trust Relationships,” you will learn to create trust relationships between your domain and another domain or forest. Selective authentication enables you to specify the users and groups from the trusted domain or forest who are allowed to authenticate to servers in your forest.

Windows Server 2008

When all domain controllers are running Windows Server 2008, and you are confident that you will not need to add domain controllers running previous versions of Windows, you can raise the domain functional level to Windows Server 2008. Windows Server 2008 domain functional level supports domain controllers running only one operating system—Windows Server 2008.

Windows Server 2008 domain functional level adds four domain-wide features to AD DS:

- **DFS-R of SYSVOL** In Chapter 10, you learned to configure SYSVOL so that it is replicated with Distributed File System Replication (DFS-R) instead of with File Replication Service (FRS). DFS-R provides a more robust and detailed replication of SYSVOL contents.
- **Advanced Encryption Services** You can increase the security of authentication with Advanced Encryption Services (AES 128 and AES 256) support for the Kerberos protocol. AES replaces the RC4-HMAC (Hash Message Authentication Code) encryption algorithm.
- **Last interactive logon information** When a user logs on to the domain, several attributes of the user object are updated with the time, the workstation to which the user logged on, and the number of failed logon attempts since the last logon.
- **Fine-grained password policies** In Chapter 8, “Authentication,” you learned about fine-grained password policies, which enable you to specify unique password policies for users or groups in the domain.

Raising the Domain Functional Level

You can raise the domain functional level after all domain controllers are running a supported version of Windows and when you are confident you will not have to add domain controllers running unsupported versions of Windows. To raise the domain functional level, open the Active Directory Domains And Trusts snap-in, right-click the domain, and choose Raise Domain Functional Level. The dialog box shown in Figure 12-1 enables you to select a higher domain functional level.

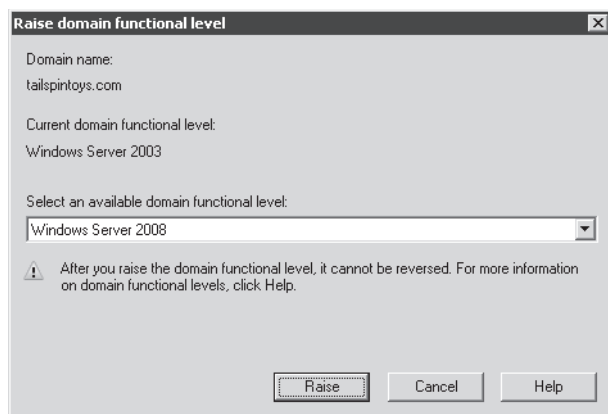


Figure 12-1 The Raise Domain Functional Level dialog box

IMPORTANT One-way operation

Raising the domain functional level is a one-way operation. You cannot roll back to a previous domain functional level.

You can also raise the domain functional level by using the Active Directory Users And Computers snap-in. Right-click the domain and choose Raise Domain Functional Level, or right-click the root node of the snap-in and choose Raise Domain Functional Level from the All Tasks menu.

Forest Functional Levels

Just as domain functional levels enable certain domain-wide functionality and determine the versions of Windows that are supported for domain controllers in the domain, forest functional levels enable forest-wide functionality and determine the operating systems supported for domain controllers in the entire forest. Windows Server 2008 Active Directory supports three forest functional levels:

- Windows 2000

- Windows Server 2003
- Windows Server 2008

Each functional level is described in the following sections.

Windows 2000

Windows 2000 forest functional level is the baseline, default functional level. At Windows 2000 functional level, domains can be running at any supported domain functional level:

- Windows 2000 Native
- Windows Server 2003
- Windows Server 2008

You can raise the forest functional level after all domains in the forest have been raised to the equivalent domain functional level.

Windows Server 2003

After all domains in the forest are at the Windows Server 2003 domain functional level, and when you do not expect to add any new domains with Windows 2000 Server domain controllers, you can raise the forest functional level to Windows Server 2003. At this forest functional level, domains can be running at the following domain functional levels:

- Windows Server 2003
- Windows Server 2008

The following features are enabled at the Windows Server 2003 forest functional level:

- **Forest trusts** In Lesson 2, you will learn to create trust relationships between forests.
- **Domain rename** You can rename a domain within a forest.
- **Linked-value replication** At Windows 2000 forest functional level, a change to a group's membership results in the replication of the entire multivalued *member* attribute of the group. This can lead to increased replication traffic on the network and the potential loss of membership updates when a group is changed concurrently at different domain controllers. It also leads to a recommended cap of 5,000 members in any one group. Linked-value replication, enabled at the Windows Server 2003 forest functional level, replicates an individual membership change rather than the entire *member* attribute. This uses less bandwidth and prevents you from losing updates when a group is changed concurrently at different domain controllers.
- **Support for read-only domain controllers** Chapter 8 discussed read-only domain controllers (RODCs). RODCs are supported at the Windows Server 2003 forest functional level. The RODC itself must be running Windows Server 2008.

Quick Check

- You want to add an RODC to a domain with Windows Server 2003 domain controllers. The domain is at the Windows Server 2003 functional level and already includes one Windows Server 2008 domain controller. The forest is at the Windows 2000 functional level. Which two things must you do prior to adding the RODC?

Quick Check Answer

- You must raise the forest functional level to Windows Server 2003, and you must run *Adprep /rodcprep*.

- **Improved Knowledge Consistency Checker (KCC) algorithms and scalability** The intersite topology generator (ISTG) uses improved algorithms that enable AD DS to support replication in forests with more than 100 sites. At the Windows 2000 forest functional level, you must manually intervene to create replication topologies for forests with hundreds of sites. Additionally, the election of the ISTG uses an algorithm that is more efficient than at Windows 2000 forest functional level.
- **Conversion of *inetOrgPerson* objects to *user* objects** You can convert an instance of an *inetOrgPerson* object, used for compatibility with certain non-Microsoft directory services, into an instance of class *user*. You can also convert a *user* object to an *inetOrgPerson* object.
- **Support for *dynamicObject* auxiliary class** The schema allows instances of the dynamic auxiliary class in domain directory partitions. This object class can be used by certain applications and by developers.
- **Support for application basic groups and LDAP query groups** Two new group types, called *application basic groups* and *LDAP query groups*, can be used to support role-based authorization in applications that use Authorization Manager.
- **Deactivation and redefinition of attributes and object classes** Although you cannot delete an attribute or object class in the schema, at Windows Server 2003 for forest level, you can deactivate or redefine attributes or object classes.

Windows Server 2008

The Windows Server 2008 forest functional level does not add new forest-wide features. However, after the forest is configured to Windows Server 2008 forest functional level, new domains added to the forest will operate at Windows Server 2008 domain functional level by default. At this forest functional level, all domains must be at Windows Server 2008 domain functional level, which means that all domain controllers must be running Windows Server 2008.

Raising the Forest Functional Level

Use the Active Directory Domains and Trusts snap-in to raise the forest functional level. Right-click the root node of the Active Directory Domains And Trusts snap-in, and choose Raise Forest Functional Level. The dialog box shown in Figure 12-2 enables you to choose a higher forest functional level.

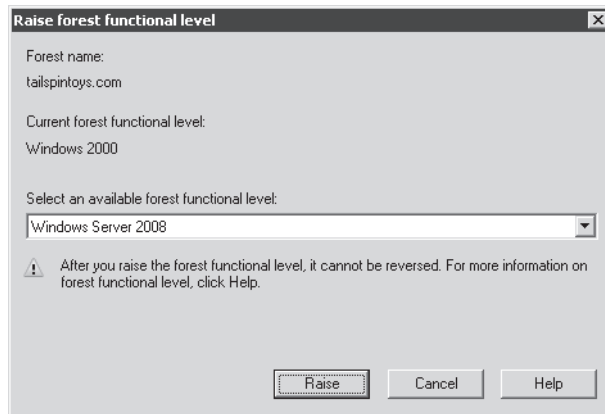


Figure 12-2 The Raise Forest Functional Level dialog box

Raise the forest functional level only when you are confident that you will not add new domains at unsupported domain functional levels. You cannot roll back to a previous forest functional level after raising it.

Exam Tip Be sure to memorize the functionality that is enabled at each domain and forest functional level. Pay particular attention to the capabilities that affect you as an administrator.

Lesson 2: Managing Multiple Domains and Trust Relationships

Previous chapters in this training kit have prepared you to configure, administer, and manage a single domain. However, your enterprise Active Directory infrastructure might include a multidomain forest or even more than one forest. You might need to move objects between domains or restructure your domain model entirely. You might also be required to enable authentication and access to resources across domains and forests. In this lesson, you will learn the skills required to support multiple domains and forests.

After this lesson, you will be able to:

- Design an effective domain and tree structure for AD DS.
- Identify the role of the Active Directory Migration Tool and the issues related to object migration and domain restructure.
- Understand trust relationships.
- Configure, administer, and secure trust relationships.

Estimated lesson time: 60 minutes

Defining Your Forest and Domain Structure

With the perspective you have gained from the previous 11 chapters of this training kit, you are prepared to consider the design of your Active Directory forest, trees, and domains. Interestingly, the best practices guidance regarding forest and domain structure has evolved as enterprises around the world have put Active Directory into production in every conceivable configuration and as the Active Directory feature set has grown.

Dedicated Forest Root Domain

In the early days of Active Directory, it was recommended to create a dedicated forest root domain. You'll recall from Chapter 1 and Chapter 10 that the forest root domain is the first domain in the forest. A dedicated forest root domain's exclusive purpose is to administer the forest infrastructure. It contains, by default, the single master operations for the forest. It also contains highly sensitive groups, such as Enterprise Admins and Schema Admins, that can have far-reaching impact on the forest. The theory was that the dedicated forest root would enhance the security around these forest-wide functions. The dedicated forest root domain would also be less likely to become obsolete and would provide easier transfer of ownership. Underneath the dedicated forest root, according to early recommendations, would be a single global child domain with all the objects one thinks of in a domain: users, groups, computers, and so on. The structure would look something like Figure 12-3.

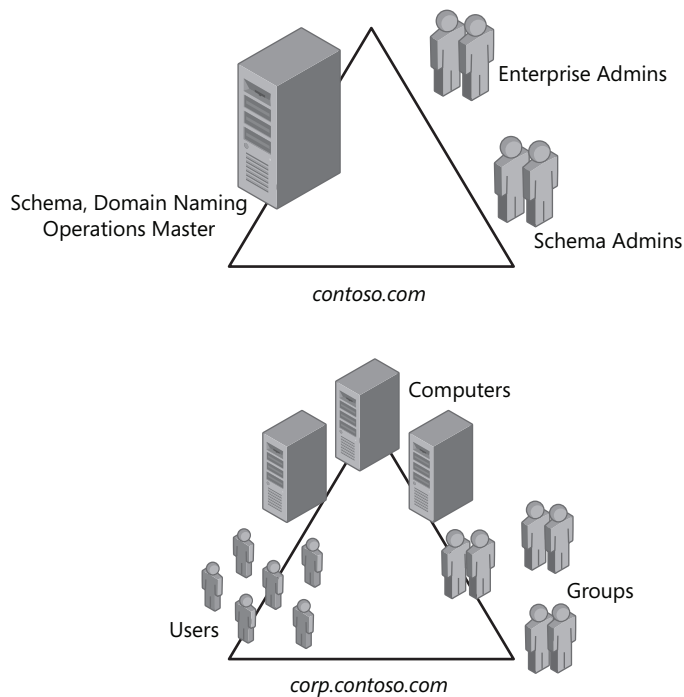


Figure 12-3 Example of a forest root domain

A Single-Domain Forest

NOTE Single-domain forest the new recommendation

It is no longer recommended to implement a dedicated forest root domain for most enterprises. A single-domain forest is the most common design recommendation. There is no single design that is appropriate for every organization, so you must examine the characteristics of your enterprise against the design criteria presented later in this lesson.

After nine years on the market, Active Directory is better understood, and the former recommendation no longer applies. It is now recommended, for most organizations, to build a forest with a single domain. The experience and knowledge that have led to the change in guidance take into account that:

- There are risks and costs associated with any multidomain forest, as you'll learn later in this lesson. A single domain bears the lowest hardware and support cost and reduces certain risks.

- There are not yet tools that enable an enterprise to perform pruning and grafting of Active Directory trees. In other words, you cannot break a domain off of your tree and transplant it in the forest of another enterprise. If that were possible, a dedicated forest root that you could maintain while transferring domains in and out of your forest would make more sense.
- You can implement least-privilege security within a single domain that is at least as secure, if not more secure, than in a forest with a dedicated forest root and a child domain.

Therefore, when you consider your domain design, you should begin with the assumption that you will have a single domain in your forest.

Multiple-Domain Forests

In some scenarios, a multiple-domain forest is required. The important point to remember is that you should never create a multiple-domain forest simply to reflect the organizational structure of your business. That structure—the business units, divisions, departments, and offices—will change over time. The logical structure of your directory service should not be dependent solely on organizational characteristics.

Instead, your domain model should be derived from the characteristics of domains themselves. Certain properties of a domain affect all objects within the domain, and if that consistent effect is not appropriate for your business requirements, you must create additional domains. A domain is characterized by the following:

- **A single domain partition, replicated to all domain controllers** The domain naming context contains the objects for users, computers, groups, policies, and other domain resources. It is replicated to every domain controller in the domain. If you need to partition replication for network topology considerations, you must create separate domains. Consider, however, that Active Directory replication is extremely efficient and can support large domains over connections with minimal bandwidth.
If there are legal or business requirements that restrict replication of certain data to locations where you maintain domain controllers, you need to either avoid storing that data in the domain partition or create separate domains to segregate replication. In such cases, you should also ensure that the global catalog (GC) is not replicating that data.
- **A single Kerberos policy** The default Kerberos policy settings in AD DS are sufficient for most enterprises. If, however, you need distinct Kerberos policies, you will require distinct domains.
- **A single DNS namespace** An Active Directory domain has a single DNS domain name. If you need multiple domain names, you would need multiple domains. However, give serious consideration to the costs and risks of multiple domains before modeling your directory service domains to match arbitrary DNS name requirements.

In domains running domain functional levels lower than Windows Server 2008, a domain can support only one password and account lockout policy. Therefore, in prior versions of Windows, an organization requiring multiple password policies would need multiple domains to support that requirement. This is no longer the case in Windows Server 2008, which, at Windows Server 2008 domain functional level, can support fine-grained password policies.

Adding domains to a forest increases administrative and hardware costs. Each domain must be supported by at least two domain controllers, which must be backed up, secured, and managed. Even more domain controllers might be required to support cross-domain resource access in a geographically distributed enterprise. Additional domains can result in the need to move users between domains, which is more complicated than moving users between OUs. Group Policy objects and access control settings that are common for the enterprise will have to be duplicated for each domain.

These are just a few of the costs associated with a multiple-domain environment. There are also risks involved with having multiple domains. Most of these risks relate to the fact that a domain is not a security boundary—a forest is the security boundary. Within a forest, service administrators can cause forest-wide damage. There are several categories of vulnerability whereby a compromised administrative account, or an administrator with bad intent, could cause denial of service or damage to the integrity of the forest.

For example, an administrator in any domain can create universal groups, the membership of which is replicated to the GC. By creating multiple universal groups and overpopulating the *member* attribute, excessive replication could lead to denial of service on domain controllers acting as domain controllers in other domains. An administrator in any domain could also restore an outdated backup of the directory, which could corrupt the forest.

MORE INFO Security considerations for domain and forest design

For more information about the security considerations related to domain and forest design, see “Best Practices for Delegating Active Directory Administration” at <http://technet2.microsoft.com/windowsserver/en/library/e5274d27-88e5-4043-8f12-a8fa71cbcd521033.mspx>.

Given the costs and risks of multiple domains, it is highly recommended to construct a single-domain forest. The most common driver to multiple-domain forests is a significant requirement related to the replication of the domain naming context.

In a multidomain forest, it might make sense to create a dedicated forest root domain as an empty domain to act as the trust root for the forest. Trust roots will be discussed later in this lesson.

Multiple Trees

Remember that a tree is defined as a contiguous DNS namespace. If you have more than one domain, you can decide whether those domains share a contiguous DNS namespace and form a single tree, as shown at the top of Figure 12-4, or are in a noncontiguous DNS namespace, thus forming multiple trees, as shown on the bottom of Figure 12-4.

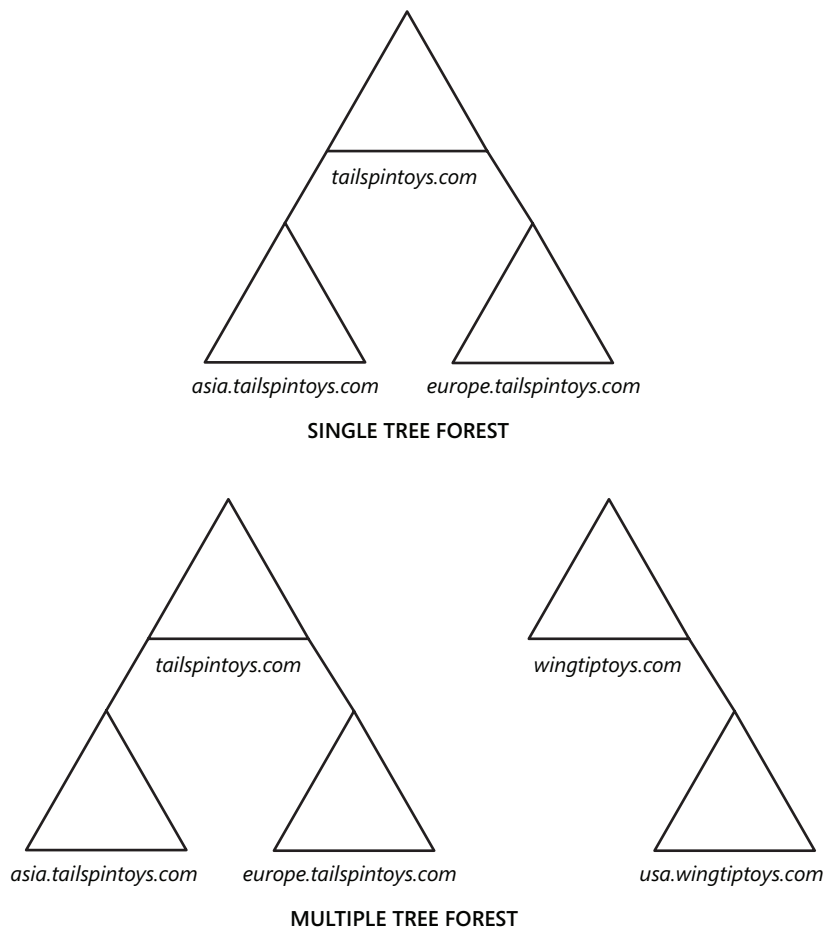


Figure 12-4 Forests with a single tree or multiple trees

Multiple Forests

A forest is an instance of Active Directory. All domains and domain controllers in a forest share replicas of the schema and configuration. Domain controllers that are GC servers host partial attribute sets for all objects in other domains in the forest. Domains in a forest share transitive, two-way trusts, meaning that all users in the domain belong to the Authenticated Users special identity in every domain. The forest's Enterprise Admins, Schema Admins, and Administrators groups in the forest root domain wield significant power over all objects in the forest.

If any of these characteristics of a forest are at odds with your business requirements, you might need multiple forests. In fact, given the market's current concerns with security, many consultants are recommending that organizations design either a single-domain forest or use multiple forests. Cross-forest trusts, discussed later in this lesson, and Active Directory Federation Services (AD FS) make it easier to manage authentication in multiple-forest enterprises.

MORE INFO Planning the architecture

For more information about planning the architecture of an AD DS enterprise, see <http://technet2.microsoft.com/windowsserver2008/en/library/b1baa483-b2a3-4e03-90a6-d42f64b42fc31033.mspx?mfr=true>.

Moving Objects Between Domains and Forests

In multidomain scenarios, you might need to move users, groups, or computers between domains or forests to support business operations. You might need to move large quantities of users, groups, or computers between domains or forests to implement mergers and acquisitions or to restructure your domain model.

In each of these tasks, you move or copy the accounts from one domain (the *source* domain) into another domain (the *target* domain). Domain restructuring terminology, concepts, and procedures apply to *inter-forest migration* (between a Windows NT 4.0 or Active Directory source domain and an Active Directory target domain in a separate forest) and to *intra-forest migration* (that is, the restructuring or moving of accounts between domains in the same forest).

An inter-forest domain restructure preserves the existing source domain and clones (or copies) accounts into the target domain. This nondestructive method enables an enterprise to time the transition and even migrate in phases. Operations go uninterrupted because both domains are maintained in parallel to support operations for users in either domain. This method also provides a level of rollback because the original environment remains unaltered in any significant way. After the migration is complete, you can simply decommission the source domain by moving any remaining accounts, member servers, and workstations into the new domain and then taking source DCs offline, at which point, you can redeploy those DCs for roles in the new domain.

An intra-forest migration involves moving objects from the source domain to the target domain without decommissioning the source domain. After you have migrated objects, you can restructure your domains to consolidate operations and build a domain and OU structure that more accurately reflects your administrative model. Many organizations consolidate multiple domains into one Active Directory domain. This consolidation can result in cost savings and simplified administration by reducing administrative complexity and the cost of supporting your Active Directory environment.

Understanding the Active Directory Migration Tool

The Active Directory Migration Tool version 3 (ADMT v3) can perform object migration and security translation tasks. You can download ADMT v3 from <http://go.microsoft.com/fwlink/?LinkID=75627>. On that page, you will also find a detailed guide to the tool.

You can use ADMT to migrate objects between a source and a target domain. The migration can take place between domains in the same forest (an intra-forest migration) or between domains in different forests (an inter-forest migration). The ADMT provides wizards that automate migration tasks such as migrating users, groups, service accounts, computers, and trusts and performing security translation. You can perform these tasks, using the ADMT console or the command line, where you can simplify and automate the *Admt.exe* command with option files that specify parameters for the migration task. Then, with a simple text file, you can list objects to migrate rather than have to enter each object on the command line. ADMT also provides interfaces that enable you to script migration tasks with languages such as Microsoft VBScript. Run the ADMT console and open the online Help function for details about how to use ADMT from the command line and about scripting the ADMT.

When performing migration tasks, ADMT enables you to simulate the migration so that you can evaluate potential results and errors without making changes to the target domain. Wizards provide the Test The Migration Settings And Migrate Later option. You can then configure the migration task, test the settings, and review the log files and wizard-generated reports. After identifying and resolving any problems, you can perform the migration task. You will repeat this process of testing and analyzing results as you migrate users, groups, and computers and perform security translations.

Security Identifiers and Migration

Uninterrupted resource access is the primary concern during any migration. Further, to perform a migration, you must be comfortable with the concepts of security identifiers (SIDs), tokens, access control lists (ACLs), and *sidHistory*.

SIDs are domain-unique values that are assigned to the accounts of security principals—users, groups, and computers, for example—when those accounts are created. When a user logs on, a token is generated that includes the primary SID of the user account and the SIDs of groups to which the user belongs. The token thus represents the user with all the SIDs associated with the user and the user's group memberships.

Resources are secured using a security descriptor (SD) that describes the permissions, ownership, extended rights, and auditing of the resource. Within the SD are two ACLs. The system ACL (SACL) describes auditing. The discretionary ACL (DACL) describes resource access permissions. Many administrators and documents refer to the DACL as the ACL. The DACL lists permissions associated with security principals. Within the list, individual access control entries (ACEs) link a specific permission with the SID of a security principal. The ACE can be an allow or deny permission.

When a user attempts to access a resource, the Local Security Authority Subsystem (LSASS) compares the SIDs in the user's token against the SIDs in the ACEs in the resource's ACL.

When you migrate accounts to a new domain, the accounts are copied or cloned from the source domain to the target domain. New SIDs are generated for the accounts in the target domain, so the SIDs of new accounts will not be the same as the SIDs of the accounts in the source domain. That is, even though the cloned accounts have the same name and many of the same properties, because the SIDs are different, the accounts are technically different and will not have access to resources in the source domain. You have two ways to address this problem: *sidHistory* or security translation:

- ***sidHistory*** Enterprises typically prefer to take advantage of the *sidHistory* attribute to perform effective domain restructuring. The capitalization, which appears odd, reflects the capitalization of the attribute in the AD schema. AD security principals (which include users, groups, and computers) have a principal SID and a *sidHistory* attribute, which can contain one or more SIDs that are also associated with the account. When an account is copied to a target domain, the unique principal SID is generated by Active Directory in the target domain. Optionally, the *sidHistory* attribute can be loaded with the SID of the account in the source domain. When a user logs on to an Active Directory domain, the user's token is populated with the principal SID and the *sidHistory* of the user account and groups to which the user belongs. The LSASS uses the SIDs from the *sidHistory* just like any other SID in the token to maintain the user's access to resources in the source domain.

- **Security translation** Security translation is the process of examining each resource's SD, including its ACLs, identifying each SID that refers to an account in the source domain and replacing that SID with the SID of the account in the target domain. The process of remapping ACLs (and other elements in the SD) to migrated accounts in the target domain is also called re-ACLing. As you can imagine, security translation or re-ACLing would be a tedious process to perform manually in anything but the simplest environment. Migration tools such as ADMT automate security translation. ADMT can translate the SDs and policies of resources in the source domain to refer to the corresponding accounts in the target domain. Specifically, ADMT can translate:
 - File and folder permissions.
 - Printer permissions.
 - Share permissions.
 - Registry permissions.
 - User rights.
 - Local profiles, which involves changing file, folder, and registry permissions.
 - Group memberships.

In most domain restructuring and migration projects, *sidHistory* is used to maintain access and functionality during the migration; then, security translation is performed.

MORE INFO Domain migration

For more information about domain migration, SIDs, and SID history, see the "Domain Migration Cookbook" at <http://technet.microsoft.com/en-us/library/bb727135.aspx>.

Group Membership

The final concern related to resource access is that of group membership. Global groups can contain members only from the same domain. Therefore, if you clone a user to the target domain, the new user account cannot be a member of the global groups in the source domain to which the source user account belonged.

To address this issue in an inter-forest migration, you will first migrate global groups to the target domain. Those global groups will maintain the source groups' SIDs in their *sidHistory* attributes, thus maintaining resource access. Then, you will migrate users. As you migrate users, ADMT evaluates the membership of the source account and adds the new account to the same group in the target domain. If the group does not yet exist in the target domain, ADMT can create it automatically. In the end, the user account in the target domain will belong to global groups in the target domain. The user and the user's groups will contain the SIDs of the source accounts in their *sidHistory* attributes. Therefore, the user will be able to access resources in the source domain that have permissions assigned to the source accounts.

In an intra-forest migration, the process works differently. A global group is created in the target domain as a universal group so that it can contain users from both the source and the target domains. The new group gets a new SID, but its *sidHistory* is populated with the SID of the global group in the source domain, thereby maintaining resource access for the new group. After all users have been migrated from the source to the target domain, the scope of the group is changed back to global.

Other Migration Concerns

You must address a number of issues in planning for and executing the migration of objects between domains and forests. Each of the concerns is detailed in the ADMT user guide, available from the ADMT download page listed earlier. Among the greatest concerns are:

- **Password migration** ADMT supports migrating user passwords; however, it cannot confirm that those passwords comply with the policies of the target domain regarding password length and complexity. Nonblank passwords will migrate regardless of the target domain password policy, and users will be able to log on with those passwords until they expire, at which time a new, compliant password must be created. If you are concerned about locking down the environment at the time of migration, this might not be a satisfactory process. You might, instead, want to let ADMT configure complex passwords or script an initial password and then force the user to change the password at the first logon.
- **Service accounts** Services on domain computers might use domain-based user accounts for authentication. As those user accounts are migrated to the target domain, services must be updated with the new service account identity. ADMT automates this process.
- **Objects that cannot be migrated** Some objects cannot be seamlessly migrated. ADMT cannot migrate built-in groups such as Domain Admins or the domain local Administrators group. The user guide provides details for working around this limitation.

Exam Tip For the 70-640 exam, you should recognize that the ADMT is used to copy or move accounts between domains. You should also understand that the new account in the target domain will have a new SID but that correct use of the tool can migrate group memberships and can populate *sidHistory* with the SID of the source account.

Understanding Trust Relationships

Whenever you are implementing a scenario involving two or more AD DS domains, it is likely you will be working with *trust relationships*, or *trusts*. It is important that you understand the purpose, functionality, and configuration of trust relationships.

Trust Relationships Within a Domain

In Chapter 5, you were guided through what happens when a domain member server or workstation joins a domain. While in a workgroup, the computer maintains an identity store in the security accounts manager (SAM) database, it authenticates users against that identity store, and it secures system resources only with identities from the SAM database. When the computer joins a domain, it forms a trust relationship with the domain. The effect of that trust is that the computer allows users to be authenticated not by the local system and its local identity store but by the authentication services and identity store of the domain: AD DS. The domain member also allows domain identities to be used to secure system resources. For example, Domain Users is added to the local Users group, giving Domain Users the right to log on locally to the system. Also, domain user and group accounts can be added to ACLs on files, folders, registry keys, and printers on the system. All domain members have similar trust relationships with the domain, enabling the domain to be a central store of identity and a centralized service providing authentication.

Trust Relationships Between Domains

With that foundation, you can extend the concept of trust relationships to other domains. A trust relationship between two domains enables one domain to trust the authentication service and the identity store of another domain and to use those identities to secure resources. In effect, a trust relationship is a logical link established between domains to enable pass-through authentication.

There are two domains in every trust relationship: a trusting domain and a trusted domain. The trusted domain holds the identity store and provides authentication for users in that identity store. When a user in the directory of the trusted domain logs on to or connects to a system in the trusting domain, the trusting domain cannot authenticate that user because the user is not in its data store, so it passes the authentication to a domain controller in the trusted domain. The trusting domain, therefore, *trusts* the trusted domain to authenticate the identity of the user. The trusting domain *extends trust* to the authentication services and the identity store of the trusted domain.

Because the trusting domain trusts the identities in the trusted domain, the trusting domain can use the trusted identities to grant access to resources. Users in a trusted domain can be given user rights such as the right to log on to workstations in the trusting domain. Users or global groups in the trusted domain can be added to domain local groups in the trusting domain. Users or global groups in the trusted domain can be given permissions to shared folders by adding the identities to ACLs in the trusting domain.

The terminology can be confusing, and it is often easier to understand trust relationships with a figure. Figure 12-5 shows a simple diagram of a trust relationship. Domain A trusts Domain B. That makes Domain A the trusting domain and Domain B the trusted domain. If

a user in Domain B connects to or logs on to a computer in Domain A, Domain A will pass the authentication request to a domain controller in Domain B. Domain A can also use the identities from Domain B—users and groups, for example—to grant user rights and resource access in Domain A. A user or group in Domain B can, therefore, be added to an ACL on a shared folder in Domain A. A user or group in Domain B can also be added to a domain local group in Domain A.

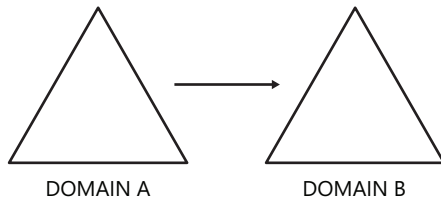


Figure 12-5 Diagram of a simple trust relationship

Exam Tip Trust relationships are highly likely to appear on the 70-640 exam. Be certain that you completely understand the terms *trusted*, *trusting*, and *trust*. It is helpful when taking the exam to draw trust relationships so that you can more easily analyze which domain is trusted and has users and groups that the trusting domain can use to grant access to resources. Always make sure that the trust is extended from the domain with resources, such as computers and shared folders, to the domain with users.

Characteristics of Trust Relationships

Trust relationships between domains can be characterized by two attributes of the trust:

- **Transitivity** Some trusts are not transitive, and others are transitive. In Figure 12-6, Domain A trusts Domain B, and Domain B trusts Domain C. If the trusts are transitive, then Domain A trusts Domain C. If they are not transitive, then Domain A does not trust Domain C. In most cases, you could create a third trust relationship, specifying that Domain A trusts Domain C. With transitive trusts, that third relationship is not necessary; it is implied.

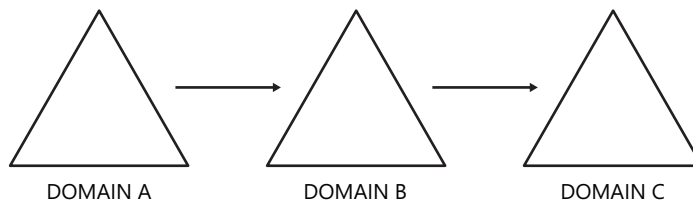


Figure 12-6 A trust relationship example

- **Direction** A trust relationship can be one-way or two-way. In a one-way trust, such as the trust illustrated in Figure 12-5, users in the trusted domain can be given access to resources in the trusting domain, but users in the trusting domain cannot be given access to resources in the trusted domain. In most cases, you can create a second, one-way trust in the opposite direction to achieve that goal. For example, you can create a second trust relationship in which Domain B trusts Domain A. Some trust relationships are by nature two-way. In a two-way trust, both domains trust the identities and authentication services of the other domain.
- **Automatic or Manual** Some trusts are created automatically. Other trusts must be created manually.

Within a forest, all domains trust each other. That is because the root domain of each tree in a forest trusts the forest root domain—the first domain installed in the forest—and each child domain trusts its parent domain. All trusts automatically created should never be deleted and are transitive and two-way. The net result is that a domain trusts the identity stores and authentication services of all other domains in its forest. Users and global groups from any domain in the forest can be added to domain local groups, can be given user rights, and can be added to ACLs on resources in any other domain in the forest. Trusts to other forests and to domains outside the forest must be manually established. With that summary, you can look at the details of trusts within and outside of an Active Directory forest.

Authentication Protocols and Trust Relationships

Windows Server 2003 Active Directory authenticates users with one of two protocols—Kerberos v5 or NT LAN Manager (NTLM). Kerberos v5 is the default protocol used by computers running Windows Server 2008, Windows Vista, Windows Server 2003, Windows XP, and Windows 2000 Server. If a computer involved in an authentication transaction does not support Kerberos v5, the NTLM protocol is used instead.

Kerberos Authentication Within a Domain

When a user logs on to a client running Kerberos v5, the authentication request is forwarded to a domain controller. Each Active Directory domain controller acts as a key distribution center (KDC), a core component of Kerberos. After validating the identity of the user, the KDC on the domain controller gives the authenticated user what is known as a ticket-granting ticket (TGT).

When the user needs to access resources on a computer in the same domain, the user must first obtain a valid session ticket for the computer. Session tickets are provided by the KDC of a domain controller, so the user returns to a domain controller to request a session ticket. The user presents the TGT as proof that he or she has already been authenticated. This enables the KDC to respond to the user's session ticket request without having to re-authenticate the user's identity. The user's session ticket request specifies the computer and the service the user wants to access. The KDC identifies that the service is in the same domain based on the service principal name (SPN) of the requested server. The KDC then provides the user with a session ticket for the service.

The user then connects to the service and presents the session ticket. The server is able to determine that the ticket is valid and that the user has been authenticated by the domain. This happens through private keys, a topic that is beyond the scope of this lesson. The server, therefore, does not need to authenticate the user; it accepts the authentication and identity provided by the domain with which the computer has a trust relationship.

All these Kerberos transactions are handled by Windows clients and servers and are transparent to users themselves.

Kerberos Authentication Within a Forest

Each child domain in a forest trusts its parent domain with an automatic, two-way, transitive trust called a *parent-child trust*. The root domain of each tree in a domain trusts the forest root domain with an automatic, two-way, transitive trust called a *tree-root trust*.

These trust relationships create what is referred to as the *trust path* or *trust flow* in a forest. The trust path is easy to understand with a diagram, as shown in Figure 12-7. The forest consists of two trees, the *tailspintoys.com* tree and the *wingtiptoys.com* tree. The *tailspintoys.com* domain is the forest root domain. On the top of Figure 12-7 is the forest as seen from a DNS perspective. On the bottom of the figure is the trust path. It indicates that the *wingtiptoys.com* tree root domain trusts the *tailspintoys.com* domain.

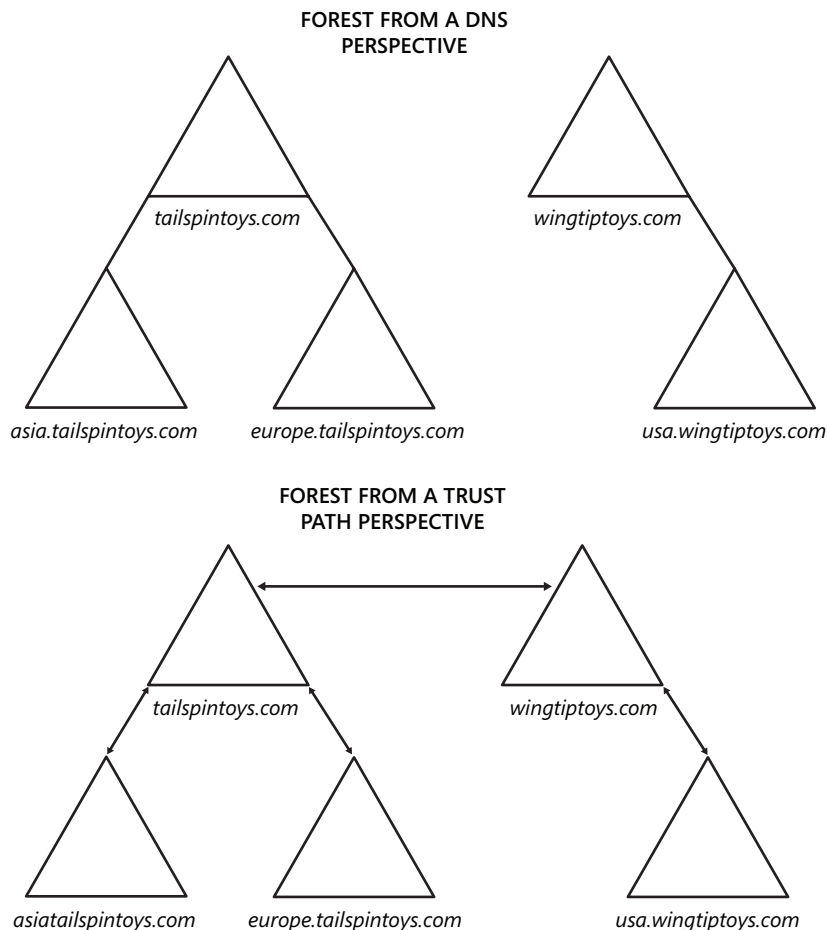


Figure 12-7 An Active Directory forest from a DNS perspective and from a trust path perspective

Kerberos authentication uses the trust path to provide a user in one domain a session ticket to a service in another domain. If a user in *usa.wingtiptoy.com* wants to access a shared folder on a server in *europe.tailspintoys.com*, the following transactions occur:

1. The user logs on to a computer in *usa.wingtiptoy.com* and is authenticated by a domain controller in *usa.wingtiptoy.com*, using the authentication process described in the previous section. The user obtains a TGT for the domain controller in *usa.wingtiptoy.com*. The user wants to connect to a shared folder on a server in *europe.tailspintoys.com*.
2. The user contacts the KDC of a domain controller in *usa.wingtiptoy.com* to request a session ticket for the server in *europe.tailspintoys.com*.
3. The domain controller in *usa.wingtiptoy.com* identifies, based on the SPN, that the desired service resides in *europe.tailspintoys.com*, not in the local domain.

The job of the KDC is to act as a trusted intermediary between a client and a service. If the KDC cannot provide a session ticket for the service because the service is in a trusted domain and not in the local domain, the KDC will provide the client with a *referral* to help it obtain the session ticket it is requesting.

The KDC uses a simple algorithm to determine the next step. If the KDC domain is trusted directly by the service's domain, the KDC gives the client a referral to a domain controller in the service's domain. If not, but if a transitive trust exists between the KDC and the service's domain, the KDC provides the client a referral to the next domain in the trust path.

4. The *usa.wingtiptoys.com* is not trusted directly by *europa.tailspintoys.com*, but a transitive trust exists between the two domains, so the KDC in the *usa.wingtiptoys.com* domain gives the client a referral to a domain controller in the next domain in the trust path, *wingtiptoys.com*.
5. The client contacts the KDC in the referral domain, *wingtiptoys.com*.
6. Again, the KDC determines that the service is not in the local domain and that *europa.tailspintoys.com* does not trust *wingtiptoys.com* directly, so it returns a referral to a domain controller in the next domain in the trust path, *tailspintoys.com*.
7. The client contacts the KDC in the referral domain, *tailspintoys.com*.
8. The KDC determines that the service is not in the local domain and that *europa.tailspintoys.com* trusts *tailspintoys.com* directly, so it returns a referral to a domain controller in the *europa.tailspintoys.com* domain.
9. The client contacts the KDC in the referral domain, *europa.tailspintoys.com*.
10. The KDC in *europa.tailspintoys.com* returns to the client a session ticket for the service.
11. The client contacts the server and provides the session ticket; the server provides access to the shared folder based on the permissions assigned to the user and the groups to which the user belongs.

This process might seem complicated, but recall that it is handled in a way that is completely transparent to the user.

The reverse process occurs if a user from *usa.wingtiptoys.com* logs on to a computer in the *europa.tailspintoys.com* domain. The initial authentication request must traverse the trust path to reach a KDC in the *usa.wingtiptoys.com* domain to authenticate the user.

Although it is not necessary to master the details of Kerberos authentication between domains in a forest for the 70-640 exam, it can help you in the real world to have a basic understanding that cross-domain authentication in a forest follows a trust path.

Manual Trusts

Four types of trusts must be created manually:

- Shortcut trusts
- External trusts
- Realm trusts
- Forest trusts

Each of these types of trusts will be discussed in the following sections.

Creating Manual Trust Relationships

The steps to create trusts are similar across categories of trusts. You must be a member of the Domain Admins or Enterprise Admins group to create a trust successfully.

To create a trust relationship, complete the following steps:

1. Open the Active Directory Domains And Trusts snap-in.
2. Right-click the domain that will participate in one side of the trust relationship, and choose Properties.

You must be running Active Directory Domains And Trusts with credentials that have permissions to create trusts in this domain.

3. Click the Trusts tab.
 4. Click the New Trust button.
- The New Trust Wizard guides you through the creation of the trust.
5. On the Trust Name page, type the DNS name of the other domain in the trust relationship, and then click Next.
 6. If the domain you entered is not within the same forest, you will be prompted to select the type of trust, which will be one of the following:

- Forest
- External
- Realm

If the domain is in the same forest, the wizard knows it is a shortcut trust.

7. If you are creating a realm trust, you will be prompted to indicate whether the trust is transitive or nontransitive.
8. On the Direction Of Trust page, shown in Figure 12-8, select one of the following:
 - Two-Way establishes a two-way trust between the domains.
 - One-Way: Incoming establishes a one-way trust in which the domain you selected in step 2 is the trusted domain, and the domain you entered in step 5 is the trusting domain.

- ❑ One-Way: Outgoing establishes a one-way trust in which the domain you selected in step 2 is the trusting domain, and a domain you entered in step 5 is the trusted domain.

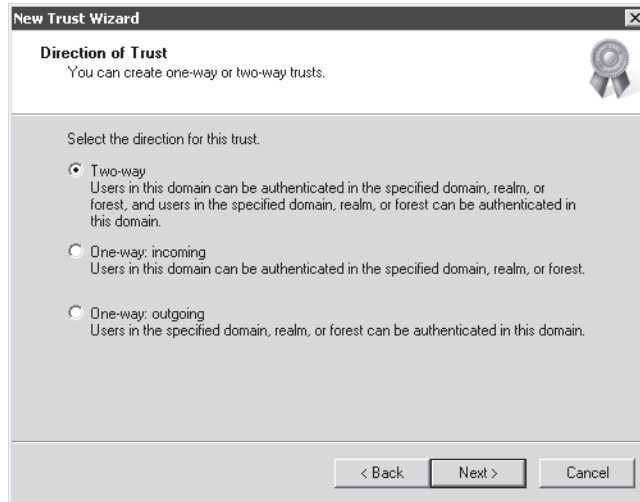


Figure 12-8 The Direction Of Trust page

9. Click Next.
10. On the Sides Of Trust page, shown in Figure 12-9, select one of the following:

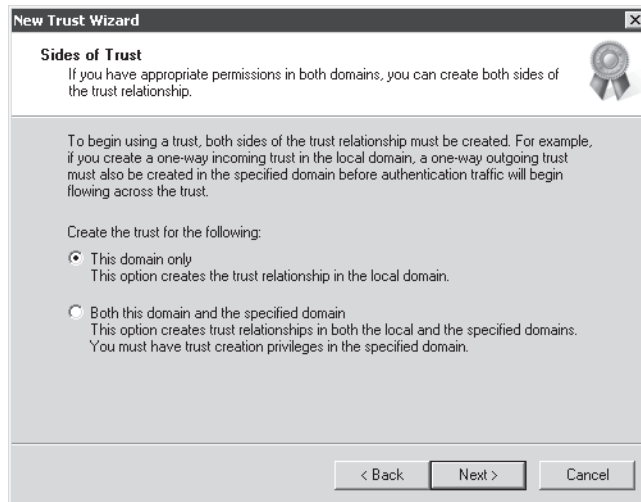


Figure 12-9 The Sides Of Trust page

- ❑ Both This Domain And The Specified Domain establishes both sides of the trust. This requires that you have permission to create trusts in both domains.

- ❑ This Domain Only creates the trust relationship in the domain you selected in step 2. An administrator with permission to create trusts in the other domain must repeat this process to complete the trust relationship.

The next steps will depend on the options you selected in steps 8 and 10. The steps will involve one of the following:

- ❑ If you selected Both This Domain And The Specified Domain, you must enter a user name and password with permissions to create the trust in the domain specified in step 5.
 - ❑ If you selected This Domain Only, you must enter a trust password. A trust password is entered by administrators on each side of a trust to establish the trust. It should not be the administrators' user account passwords. Instead, it should be a unique password used only for the purpose of creating this trust. The password is used to establish the trust, and then the domains change it immediately.
11. If the trust is an outgoing trust, you are prompted to choose one of the following:
 - ❑ Selective Authentication
 - ❑ Domain-Wide Authentication or Forest-Wide Authentication, depending on whether the trust type is an external or forest trust, respectively.

Authentication options are discussed in the section "Securing Trust Relationships," later in this chapter.

12. The New Trust Wizard summarizes your selections on the Trust Selections Complete page. Click Next.

The Wizard creates the trust.

13. The Trust Creation Complete page appears. Verify the settings, and then click Next. You will then have the opportunity to confirm the trust. This option is useful if you have created both sides of the trust or if you are completing the second side of a trust.

If you selected Both This Domain And The Specified Domain in step 8, the process is complete. If you selected This Domain Only in step 8, the trust relationship will not be complete until an administrator in the other domain completes the process:

- If the trust relationship you established is a one-way, outgoing trust, then an administrator in the other domain must create a one-way, incoming trust.
- If the trust relationship you established is a one-way, incoming trust, an administrator in the other domain must create a one-way, outgoing trust.
- If the trust relationship you established is a two-way trust, an administrator in the other domain must create a two-way trust.

MORE INFO Procedures for creating trusts

You can find detailed procedures for creating each type of trust at <http://technet2.microsoft.com/WindowsServer/en/library/f82e82fc-0700-4278-a166-4b8ab47b36db1033.aspx>.

Shortcut Trusts

In an earlier section, you followed 11 steps of the process used to grant a session ticket for a client to access a resource in another domain within a forest. Most of those steps involved referrals to domains on the trust path between the user's domain and the domain of the shared folder. When a user from one domain logs on to a computer in another domain, the authentication request must also traverse the trust path. This can affect performance, and, if a domain controller is not available in a domain along the trust path, the client will not be able to authenticate or to access the service.

Shortcut trusts are designed to overcome those problems by creating a trust relationship directly between child domains in the forest trust path. Two shortcut trusts are illustrated in Figure 12-10.

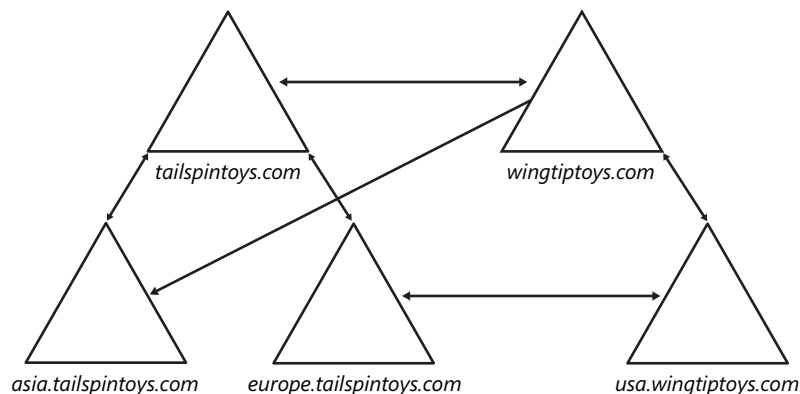


Figure 12-10 Shortcut trusts

Shortcut trusts optimize authentication and session ticket requests between domains in a multidomain forest. By eliminating the trust path, they eliminate the time required to traverse the trust path and, thereby, can significantly improve performance of session ticket requests.

Shortcut trusts can be one-way or two-way. In either case, the trust is transitive. In Figure 12-10, a one-way shortcut trust exists whereby *wingtiptoys.com* trusts *asia.tailspintoys.com*. When a user from *asia.tailspintoys.com* logs on to a computer in *wingtiptoys.com* or requests a resource in *wingtiptoys.com*, the request can be referred directly to a domain controller in the trusted domain, *asia.tailspintoys.com*. However, the reverse is not true. If a user in *wingtiptoys.com* logs on

to a computer in *asia.tailspintoys.com*, the authentication request will traverse the trust path up to *tailspintoys.com* and down to *wingtiptoy.com*.

A two-way shortcut trust is illustrated between *usa.wingtiptoy.com* and *europa.tailspintoys.com*. Users in both domains can be authenticated by and can request resources from computers in the other domain, and the shortcut trust path will be used.

External Trusts

When you need to work with a domain that is not in your forest, you might need to create an external trust. An external trust is a trust relationship between a domain in your forest and a Windows domain that is not in your forest. Examples are shown in Figure 12-11.

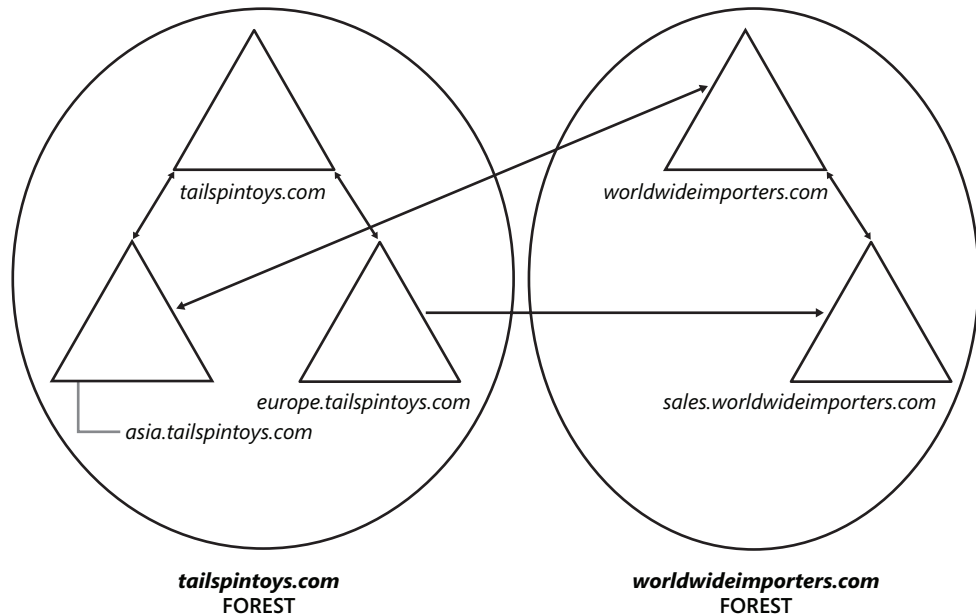


Figure 12-11 An external trust to a domain in another forest

In Figure 12-11, you can see a one-way trust between the *sales.worldwideimporters.com* domain and the *europe.tailspintoys.com* domain. The Europe domain trusts the Sales domain, so users in the Sales domain can log on to computers in the Europe domain or connect to resources in the Europe domain.

Figure 12-11 also shows a two-way trust between the *worldwideimporters.com* domain and the *asia.tailspintoys.com* domain. Users in each domain can be given access to resources in the other domain. Technically, all external trusts are nontransitive, one-way trusts. When you create a two-way external trust, you are actually creating two one-way trusts, one in each direction.

When you create an outgoing external trust, Active Directory creates a foreign security principal object for each security principal in the trusted domain. Those users, groups, and computers can then be added to domain local groups or to ACLs on resources in the trusting domain.

To increase the security of an external trust relationship, you can choose Selective Authentication on the Outgoing Trust Authentication Level page of the New Trust Wizard. Additionally, domain quarantine, also called SID filtering, is enabled by default on all external trusts. Both of these configurations are detailed in the “Securing Trust Relationships” section, later in this chapter.

Realm Trusts

When you need cross-platform interoperability with security services based on other Kerberos v5 implementations, you can establish a realm trust between your domain and a UNIX Kerberos v5 realm. Realm trusts are one-way, but you can establish one-way trusts in each direction to create a two-way trust. By default, realm trusts are nontransitive, but they can be made transitive.

If a non-Windows Kerberos v5 realm trusts your domain, the realm trusts all security principals in your domain. If your domain trusts a non-Windows Kerberos v5 realm, users in the realm can be given access to resources in your domain; however, the process is indirect. When users are authenticated by a non-Windows Kerberos realm, Kerberos tickets do not contain all the authorization data needed for Windows. Therefore, an account mapping system is used. Security principals are created in the Windows domain and are mapped to a foreign Kerberos identity in the trusted non-Windows Kerberos realm. The Windows domain uses only these proxy accounts to evaluate access to domain objects that have security descriptors. All Windows proxy accounts can be used in groups and on ACLs to control access on behalf of the non-Windows security principal. Account mappings are managed through Active Directory Users and Computers.

Forest Trusts

When you require collaboration between two separate organizations represented by two separate forests, you can consider implementing a forest trust. A forest trust is a one-way or two-way transitive trust relationship between the forest root domains of two forests. Figure 12-12 shows an example of a forest trust between the *tailspintoys.com* forest and the *worldwideimporters.com* forest.

A single forest trust relationship allows the authentication of a user in any domain by any other domain in either forest, assuming the forest trust is two-way. If the forest trust is one-way, any user in any domain in the trusted forest can be authenticated by computers in the trusting forest. Forest trusts are significantly easier to establish, maintain, and administer than are separate trust relationships between each of the domains in the forests. Forest trusts are particularly useful in scenarios involving cross-organization collaboration, mergers and

acquisitions, or within a single organization that has more than one forest, to isolate Active Directory data and services.

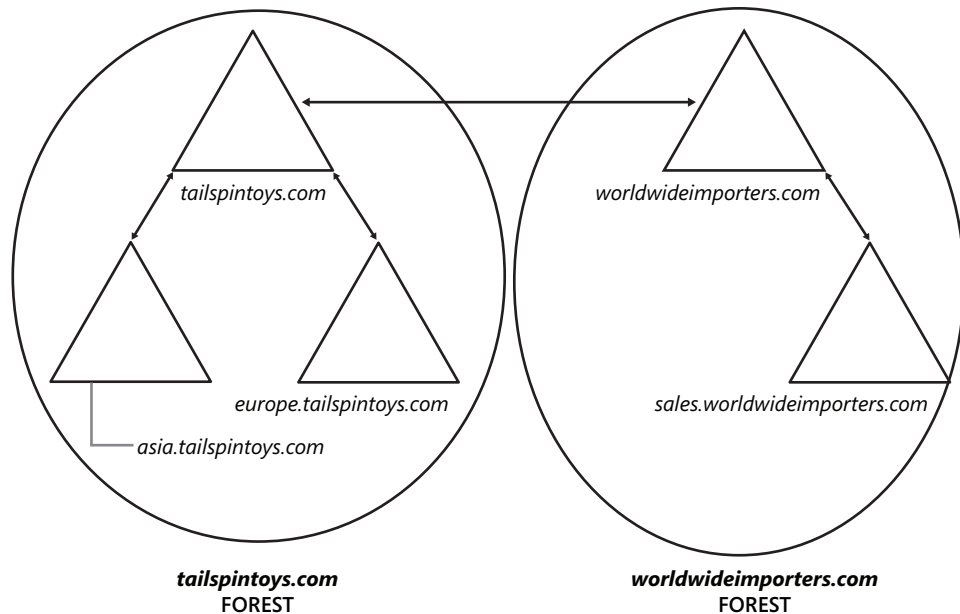


Figure 12-12 A forest trust

When you establish a forest trust relationship, domain quarantine (also called SID filtering) is enabled by default. Domain quarantine is discussed in the “Securing Trust Relationships” section, later in this chapter. You can specify whether the forest trust is one-way, incoming or outgoing, or two-way. As mentioned earlier, a forest trust is transitive, allowing all domains in a trusting forest to trust all domains in a trusted forest. However, forest trusts are not themselves transitive. For example, if the *tailspintoys.com* forest trusts the *worldwideimporters.com* forest, and the *worldwideimporters.com* forest trusts the *northwindtraders.com* forest, those two trust relationships do not allow the *tailspintoys.com* forest to trust the *northwindtraders.com* forest. If you want those two forests to trust each other, you must create a specific forest trust between them.

Several requirements must be met before you can implement a forest trust. The forest functional level must be Windows Server 2003 or later. In addition, you must have a specific DNS infrastructure to support a forest trust.

MORE INFO DNS requirements for a forest trust

You can learn about the DNS requirements for a forest trust at <http://technet2.microsoft.com/WindowsServer/en/library/f5c70774-25cd-4481-8b7a-3d65c86e69b11033.mspx>.

Administering Trusts

If you are concerned that a trust relationship is not functioning, you can validate a trust relationship between any two Windows domains. You cannot validate a trust relationship to a Kerberos v5 realm. To validate a trust relationship, complete the following steps.

1. Open Active Directory Domains And Trusts.
2. In the console tree, right-click the domain that contains the trust that you want to validate, and then click Properties.
3. Click the Trusts tab.
4. Select the trust you want to validate.
5. Click Properties.
6. Click Validate.
7. Do one of the following, and then click OK:
 - Click Yes, Validate The Incoming Trust. Enter credentials that are members of the Domain Admins or Enterprise Admins groups in the reciprocal domain.
 - Click No, Do Not Validate The Incoming Trust. It is recommended that you repeat this procedure for the reciprocal domain.

You can also verify a trust from the command prompt by typing the following command:

```
netdom trust TrustingDomainName /domain:TrustedDomainName /verify
```

There can also be reason to remove a manually created trust. To do so, follow these steps.

1. Open Active Directory Domains And Trusts.
2. In the console tree, right-click the domain that contains the trust you want to validate, and then click Properties.
3. Click the Trusts tab.
4. Select the trust you want to remove.
5. Click Remove.
6. Do one of the following, and then click OK:
 - Click Yes, Remove The Trust From Both The Local Domain And The Other Domain. Enter credentials that are members of the Domain Admins or Enterprise Admins groups in the reciprocal domain.
 - Click No, Remove The Trust From The Local Domain Only. It is recommended that you repeat this procedure for the reciprocal domain.
7. To delete a manually created trust from the command prompt, use the *Netdom.exe* command with the following syntax:

```
netdom trust TrustingDomainName /domain:TrustedDomainName  
/remove [/force] /UserD:User /PasswordD:*
```

The *UserD* parameter is a user with credentials in the Enterprise Admins or Domain Admins group of the trusted domain. Specifying the *PasswordD:** parameter causes *Netdom.exe* to prompt you for the password to the account. The */force* switch is required when removing a realm trust.

NOTE Command-line tools to manage and test trust relationships

The Windows Domain Manager, *Netdom.exe*, and other command-line tools can be used to manage and test trust relationships. See <http://technet2.microsoft.com/windowsserver/en/library/108124dd-31b1-4c2c-9421-6adbc1ebceca1033.msp?mfr=true> for details regarding these commands.

Securing Trust Relationships

When you configure a trust relationship that enables your domain to trust another domain, you open up the possibility for users in the trusted domain to gain access to resources in your domain. The following sections examine components related to the security of a trusting domain's resources.

Authenticated Users

A trust relationship itself does not grant access to any resources; however, it is likely that by creating a trust relationship, users in the trusted domain will have immediate access to a number of your domain's resources. This is because many resources are secured with ACLs that give permissions to the Authenticated Users group.

Membership in Domain Local Groups

As you learned in Chapter 4, "Groups," the best practice for managing access to a resource is to assign permissions to a domain local group. You can then nest users and groups from your domain into the domain local group and, thereby, grant them access to the resource. Domain local security groups can also include users and global groups from trusted domains as members. Therefore, the most manageable way to assign permissions to users in a trusted domain is to make them or their global groups members of a domain local group in your domain.

ACLs

You can also add users and global groups from a trusted domain directly to the ACLs of resources in a trusting domain. This approach is not as manageable as the previous method, using a domain local group, but it is possible.

Transitivity

When you create a realm trust, the trust is nontransitive by default. If you make it transitive, you open up the potential for users from domains and realms trusted by the Kerberos v5 realm to gain access to resources in your domain. It is recommended to use nontransitive trusts unless you have a compelling business reason for a transitive realm trust.

Domain Quarantine

By default, domain quarantine, also called SID filtering, is enabled on all external and forest trusts. When a user is authenticated in a trusted domain, the user presents authorization data that includes the SIDs of the user's account in the groups to which the user belongs. Additionally, the user's authorization data includes security identifiers from other attributes of the user and his or her groups.

Some of the SIDs presented by the user from the trusted domain might not have been created in the trusted domain. For example, if a user is migrated from one domain into another, a new SID is assigned to the migrated account. The migrated account will, therefore, lose access to any resources that had permissions assigned to the SID of the user's former account. To enable the user to continue to access such resources, an administrator performing a migration can specify that the *sidHistory* attribute of the user's migrated account will include the former account's SID. When the user attempts to connect to the resource, the original SID in the *sidHistory* attribute will be authorized for access.

In a trusted domain scenario, it is possible that a rogue administrator could use administrative credentials in the trusted domain to load SIDs into the *sidHistory* attribute of a user that are the same as SIDs of privileged accounts in your domain. That user would then have inappropriate levels of access to resources in your domain.

Domain quarantine prevents this problem by enabling the trusting domain to filter out SIDs from the trusted domain that are not the primary SIDs of security principals. Each SID includes the SID of the originating domain, so when a user from a trusted domain presents the list of the user's SIDs and the SIDs of the user's groups, SID filtering instructs the trusting domain to discard all SIDs without the domain SID of the trusted domain.

Domain quarantine is enabled by default for all outgoing trusts to external domains and forests. Disable domain quarantine only if one or more of the following are true:

- You have extremely high levels of confidence in the administrators of the trusted domain.
- Users or groups have been migrated to the trusted domain with their SID histories preserved, and you want to grant those users or groups permissions to resources in the trusting domain based on the *sidHistory* attribute.

To disable domain quarantine, type the following command:

```
netdom trust TrustingDomainName /domain:TrustedDomainName /quarantine:no
```

To re-enable domain quarantine, type this command:

```
netdom trust TrustingDomainName /domain:TrustedDomainName /quarantine:yes
```

Exam Tip You might encounter either term—*domain quarantine* or *SID filtering*—on the 70-640 exam. Remember that this procedure is used so that users from a trusted domain are authorized using only the SIDs that originate in the trusted domain. An effect of domain quarantine is that the trusting domain ignores SIDs in the *sidHistory* attribute, which typically contains the SIDs of accounts from a domain migration.

Selective Authentication

When you create an external trust or a forest trust, you can control the scope of authentication of trusted security principals. There are two modes of authentication for an external or forest trust:

- Selective authentication
- Domain-wide authentication (for an external trust) or forest-wide authentication (for a forest trust)

If you choose domain-wide or forest-wide authentication, all trusted users can be authenticated for access to services on all computers in the trusting domain. Trusted users can, therefore, be given permission to access resources anywhere in the trusting domain. With this authentication mode, you must have confidence in the security procedures of your enterprise and in the administrators who implement those procedures so that inappropriate access is not assigned to trusted users. Remember, for example, that users from a trusted domain or forest are considered Authenticated Users in the trusting domain, so any resource with permissions granted to Authenticated Users will be immediately accessible to trusted domain users if you choose domain-wide or forest-wide authentication.

If, however, you choose selective authentication, all users in the trusted domain are trusted identities; however, they are allowed to authenticate only for services on computers that you have specified. For example, imagine that you have an external trust with a partner organization's domain. You want to ensure that only users from the marketing group in the partner organization can access shared folders on only one of your many file servers. You can configure selective authentication for the trust relationship and then give the trusted users the right to authenticate only for that one file server.

To configure the authentication mode for a new outgoing trust, use the Outgoing Trust Authentication Level page of the New Trust Wizard. Configure the authentication level for an existing trust, open the properties of the trusting domain in Active Directory Domains And Trusts, select the trust relationship, click Properties, and then click the Authentication tab, shown in Figure 12-13.

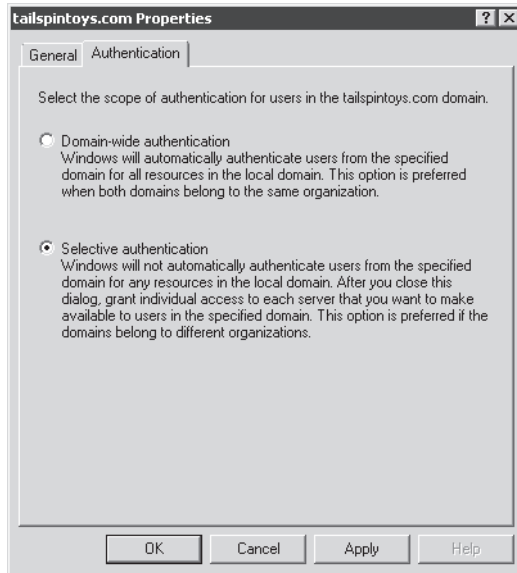


Figure 12-13 The Authentication tab of a trust relationship's Properties dialog box

After you have selected Selective Authentication for the trust, no trusted users will be able to access resources in the trusting domain, even if those users have been given permissions. The users must also be assigned the Allowed To Authenticate permission on the computer object in the domain. To assign this permission, open the Active Directory Users And Computers snap-in and make sure that Advanced Features is selected in the View menu. Open the properties of the computer to which trusted users should be allowed to authenticate—that is, the computer that trusted users will log on to or that contains resources to which trusted users have been given permissions. On the Security tab, add the trusted users or a group that contains them, and select the Allow check box for the Allowed To Authenticate permission, shown in Figure 12-14.

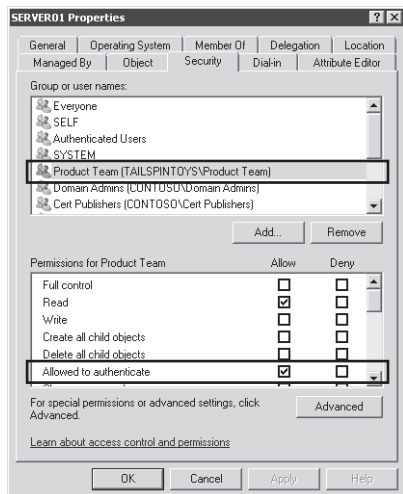


Figure 12-14 Assigning the Allowed To Authenticate permission to a trusted group

Quick Check

- You have configured selective authentication for an outgoing trust to the domain of a partner organization. You want to give a group of auditors in the partner organization permission to a shared folder on SERVER32. Which two permissions must you configure?

Quick Check Answer

- You must assign the auditors the Allowed To Authenticate permission for the SERVER32 computer object. You must also give the auditors NTFS permissions to the shared folder.