Module 9

Using Active Directory and Domains

CRITICAL SKILLS

9.1 Review the Active Directory Environment

9.2 Install Active Directory

9.3 Understand Active Directory's Structure and Configuration

9.4 Replicate Active Directory among Sites

Probably the single biggest change in Windows 2000 over Windows NT was the addition of Active Directory (AD). In Windows Server 2003 AD has been enhanced, making it an even more important part of the operating system. *Active Directory* provides a single reference, called a *directory service,* to all the objects in a network, including users, groups, computers, printers, policies, and permissions. For a user or an administrator, AD provides a single hierarchical view from which to access and manage all of the network's resources. AD utilizes Internet protocols and standards, including Kerberos, Secure Sockets Layer (SSL), and Transport Layer Security (TLS) authentication; the Lightweight Directory Access Protocol (LDAP); and the Domain Name System (DNS). AD requires one or more domains in which to operate.

A *domain,* as used within Windows NT, 2000 and Windows Server 2003, is a collection of computers that share a common set of policies, a name, and a database of their members. A domain must have one or more servers that serve as *domain controllers* and store the database, maintain the policies, and provide the authentication of domain logons. A *domain,* as used within the Internet, is the highest segment of an Internet domain name and identifies the type of organization; for example, .gov for government agencies, and .net for Internet service providers (ISPs). A *domain name* is the full Internet address used to reach one entity registered on the Internet. For example, www.osborne.com or www.mit.edu.

CRITICAL SKILL 9.1

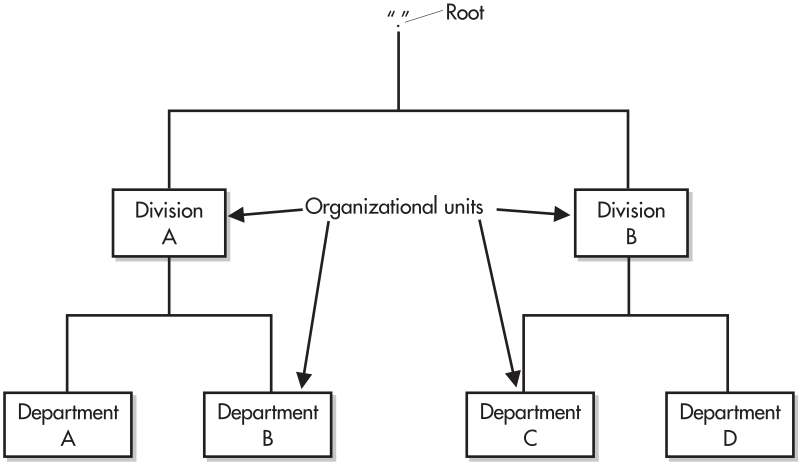
Review the Active Directory Environment

AD plays two basic functions within a network: that of a directory service containing a hierarchical listing of all the objects within the network, and that of an authentication and security service that controls and provides access to network resources. These two roles are very different in nature and focus, but they combine together to provide increased user capabilities while decreasing administrative overhead. At its core, Windows Server 2003 AD is a directory service that is integrated into DNS, plus a user authentication service for the Windows Server 2003 operating system. This explanation, however, introduces a few new terms and involves a number of complex concepts.

While AD is both a directory and a directory service, the terms are not interchangeable. In Windows Server 2003 networking, a *directory* is a listing of the objects within a network. A hierarchical directory has a structure with a top-to-bottom configuration that allows for the logical grouping of objects, such that lower-level objects are logically grouped and contained in higher-level objects for as many levels as you want. This grouping can be based on a number of different criteria, but the criteria should be logical and consistent throughout the directory structure.

Two of the more common directory structures in use within networks are based on object function (such as printers, servers, and storage devices) and organizational responsibility (such as marketing, accounting, and manufacturing). The organizational model allows you to store objects in groups, or *containers,* based on where they are in an organization, which might have its own structure, such as departments within divisions. A particular department would be the first organizational point within an organization. A container holding all the objects in a department is called an *organizational unit (OU)* and is itself grouped into higher-level OUs based on the logical structure.

After you create a group of OUs, you may find that the structure causes your directory to be cluttered and/or awkward to navigate. As a result, you may need to change your network to have more high-level OUs or more low-level OUs. At the top of all directories is the master OU that contains all the other OUs. This directory is referred to as the *root* and is normally designated by a single period. Such a hierarchical structure might look like this:



AD is just as basic as the organization just displayed. However, much of AD's core structure has already been mapped out by Microsoft and is consistent throughout all Windows Server 2003 implementations. For this reason, some of the containers, which are just OUs, have been assigned specific names and roles within AD. As this preconfigured directory structure is explained in the rest of the module, don't let the terms and names confuse you. Everything is still simply a collection of objects within OUs.

The "service" in "directory service" adds to a server features that are not otherwise available. Primarily, a directory service provides access to the directory of information, as well as to services that provide information about the location, access methods, and access rights for the objects within the directory service tree. This means that users can access a single directory and then be directly connected to a variety of other servers and services that appear to all be coming from the original directory. Much of this module discusses the different kinds of objects and methods of access that AD can provide to both users and administrators.

NOTE

AD, Microsoft Exchange, and Novell Directory Services (NDS) are all based on the X.500 standard, which is an internationally recognized standard used to create a directory structure. Specifically, AD is based on the newer X.509 version of the X.500 family.

Integration with DNS

Much of AD's structure and services, as well as the namespace that it uses, is based on DNS. (*Namespace* is the addressing scheme that is used to locate objects on the network. Both AD and the Internet use a hierarchical namespace separated by periods, as described earlier in this module.) How AD uses DNS will be discussed in a moment, but it is necessary to first look at the structure and workings of DNS and how it is used to build the AD foundation.

All servers and services on the Internet are given an Internet Protocol (IP) numerical address, and all Internet traffic uses this IP number to reach its destination. IP numbers change, and may host multiple services at the same time. In addition, most people have a hard time remembering large arbitrary numbers such as IP addresses. IP addresses are decimal-based descriptions of binary numbers without a discernable pattern. DNS services were created to solve these problems by allowing servers and other objects on the network to be given a user-friendly name, which DNS translates to an IP number. For example, a user-friendly name such as mail.osborne.com might be translated, or *resolved,* in a DNS server to an IP address such as 168.143.56.34, which the network can then use to locate the desired resource.

DNS servers use hierarchical directory structures, just like the example described at the beginning of the module. At the core of DNS servers are root domains with a root directory, which is described by a single period. The first groups of OUs below the root are the various types of domains that can exist, for example, COM, NET, ORG, US, GOV, and EDU. Over 250 of these top-level domains are controlled within the United States by InterNIC, an arm of the U.S. Department of Commerce and run by a private, nonprofit corporation named the Internet Corporation for Assigned Names and Numbers (ICANN), which controls a number of root servers that contain a listing of all the entries within each subdomain.

The next group of OUs following the ".COMs" consists of domain names, such as coke.com, microsoft.com, and osborne.com. These domains are registered and administered by the organizations or individuals who own them. A number of companies have contracted with InterNIC/ICANN to register new domain names added to the Internet; you can see an alphabetical list of those companies at http://www.internic.com/alpha.html.

A domain name, such as osborne.com, can contain both additional OUs, called *subdomains,* and actual server objects. In the example previously given, mail.osborne.com, the mail server is an object in the osborne.com domain. A server name such as mail.osborne.com that contains all OUs between itself and the root is called a fully qualified domain name (FQDN). Figure 9-1 shows the actual name resolution process required when a client such as the one in the lower left of Figure 9-1 requests a DNS server to resolve an FQDN to an IP address by going up and back down the chain of DNS servers.

Active Directory and Domains

AD and DNS share the same central OU, called a *domain.* For those familiar with Windows NT 4 or Windows 2000, the domain concept should be a familiar one. A domain is a central authentication and directory service that contains all the information for a group of computers.

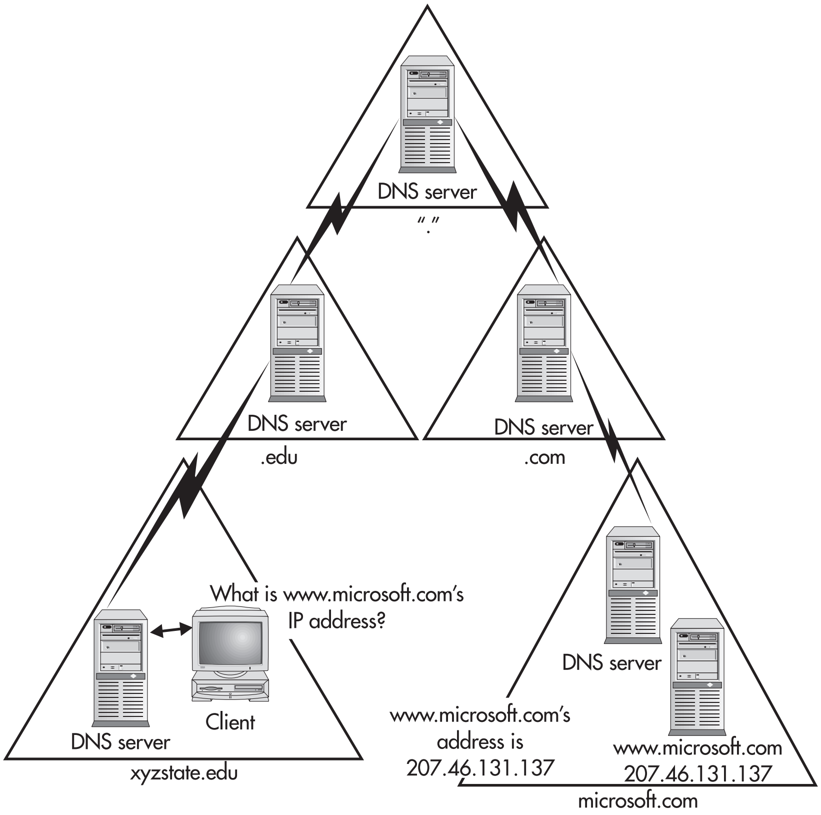


Figure 9-1 The process of resolving a domain name to an IP address

NT 4 had a number of significant limitations in the utilization of domains. Primarily, domains were not actual directory servers, because they contained only users, groups, and computers in a nonhierarchical structure. Although all computers were able to use the central repository of information for authentication issues, the directory was not available for object location or resource management. In AD, the core features and look of legacy NT domains are still present, but they have been greatly extended. Among the many enhancements, which will be discussed over the course of this module, is the ability of AD domains to scale to virtually any size, as opposed to the 40,000-object limit placed on NT domain structures. Another enhancement is AD's ability to form transitive two-way trusts with other domains in the network (this will be discussed further, later in the module).

The close integration between AD and DNS can, at first, be a little confusing. Looking at AD and DNS, it is easy to think they are actually the same thing, because they use the same names and naming scheme. However, this is not true. In actuality, DNS and AD are separate directory services that are using the same names for different namespaces. Each directory contains different objects, and different information about the objects in its own database. However, those object names, as well as the directory structure, often are identical.

Every Windows Server 2003 computer has an FQDN. This is a combination of its own computer name and the domain name of the domain in which it currently resides. For example, Windows Server 2003 computers in the Osborne domain may very well have a computer name equal to *computername*.osborne.com. However, that same computer may in fact be a member of the subdomain of editorial.osborne.com. In this case, the computer's FQDN would actually be *computername*.editorial.osborne.com.

DNS Directories

A DNS directory doesn't really store objects in its database. Rather, DNS stores domains, the access information for each domain, and the access information for the objects (such as servers and printers) within the domain. The access information is normally just the FQDN and the related IP address. All queries for an object's IP address will match the FQDN in the request to the FQDN index in the DNS directory and return (*resolve to*) the IP address. In some cases, the access information (or *resource reference*) simply points to another object (or *resource*) within the same or a different DNS domain.

A standard DNS domain is not capable of reversing this process by returning an FQDN when provided with an IP address. To make this kind of resolution possible, a reverse lookup zone is required, as discussed in Module 8. These domains are referenced as "in-addr-arpa" domains within the DNS hierarchy.

Among the other special functions of DNS that add features to a network is the Mail Exchanger reference that can be added to a DNS domain name. A Mail Exchanger reference (referred to as MX) enables mail servers to locate the mail servers of other domains to allow for the transferring of e-mail across the Internet.

Active Directory Services

AD services contain a lot more information than what is available in DNS directories, even though the names and structure are nearly identical. AD resolves all information requests for objects within its database using LDAP queries. The AD server is able to provide a varied amount of information about each object within its database. The information that AD can provide includes, but isn't limited to, the following:

- Username
- Contact information, such as physical address, phone numbers, and e-mail address
- Administrative contacts

- Access permissions
- Ownerships
- Object attributes, such as object name features; for example, Color Laser Jet Printer, 20 sheets per minute, duplex printing

Although DNS does not require AD, AD requires a DNS server to be in place and functioning correctly on the network before a user will be able to find the AD server. With Windows Server 2003 moving entirely to Internet standards for its network operating system, a method of locating network services had to be found other than using the NetBIOS broadcasts used in Windows NT. This was done through the use of a new DNS domain type known as *dynamic DNS (DDNS)* domains. A DDNS domain, which is integrated into AD, allows all domain controllers to use the same database, which is automatically updated as new Windows Server 2003 computers are added and removed from the network. The DDNS domain also allows DNS to function with networks based on DHCP (Dynamic Host Configuration Protocol), where the IP addresses of the network objects are constantly changing. Besides providing the name resolution for the network, the DDNS domains also contain a listing of all the domains and domain controllers throughout the network. This means that as new Windows Server 2003 systems are added to a network, they will query the DDNS servers to get the name and connection information, including IP address, of the domain controllers they are closest to.

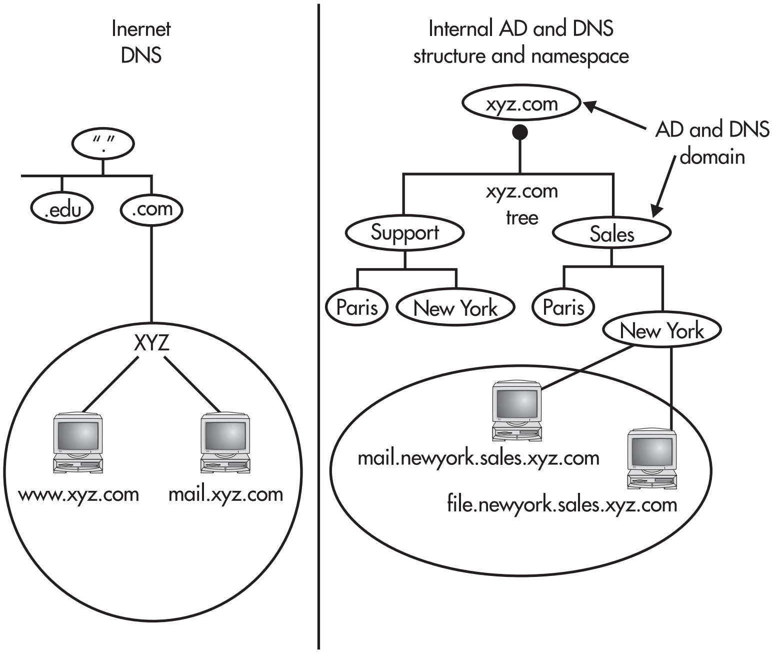
NOTE

In Windows NT installations, Windows Internet Naming Service (WINS) servers provided new workstations with the location of the domain controllers. To allow for compatibility with Windows 3.11, Windows 95, and Windows 98 workstations not running the AD client, WINS servers are still required on the network. Both Windows Server 2003 and Windows 2000 servers as well as legacy NT servers have the ability to host WINS services and integrate them with DNS.

Active Directory and the Global DNS Namespace

AD domains are designed and intended to exist within the naming scheme of the global DNS domain operated through the Internet. This means that, by design, the DNS domain of your network would also match the AD domain-naming scheme. In some organizations, migration from a legacy NT local area network (LAN) is difficult, because independent LAN and Internet domains are already in place and entrenched. In this case, DDNS servers can be used for the LAN AD domain and hosted on internal DNS servers. The external servers providing Internet web hosting services, such as the Simple Mail Transfer Protocol (SMTP) and the Hypertext

Transfer Protocol (HTTP), would still use the Internet DNS structure and provide the necessary resource mapping in each domain to allow for coexistence, as shown here:



NOTE

The functions provided by the DDNS domain that are required for AD are the listing of the domain information and the location of the domain servers. These functions are provided by a resource object called a Service Location Resource (SRV). SRVs are not unique to Windows DDNS domains, and therefore some third-party DNS server products work with AD. However, configuring these third-party DNS servers to integrate with AD can be a significant undertaking.

CRITICAL SKILL 9.2

Install Active Directory

There are two ways to install AD: into an existing domain, or to form a new domain. (For those a little more familiar with AD, some issues regarding forests and trees are affected here, as well, but those will be discussed later in the module.) Installing AD on a server turns the server into a *domain controller,* a server that hosts a central database of all users and groups within the domain and manages all domain-related functions, such as user logon authentication and trust relationships. This process can be done on existing NT servers or on newly installed Windows Server 2003 or Windows 2000 AD servers. Windows 2000 was the first Windows platform that allowed for the promotion of member servers to domain controllers. If a domain contains multiple AD servers, then the AD services can be removed from a domain controller and the server can be returned to a standard member server within the domain.

When installing AD into an existing legacy NT-based domain, the primary domain controller (PDC) of the domain has to be a Windows Server 2003 or Windows 2000 server running AD. This is obviously only an issue in cases in which both Windows Server 2003 or Windows 2000 and legacy NT domain controllers exist in the domain (called a mixed-mode network). In cases where you are installing the first AD server in an existing legacy domain, the existing PDC will have to be upgraded to Windows Server 2003 or Windows 2000, and then AD can be added. A Windows Server 2003 or Windows 2000 server can exist without AD in an NT legacy domain, but before AD can be added, the PDC has to be upgraded to Windows Server 2003 or Windows 2000.

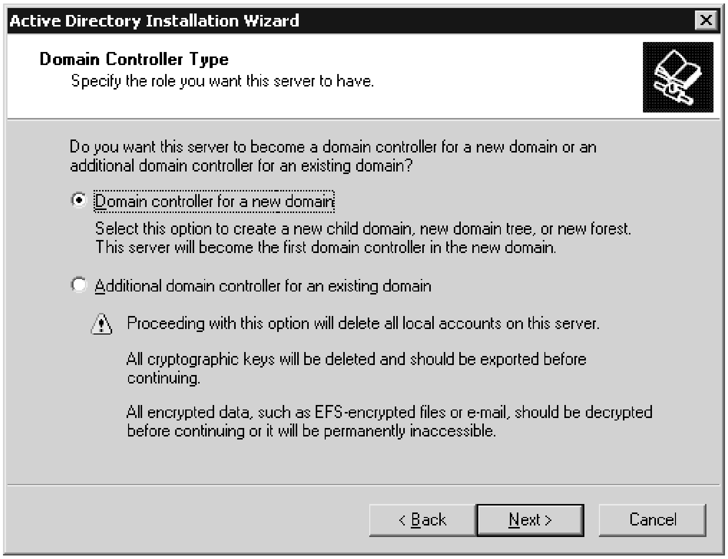
TIP

In some cases, adding a new domain controller with no services other than the NT domain functions is advisable to make the upgrade as smooth as possible. This domain controller can then be upgraded to assume the role of PDC in the legacy domain. With this migration complete, the existing servers can be migrated in any manner, including reformatting and reloading the hard drive, that is advisable. Some companies, such as Hewlett-Packard, Compaq, and IBM, have automated Windows 2000 (and probably Windows Server 2003 shortly) installation CDs for many of their servers, which will automatically load the correct drivers for their system's hardware. The extra domain controller can be removed from the network at a later time or can be used to replace one of the exiting domain controllers.

To install AD on Windows Server 2003, use the AD Installation Wizard, which appears automatically if you are upgrading a domain controller, or it can be started in the Manage Your Server Wizard (see "Set Up a Domain Controller" in Module 4) when you choose to make the server a domain controller and install AD. In the AD Installation Wizard, you are asked if you want to create a new domain or add a domain controller to an existing domain, as shown in Figure 9-2.

CAUTION

As you can read in Figure 9-2, if you are up grading a computer as a domain controller in an existing domain, all existing local accounts, including all cryptographic keys, will be deleted. All cryptographic keys should be exported to another computer, and all encrypted files and e-mail should be decrypted before going ahead with the installation of AD. Also, note that computers running Windows 95 or Windows NT 4 SP3 and earlier will no longer be able to log on to a Windows Server 2003 domain controller or access domain resources unless you install the Active Directory client for those systems.



Domain Controller Type

Specify the role you want this server to have.

Do you want this server to become a domain controller for a new domain or an additional domain controller for an existing domain?

● Domain controller for a new domain

Select this option to create a new child domain, new domain tree, or new forest. This server will become the first domain controller in the new domain.

○ Additional domain controller for an existing domain

⚠ Proceeding with this option will delete all local accounts on this server.

All cryptographic keys will be deleted and should be exported before continuing.

All encrypted data, such as EFS-encrypted files or e-mail, should be decrypted before continuing or it will be permanently inaccessible.

Figure 9-2 Choosing between creating a new domain and adding a domain controller to an existing one

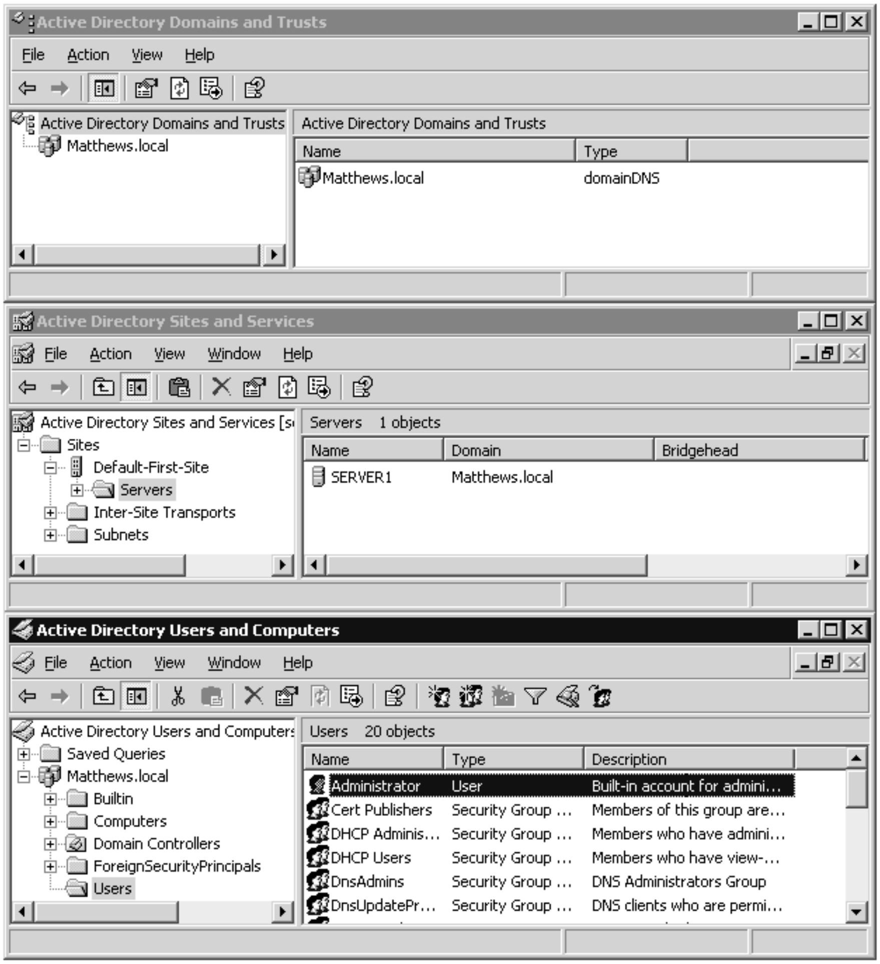
NOTE

Active Directory clients for Windows 95, Windows 98, and early versions on Windows NT 4 are available for download on the Microsoft web site on pages for the various operating systems.

When installing into an existing domain, your administrative rights on the domain will be checked. It is possible to have administrative rights on the server being upgraded but to not have sufficient rights on the domain and domain controllers to do the installation. After you install the AD service, you do the remaining configuration through the AD Domains And Trusts, AD Sites And Services, and AD Users And Computers windows, which Microsoft calls “snap-ins for the Microsoft Management Console (MMC),” as shown in Figure 9-3.

Replace Existing Domain Controllers

Windows Server 2003 servers functioning in native domains (domains that contain only Windows Server 2003 domain controllers) act as peers with all members containing the AD services database with equal read/write privileges. In legacy NT domains, only the PDC contains the master, read/write copy of the domain’s directory store. All the other domain controllers in NT domains, referred to as *backup domain controllers (BDCs),* contain read-only copies of the domain directory information store. BDCs are able to authenticate users and provide domain information, but all additions and modifications to the existing data have to be made to the PDC. When installed in


Active Directory Domains and Trusts
File Action View Help
Active Directory Domains and Trusts
Matthews.local
Active Directory Domains and Trusts

Name	Type
Matthews.local	domainDNS

Active Directory Sites and Services
File Action View Window Help
Active Directory Sites and Services [s
Sites
Default-First-Site
Servers
Inter-Site Transports
Subnets
Servers 1 objects

Name	Domain	Bridgehead
SERVER1	Matthews.local	

Active Directory Users and Computers
File Action View Window Help
Active Directory Users and Computers
Saved Queries
Matthews.local
Builtin
Computers
Domain Controllers
ForeignSecurityPrincipals
Users
Users 20 objects

Name	Type	Description
Administrator	User	Built-in account for admini...
Cert Publishers	Security Group ...	Members of this group are...
DHCP Adminis...	Security Group ...	Members who have admini...
DHCP Users	Security Group ...	Members who have view-...
DnsAdmins	Security Group ...	DNS Administrators Group
DnsUpdatePr...	Security Group ...	DNS clients who are permi...

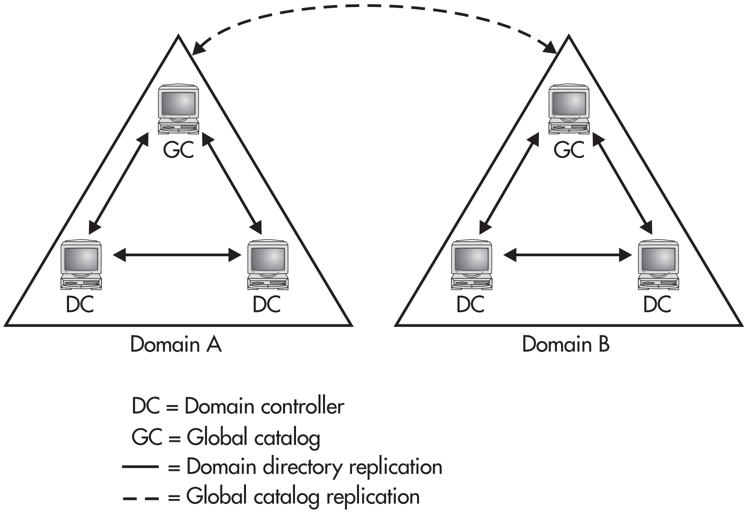
Figure 9-3 Configuring AD in one of three MMC snap-ins

mixed mode, an AD server will still respond to remote procedure calls (RPCs) as if it were a PDC and then replicate directory changes to the legacy BDCs. This means that all remaining legacy domain controllers do not recognize that there has been any major modifications to the network. In Windows 2000 domains, all domain controllers act as peers with all members containing the AD services database with equal read/write privileges similar to Windows Server 2003, but if you mix Windows 2000 and Windows Server 2003 domain controllers in the same forest, they will operate as if they are all Windows 2000 domain controllers and several of the AD replication improvements in Windows Server 2003 will not be available. When all Windows 2000 domain controllers have been upgraded to Windows Server 2003, the functionality level can be advanced to Windows Server 2003, providing access to all the new features.

The method of operation used by AD is that of *multimaster replication,* which means that changes can be made to any AD server and those changes will be replicated to other servers throughout the network. Within this multimaster replication scheme are a couple of important concepts that involve the first AD server installed in the domain, which is automatically configured to be both a global catalog server and an operations master.

Global Catalog Server

The global catalog is a new database type that is at the core of the directory services. The *global catalog* contains a listing of the services that can be accessed within the network, and not just the local domain. The global catalog can be kept on multiple domain controllers, but it always has to be installed on at least one, and by default it is always created on the firstAD server installed in a new forest. (The forest concept is explained later in the module.) The configuration of the global catalog and its placement on various servers throughout the network is done through the AD Sites And Services MMC snap-in.



TIP

When operating in mixed mode, all domain controllers can perform user logins independently. However, when operating in native mode, a query to a global catalog server is required (because it determines a user's global group memberships). Therefore, the rules for installing multiple global catalogs within a network are very similar to those for installing additional BDCs to an NT 4 domain. When installing multiple distant sites, having a global catalog server in each site will decrease the network load on WAN links otherwise used to provide user authentication. Additionally, having multiple global catalog servers to distribute the network load of authentication traffic evenly across multiple servers, instead of stacking this load onto a single server to handle, is much more efficient. Of course, the need for additional global catalog servers increases the network bandwidth consumed for directory synchronization.

In general, the global catalog servers within a network provide two main features: logon and querying. When operating a domain in native mode, universal groups are allowed to exist within the forest. Because universal groups can contain users and groups from multiple domains, the universal group cannot exist within an individual domain. Therefore, the global catalog maintains the universal groups within the network as well as each group's memberships. In Windows 2000 domains, this means that whenever operating a network in native mode, the AD server logging a user into the network has to query a global catalog server to determine the universal group memberships that the user may be a part of. In cases in which a user logs in and a global catalog server is not available, the user will be granted access only to the local computer. This avoids any potential security issues in which a global group based in the domain granted a user access to a resource that the universal group excluded the user from gaining access to. In case of an emergency, however, any user account that is a member of the local domain's Domain Admin group will be allowed to log in to the network. In Windows Server 2003, domain controllers can cache universal group memberships that they have looked up on a global catalog server during user logon, so the next time a universal group member logs through a particular domain controller, the membership can be confirmed locally. This both reduces network traffic and provides confirmation when the global catalog server is down.

The second major feature provided by the global catalog, querying, is a little more obvious. A large network may have numerous domains that all exist together. In this case, the global catalog provides a single place for all of the network's users to reference whenever searching for specific resources. The alternative to the global catalog in this instance would be the requirement for each user either to know the exact location of the resources the user wants to use, or to search each domain independently. The querying feature of AD provides another main reason to have multiple global catalog servers throughout a network.

Operations Master

The second main service added to the first AD server within a network is the operations master. Within Windows Server 2003 domains, there is a need to centralize some changes to the directory services. Even though AD is based on the multimaster replication model, and all domain controllers are peers within Windows Server 2003, there are still some changes that can cause too many issues if configured from multiple locations. For this reason, only one Windows Server 2003 domain controller within any forest and within any domain is assigned to become the operations master for that domain or forest. For domain operations, three roles are assigned to the domain's operations master:

- **PDC emulator** Looks like a PDC to legacy network members in a mixed-mode network
- **Relative ID master** Assigns blocks of security IDs to the other domain controllers in a network
- **Infrastructure master** Updates the other domain controllers for changes in usernames and group-to-user references

Ask the Expert

Q: Does every network need the infrastructure master role?

A: No, the infrastructure master role is not needed in all networks. In cases in which all domain controllers in the network contain a copy of the global catalog (especially in cases in which only one server is in the network), the infrastructure master service is not needed. If the infrastructure master service is installed on a server with the global catalog, it will not function.

A forest has only two roles that are assigned to the forest's operations master:

- **Domain naming master** Adds and removes domains in a forest
- **Schema master** Updates the directory schema and replicates that to the other domain controllers in a forest

Progress Check

1. What is Active Directory?
2. What is a Windows Server 2003 directory and in particular a hierarchical directory?
3. Can computers running Windows 95 or Windows NT 4 SP3 or earlier log on to a Windows Server 2003 domain controller?

1. Active Directory provides a single reference, called a directory service, to all the objects in a network, including users, groups, computers, printers, policies, and permissions. AD also provides a single hierarchical view from which to access and manage all of the network's resources. AD utilizes Internet protocols and standards and requires one or more domains in which to operate.

2. In Windows Server 2003 networking, a directory is a listing of the objects within a network. A hierarchical directory has a structure with a top-to-bottom configuration that allows for the logical grouping of objects, such that lower-level objects are logically grouped and contained in higher-level objects for as many levels as you want.

3. Computers running Windows 95 or Windows NT 4 SP3 or earlier will no longer be able to log on to a Windows Server 2003 domain controller or access domain resources unless you install the Active Directory client for those operating systems.

CRITICAL SKILL 9.3

Understand Active Directory Structure and Configuration

An AD network contains a number of objects in a fairly complex structure and can be configured in several ways. This section focuses on some of the main objects in Active Directory, as well as some of the basic configuration involved with each of these objects as follows:

- AD objects and what they do
- Domains, trees, forests, and the other OUs within AD
- Sites and site-based replication

Active Directory Objects

An *object* within AD is a set of attributes (name, address, and ID) that represents something concrete, such as a user, printer, or application. Like DNS, AD groups and lists these objects in OUs, which are then grouped into other OUs until you reach the root. Also, like DNS, AD then provides access to and information about each of these different objects. As a directory service, AD maintains a list of all objects within the domain and provides access to these objects either directly or through redirection. The focus of this section is the foundation and structure of the objects in AD. By recalling the differences between AD and DNS and the objects that they contain, you can appreciate the wide variety of objects allowed in directory services. In the case of AD and the other X.500-based directory services, the creation and use of this variety of objects is governed by the schema used in the directory.

Schema

The *schema* defines the information stored and subsequently provided by AD for each of its objects. Whenever an object is created in AD, the object is assigned a globally unique identifier, or GUID, which is a hexadecimal number unique to the object. A GUID allows an object's name to be changed without affecting the security and permissions assigned to the object, because the GUID is still the same. Once the object is created, AD uses the schema to create the fields defined for the object, such as phone number, owner, address, description, and so forth. The information for each of these fields is supplied by the administrator or a third-party application pulling the information from a database or preexisting directory structure, such as Microsoft Exchange.

NOTE

For reverse-compatibility reasons, support for security identifiers (SIDs), NT 4's version of GUIDs, is also maintained within AD.

Add to the Base Schema

Because the schema provides the rules for all objects within AD, you'll eventually need to extend the default object classes that come standard within AD. One of the first times this will happen is when a mail server is added to the network, which will require that mail-specific attributes be added to the AD schema. This addition may also require that new object classes be created. The process of adding classes and attributes is done by modifying or adding to the AD schema. The modification of the schema takes place using an automated installation function that affects the schema for the entire network. Schema updates cannot be reversed, so always exercise caution when updating the schema.

TIP

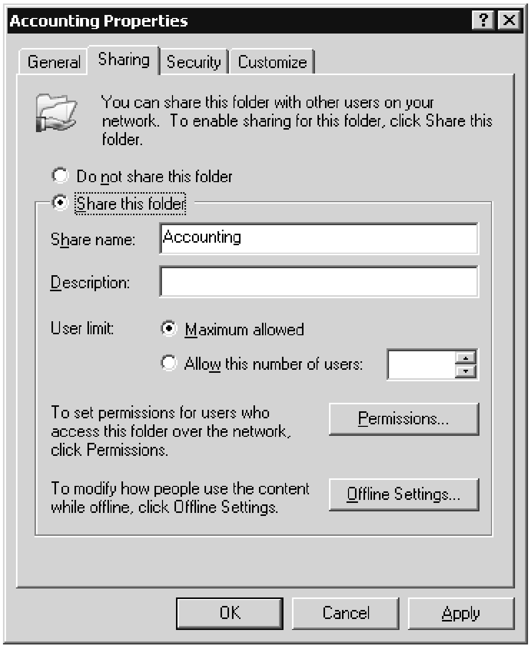
It is possible to modify the schema through the Active Directory Service Interface (ADSI) and by using the LDAP Data Interchange Format utility. It is also possible to modify the schema directly by using the AD Schema tool. These programs should be used only by AD experts and should always be tested in a lab environment before being implemented on production servers.

Publish Items to the Directory

At face value, Windows Server 2003 functions very much like legacy NT-based products, especially in its sharing and security functions. When an item is shared or a new resource is added to Windows Server 2003, such as a printer, the object is shared and secured using nearly the same process as with legacy NT servers. Additionally, not all Windows NT–based servers may become AD servers. With these two facts in mind, some method is needed to distribute information about objects hosted on Windows Server 2003 servers throughout the network into AD. This process is called *publishing*. When an object is published in AD, information about the resource is added to AD, and users are then able to access that resource through AD. The main benefit provided by publishing is that it allows large networks with resources hosted by servers throughout their various sites to all share their information from a central point on the network. Users within the network can access the AD servers and search for, locate, and access the resources they need through the single entry point of AD.

Some objects are added to AD automatically, such as user, group, and server objects. However, some objects have to be published in AD by an administrator. The two most common items that most administrators will find themselves adding to AD are directories and printers.

Directory Publishing To add a directory to AD, the directory or folder must first be created in Windows Explorer or My Computer and secured using the Sharing and Security tabs in the directory's properties dialog box, as shown next. This security has to include both share-level security and the NTFS permissions associated with the directory and all of the files and subdirectories within that share.



You can share this folder with other users on your network. To enable sharing for this folder, click Share this folder.

○ Do not share this folder

● Share this folder

Share name: Accounting

Description:

User limit: ● Maximum allowed

○ Allow this number of users:

To set permissions for users who access this folder over the network, click Permissions.

To modify how people use the content while offline, click Offline Settings.

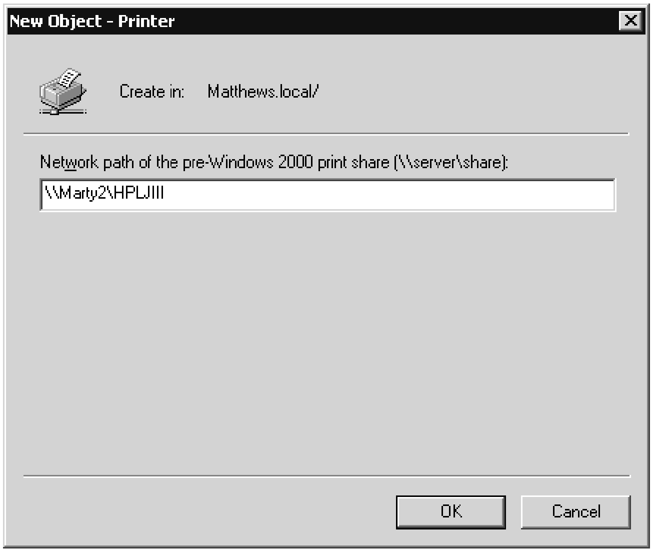
TIP

With the advent of AD, new methods of organizing and providing access to network resources have been added to the network administrator's tools. However, the methods for administering network security that were used with Windows NT are still very much in effect.

Printer Publishing Publishing a printer in AD is a simple process that provides several new features that were not available with previous versions of NT. Primarily, when a number of printers exist within an AD network, users within that network can search for printers according to specific features, such as color, resolution, or duplexing.

To publish a printer in AD, the printer must first be installed and configured to function on the print server in question. (The print server in this case can be any Windows Server 2003 or Windows 2000 Server or any Windows XP or 2000 Professional computer to which a printer is attached.) After a printer is configured, is shared, and has the proper security set (see Module 15 for details on how to do this), it is ready to be published in AD. In fact, the default action taken by a Windows Server 2003/2000/XP print server is to publish any printer automatically after it has been installed and shared. Once published, the printer can then be managed and accessed through all AD servers within the domain, and it is automatically included in the global catalog for the other domains within the forest.

In some cases, especially domains still running in mixed mode, non–Windows Server 2003/2000/XP print servers may be hosting printers that are of significant importance to the domain. In this case, an object can be added to the AD domain by using the URL of the printer share. This can be accomplished using the AD Users And Computers snap-in, selecting the domain, and choosing Action | New | Printer (shown next), or through the use of a Visual Basic script (Pubprn.vbs) included with Windows Server 2003 and located in the System32 folder. Although a little more difficult to set up, using Pubprn.vbs enables you to add numerous printers to AD at one time.



Create in: Matthews.local/

Network path of the pre-Windows 2000 print share (\\server\share):

\\Marty2\HPLJIII

NOTE

Resources shared on the network should not always be published in AD. Only objects used by a large part of the network or that need to be available for possible searches should be published. Publishing objects in AD increases the domain's replication traffic that is necessary to ensure all AD servers have the most recent information. In large networks with multiple domains, the replication traffic to synchronize the global catalog servers for each domain can be overwhelming if poorly planned.

The Structure of Active Directory

There are various OUs within AD that have very specific roles within the network. These roles are established via the schema. To administer and configure an AD network, you need to understand what each of these OUs are, and the role that each plays within the network.

Active Directories are made up of one or more domains. When the first AD server is installed, the initial domain is created. All AD domains map themselves to DNS domains, and DNS servers play a crucial role within any AD domain.

Domains

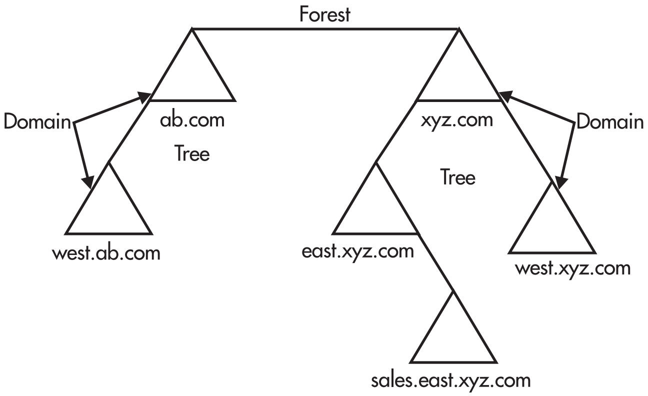
Domains are at the core of all Windows NT/2000/Windows Server 2003–based network operating systems. This section looks at the structure of domains within AD, as well as the various factors involved in creating multiple domains within a network.

Domains in AD delineate a partition within the AD network. The primary reason for creating multiple domains is the need to partition network information. Smaller networks have very little need for more than one domain, even with a network spread across multiple physical sites, since domains can span multiple Windows Server 2003 sites. (The site concept is covered in more detail later in this module.) However, there are still reasons to use multiple domains within a network:

- **Provide network structure** Unlike in legacy NT domains, there is no real limit to the number of objects that can be added to an AD domain running in native mode. For this reason, most networks do not need to establish separate domains for each business unit. However, in some very large networks, various political factors may necessitate multiple domains. For example, one company may own several subsidiary companies that are completely autonomous from the parent company. A central AD shared between the companies provides numerous benefits; a shared domain may not make as much sense. In this case, a separate domain can be set up for each company.
- **Replication** AD servers contain information only about their own domain. Global catalog servers are required to publish information between domains for user access. This means that all objects within a domain are replicated to all the other domain controllers, but external resources are replicated only between global catalog servers. In large networks spread across multiple WAN links, each physical site should be its own domain, to ensure that unnecessary replication traffic does not consume the limited bandwidth of the WAN links.
- **Security and administration** Although AD provides the appearance of a central network infrastructure to the users of the network, administrative abilities and user permissions will not cross domain partitions. This limitation is overcome through the use of global and universal groups and trust relationships, but domains are truly separate administrative groups that may or may not be linked together.
- **Delegation of administration** Although the delegation of administrative authority throughout the network makes multiple domains easy to handle, and Windows Server 2003 provides a number of administrative tools, there still may be benefits for some networks in splitting domains along the lines of administrative authority and responsibility.

Forests

Besides domains, AD is composed of forests, trees, and other custom OUs. Each of these OUs exists on a specific level of AD's hierarchy, beginning with the uppermost container, forests. A *forest* is the highest OU within the network and can contain any number of trees and domains. All domains within a forest share the same schema and global catalog. In essence, forests are similar to DNS's root container. A vast majority of organizations implementing AD will have only one forest; in fact, smaller organizations that have only one domain may not even realize the existence of the forest, because all functions appear to exist on the domain level only. In effect, the forest is used as the main directory for the entire network. The forest encompasses all the trees, domains, and other OUs, as well as all the published information for all the objects in the forest, as you can see here:



Domains within a forest are automatically configured with two-way transitive trusts. A *trust relationship* allows two domains to share user and group resources so that users authenticated by a "trusted" domain can access resources on the "trusting" domain. *Transitive trusts,* shown in Figure 9-4, allow user accounts within a domain to use a second domain's trust relationships to access resources in a third domain. Legacy NT domains did not support transitive trust relationships. In Figure 9-4, a user account from Domain A is able to access resources in Domain C because both Domain A and Domain C have established transitive trust relationships with Domain B. If Figure 9-4 were illustrating a legacy NT domain, a trust relationship would have to be directly established between Domains A and C before the resources could be accessed.

Creating additional forests in a network should be undertaken with great care. The additional forests can cause a tremendous amount of administrative overhead, especially when adding AD-aware messaging platforms, such as Exchange. Other than the obvious political issues, there are very few reasons for any one network to have multiple forests.

Trees

Within AD, trees are used more for administrative grouping and namespace issues than anything else. Basically, a *tree* is a collection of domains that share a contiguous namespace and form a hierarchical environment. For example, it is possible for an organization such as Microsoft to split its DNS structure so that the Microsoft.com domain name is not the primary name used in e-mails and resource references. For example, assume that Microsoft is split geographically first, so that there is a West Coast OU and an East Coast OU within the Microsoft domain, and that

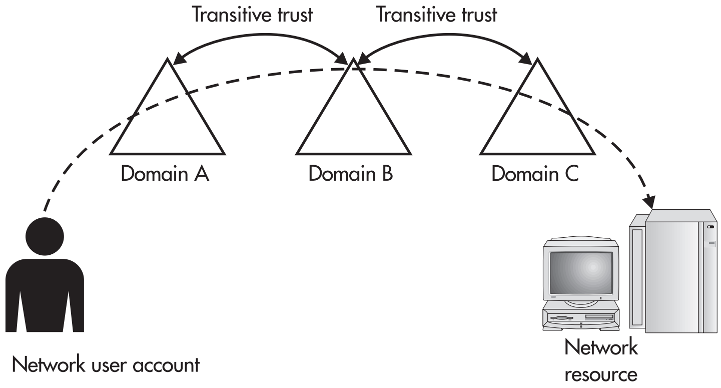


Figure 9-4 Transitive trust relationships allow Domain A to access resources in Domain C.

these OUs (or child domains) each contain a tree of further divisions, such as Sales and Tech Support, which could be further split into Operating Systems and Applications. In this example, so far, Microsoft would have two trees within its DNS structure, West Coast and East Coast. Both of these domains are then split further still, creating a potential FQDN of a server within the technical support department as follows:

```
ServerName.OperatingSystems.TechSupport.WestCoast.Microsoft.Com
```

The entire discussion of trees within AD so far has focused entirely on DNS, since AD has to match the DNS domain structure in the network, although the directory services contained by the two services are independent. In cases in which an organization has decided to implement multiple domains, and those multiple domains exist within a contiguous DNS naming scheme, as previously outlined, the network will have formed a tree connecting the multiple domains. Because the DNS requirements to host such a domain are very large, most organizations implementing trees host the DNS services for the tree internal to the network only, and maintain a separate DNS substructure for Internet hosting services.

All domains within a tree are linked by two-way transitive trust relationships, although the domains are independent. Domains within a tree, just like domains within a forest, share a trust relationship and a namespace (in the case of a tree, it's a contiguous namespace), but the independence and the integrity of each domain remain unchanged. Administration of each domain, as well as directory replication of each domain, is conducted independently, and all information shared between the domains is done through the trust relationships and the global catalog.

When there are multiple trees within a forest, it is possible for each tree to maintain its own independent naming scheme. Looking at the Microsoft example, if you move up the DNS hierarchy, the Microsoft forest may very well include both MSN.com and Microsoft.com. In this case, there is no clear upper layer to the AD domain structure.

The first domain created in the forest is called the *forest root domain.* Two default groups exist within the entire forest, and they exist within the forest root domain:

- Enterprise Administrators
- Schema Administrators

Additionally, the root domain for each tree automatically establishes a transitive trust relationship with the forest's root domain. This relationship is highlighted in Figure 9-5, which is based on the hypothetical Microsoft AD structure. In this example, a third domain is added, MSNBC.com, to further highlight the transitive trusts that are formed throughout the network. Microsoft.com is the forest root domain in this example.

TIP

Domains cannot be moved between forests and cannot be removed if they contain child domains underneath them. Therefore, it is best to plan the entire AD network from top to bottom, before the first AD is installed. A little time spent planning can save a lot of time improvising.

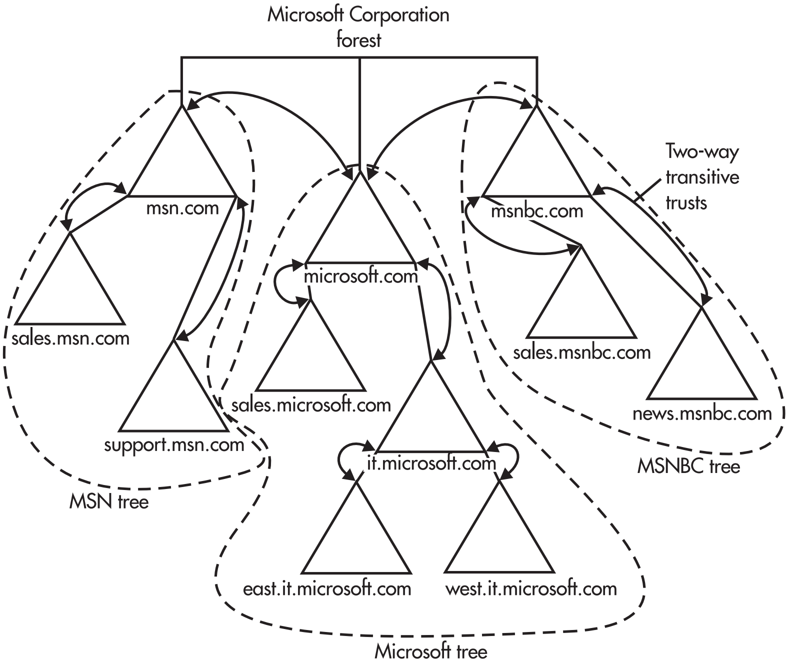


Figure 9-5 Transitive trust relationships among trees in a hypothetical Microsoft forest

Shortcut Trusts

The transitive trust relationships that exist between domains in AD forests can be a very tricky subject. Specifically, the overhead and time involved using the transitive trust relationships to pass all authentication requests throughout a network can sometimes be tremendous. Looking back to the Microsoft example and Figure 9-5, assume that the users who exist in the Microsoft IT group are always logging in to the News.MSNBC.com domain to fix some problems. This means that all login traffic is first passed up the MSNBC.com tree, and then down the Microsoft.com tree, before being authenticated by the IT domain. In cases like these, a *shortcut trust* can be created to pass the information directly between the two domains in question. Shortcut trusts are one-way transitive trusts that are used for authentication in large networks with diverse tree structures. A modified version of the forest from Figure 9-5 is shown in Figure 9-6, with the necessary shortcut trust added.

Other OUs

OUs are simply containers in which multiple objects and additional OUs can be stored. Within AD, some of these OUs are predefined and serve specific roles within the network, such as creating domains. Other OUs revolve around the needs and interests of the administrator, and not those of AD. Within AD, administrators have the ability to create their own AD structure below the OUs that are predefined or serve a specific role. These other OUs can be used to group users, printers, or servers together for ease of administration.

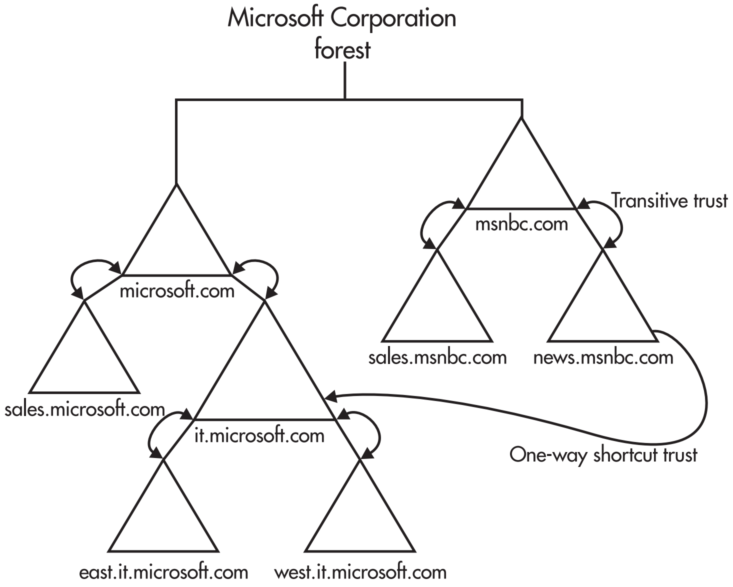


Figure 9-6 A shortcut trust short-circuits trust relationships between domains.

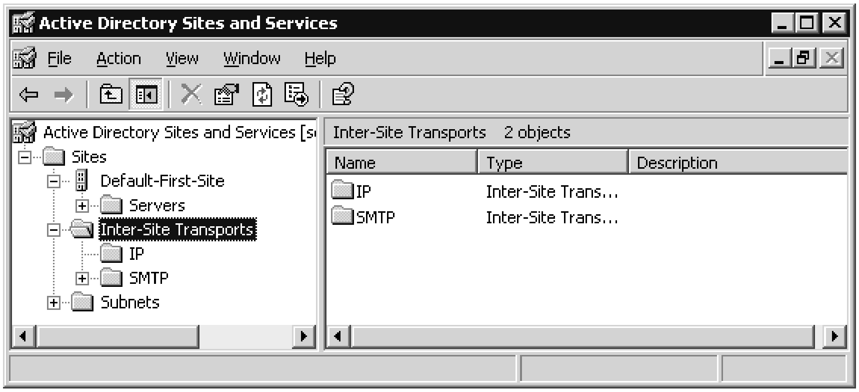
There aren't a lot of rules on the creation of OUs within AD, so instead you must rely on general guidelines and practical uses. Initially, recall that users have the ability to search AD for the resources they need to use. This searching function allows users to specify the specific resource or the type of resource they want to locate, so this should be considered the primary means by which most users will operate with AD. Therefore, administrative needs rather than user needs can determine the creation of OUs. Group policies can be set in OUs that allow the administrator to customize the desktop and permissions of the users and resources within that container. Likewise, giving permission to an entire OU allows an administrator to easily delegate to others administrative rights to specific objects within the domain without compromising the security of the overall network.

Sites

Domains are at the root of the directory services in Windows Server 2003. Everything within AD is simply a collection of one or more domains. Even with the fundamental role of domains defined, many issues still need to be addressed. For example, how is interdomain synchronization handled within small and large domains? How does a large domain span multiple physical networks? How is interdomain replication traffic controlled? For the answers to all of these questions, a new concept needs to be introduced. Windows Server 2003 domains can be divided into units called *sites.* Sites can be used to regulate replication traffic across slower WAN links, and can use various connection methods and replication schedules to ensure a minimal network overhead bandwidth. In general, sites provide a number of basic services for Windows Server 2003 networks, including:

- Minimizing the bandwidth used for intersite replication
- Directing clients to domain controllers in the same site, where possible
- Minimizing replication latency for intrasite replication
- Allowing the scheduling of intersite replication

In general, sites are not related to domains, even though they provide a solution to many of the issues faced by domains previously listed. Sites map to the physical layout of the network, such as offices, floors, and buildings, whereas a domain's layout can be affected by everything from the physical structure to the political structure to the administrative needs of a network. Organizations should split a domain into multiple sites whenever the network will span a connection that is less than LAN speed, which has been 10 Mbps, but increasingly is 100 Mbps. Because most WAN connections will be routed, a separate IP subnet should exist on both sides of the WAN connection. For this reason, Microsoft has mapped most site discrimination to IP subnets. Officially, a site is a collection of "well-connected" (meaning LAN speed or better) Windows Server 2003 systems on the same IP subnet. Sites are managed and created using the AD Sites And Services console.


Active Directory Sites and Services
File Action View Window Help

Active Directory Sites and Services [s
- Sites
  - Default-First-Site
    - Servers
  - Inter-Site Transports
    - IP
    - SMTP
  - Subnets

Inter-Site Transports 2 objects

Name	Type	Description
IP	Inter-Site Trans...	
SMTP	Inter-Site Trans...	

CRITICAL SKILL 9.4

Replicate Active Directory among Sites

In AD, *replication* means the copying of directory information among domain controllers so that they all have the same information and any of the domain controllers can be queried with the same results. Within an AD domain, four main categories of information require replication: configuration, schema, domain, and global catalog information. Each of these categories is stored in separate directory partitions. These partitions are what each AD server replicates and are used by different servers throughout the forest depending on their role within the network. The following three partitions are held by all the AD servers in a forest:

- **Configuration data partition** Holds information stored and used by AD-aware applications and is replicated across all domains in the forest.
- **Schema data partition** Holds the definitions of the different types of objects, as well as their allowable attributes, and is replicated across all domains in the forest.
- **Domain data partition** Holds information unique to the domain in which the server resides. It contains all the objects in the directory for the domain and is replicated only within the domain. The data in this partition is not replicated between domains and will differ greatly from domain to domain.

The fourth type of partition is used by global catalog servers to allow directory information to be shared between domains:

- **Global domain data partition** Holds information about all the objects in the global catalog but includes details on only a few of the objects' attributes, to allow quick searches for and access to resources in external domains, and to reduce the bandwidth used in replications. This information is replicated to all the other global catalog servers within the network. When a client in a foreign domain actually needs access to the resources or the nonreplicated attributes of a resource, the client is directed to that resource's native domain.

NOTE

Most replication among domain controllers is done over the network. This can be a problem if it is a low-bandwidth network or if a number of domain controllers are added rapidly. A new feature in Windows Server 2003 allows the Active Directory database files to be backed up to tape, CD, DVD, or a removable hard disk and used as the source of an initial replication on a new domain controller or global catalog server.

Internal Site Replication

There will always be at least one site within every AD implementation. When the first Windows Server 2003 domain controller is installed, it creates a site called the Default-First-Site-Name. The new domain controller then adds itself to that site. Whenever new domain controllers are added to the network, they are automatically added to this new site first; they can be moved at a later time. There is, however, an exception to this statement. When a new site is created, one or more IP subnets are assigned to that site. After there are two or more sites, all new domain controllers added to the forest will have their IP addresses checked and will be added to the site with a matching IP address.

Directory information within a site is replicated automatically to ensure all domain controllers within the site have the same information. Additionally, all intersite replication occurs in an uncompressed format, which consumes more network traffic but less system resources. For these reasons, a site should always be a LAN network of high-speed connections (10MB or higher). When multiple domains exist within a single site, replication occurs only among domain controllers in a given domain. However, replication traffic between global catalog servers occurs across the site on an as-needed/uncompressed basis, as well. Figure 9-7 shows the single-site replication traffic between two domains.

Site-to-Site Replication

When deciding to split a network into multiple sites, a number of new issues arise. These issues, and the configuration needed to make multiple sites work, are the subject of the remainder of this section.

Two main rules should be adhered to whenever planning a multisite design, to minimize the potential problems:

- Sites always should be split by geographic regions. When two or more LANs are connected by WAN links, each LAN should represent its own site.
- Whenever possible, each site should have its own AD and global catalog server. In some smaller networks, this server may be the only server in the site, in which case it should also serve as the DNS and DHCP server. This will increase fault tolerance as well as client performance, while decreasing WAN bandwidth utilization.

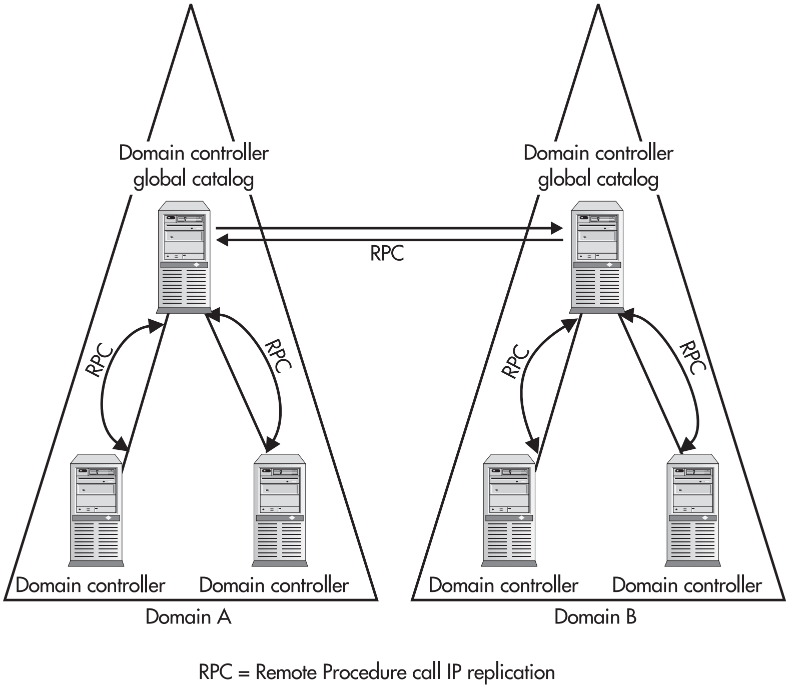


Figure 9-7 Replication traffic between two domains within the same site

Site Connectivity

Sites are connected using *site links,* which are domain controllers configured to serve as a connection to a particular site. Site links have to be manually configured using the AD Sites And Services console. When creating a site link, the administrator has a number of configuration options.

The first of these configuration options is the replication schedule. An administrator can determine when replication should occur across a particular site link. When setting the replication schedule, an administrator will set the replication cost, replication availability, and replication frequency associated with the site link. When the site link is created and configured, AD automatically creates a connection object, which will use the information provided by the site link to actually transfer the information between the domains.

By default and design, all site links are transitive, which allows sites to be used in the same manner transitive trusts are used. This means that a site can replicate its changes to another site, via a site connector to a third site, as long as the third site has a valid connection to both sites. In some organizations in which multiple field offices all connect to central corporate offices, this kind of replication allows for the greatest efficiency. All sites replicate their information to the central office site, which in turn replicates the information back out to the other field offices.

Protocols

There are two protocol options within any network for connecting sites together. Both options, though, are protocols within the TCP/IP protocol stack. There is no way to connect two sites using anything other than IP.

- **RPC (IP replication)** Remote Procedure Call (RPC) is an IP-based, connection-oriented protocol that is at the base of legacy Exchange installations. RPC is fast and reliable when used on connection-friendly networks (networks that allow packets to travel the same path and arrive in sequence at the destination). However, RPC is less than reliable in large mesh networks, such as the Internet. RPC communication is still the default replication method for servers within the same site as well as intrasite communication for other Microsoft services such as Exchange.
- **SMTP** A feature that was introduced with Windows 2000 and Exchange 2000 is the ability to connect sites using SMTP connectors. SMTP is just beginning to be used and cannot be used for replication between servers in the same site (although it is the default for Exchange 2000). SMTP is not capable of replicating the domain partition, and therefore it is suitable only for linking two sites that are also separate domains. SMTP can pass the schema, configuration, and global catalog partitions very efficiently.

NOTE

SMTP is, by nature, very insecure. All SMTP messages are sent in formats that can be easily interpreted by the most basic of sniffing tools. For this reason, using the SMTP protocol to connect two sites requires that an enterprise certification authority be installed and configured on the network. This allows for all SMTP traffic generated by AD to be encrypted using at least 56-bit encryption.

Collision Detecting and Resolution

What happens when the same object is modified from two different spots in the network at the same time, or two objects with the same name are created at the same time within the network? Even in the most basic networks, if two domain controllers exist, the two directories that exist on each domain controller will not be exactly the same at all times, because replication takes time. Legacy versions of NT dealt with this issue by allowing only one read/write copy of the directory to exist on the network at any one time. This meant that all changes could be made only by one server, and those changes were then propagated to the read-only copies of the directories that the remaining domain controllers maintained.

When a change to an object occurs before a previous change to that object has been completely replicated, a replication *collision* occurs. AD can track the versions of objects by looking at each object's version number. When an object is changed, its version number is increased, so that when a server receives an update, it can compare the version number of the incoming object with the version number of the existing object. When the existing object's version number is less than the version number coming in, the replication continues and all is considered well. A collision occurs when the version number of the existing item is equal to or greater than the version number of the incoming item. In this case, AD compares the timestamp of the incoming object to the timestamp of the existing object to determine which one it's going to keep. This is the only instance in which time is used in replication. If the timestamps don't settle the issue, the item with the highest GUID is kept. In situations in which the incoming version number is actually lower than the existing version number, the replication object is considered to be stale and is discarded.

Active Directory Summary

Active Directory is one of most important features of Windows Server 2003. AD is a directory service based on the X.500 directory scheme that contains a variety of predefined and preconfigured objects and OUs. A large part of any Windows Server 2003 administrator's job will be the administration and configuration of the special objects and OUs within the AD forest. Additionally, the namespace of AD has to match that of a DNS domain, and in cases in which the AD domain and the Internet DNS domain for the company don't match, separate DNS domains have to be maintained, because AD requires DNS to allow for client connectivity.

If used correctly, AD can add tremendous value and reliability to a network, as well as decrease the administrative overhead involved in the network's daily maintenance. However, if configured incorrectly or if ill-planned, AD can drastically increase the administrator's workload, and decrease the network customer's satisfaction. When dealing with AD, a little bit of forethought and planning can save a huge amount of work and heartache.

Progress Check

1. What is an object within AD?

2. What are GUIDs and SIDs and how are they used in AD?

3. After a printer is configured, shared, and has the proper security set, what action must be taken to publish it in AD?

1. An object within AD is a set of attributes (name, address, and ID) that represents something concrete, such as a user, printer, or application. Like DNS, AD groups and lists these objects in OUs, which are then grouped into other OUs until you reach the root.

2. A GUID is a globally unique identifier, and a SID is a security identifier. A GUID is given to each AD object to identify it independent of the object's name, so the name can be changed. For reverse-compatibility reasons, support for SIDs, NT 4's version of GUIDs, is also maintained within AD.

3. None; after a printer is configured, shared, and has the proper security set, the default action taken by a Windows Server 2003/2000/XP print server is to publish any printer automatically.

Project 9 Plan an Active Directory Implementation

Active Directory is a very powerful and important feature in Windows Server 2003, but it is also reasonably complex with many nuances that require some consideration before it is implemented. This project will help plan an AD implementation.

Step by Step

1. Review the current or planned network to which AD will be added and determine the answers to the following questions:
   a. Will there be more than one forest? If so, determine
      i. What will constitute the forest boundaries?
      ii. How many forests will that make?
      iii. What will the forests be named?
      iv. How will the forests be managed?
   b. Will there be more than one tree in each forest? If so, determine
      i. What will constitute the tree boundaries?
      ii. How many trees will that make?
      iii. What will the trees be named?
      iv. How will the trees be managed?
   c. Will there be more than one domain in each tree? If so, determine
      i. What will constitute the domain boundaries?
      ii. How many domains will that make?
      iii. What will the domains be named?
      iv. How will the domains be managed?
   d. Will there be more than one OU in each domain? If so, determine
      i. What will constitute the OU boundaries?
      ii. How many OUs will that make?
      iii. What will the OUs be named?
      iv. How will the OUs be managed?
   e. Will there be subsidiary OUs in each OU? If so, determine
      i. What will constitute the subsidiary OU boundaries?
      ii. How many subsidiary OUs will that make?

        iii. What will the subsidiary OUs be named?

        iv. How will the subsidiary OUs be managed?

    f. Repeat Step e for each additional level of OU in the organization.

2. In each of the bottom level OUs, determine

    a. The number of servers

        i. What is the function of each server?

        ii. What is the name of each server?

        iii. Who is responsible for each server?

    b. The number of storage devices

        i. What is the name of each storage device?

        ii. Who is responsible for each storage device?

    c. The number of printers

        i. What is the name of each printer?

        ii. Who is responsible for each printer?

    d. The number of workstations

        i. What is the name of each workstation (if known)?

        ii. Who is responsible for each workstation?

3. For the network structure you have just laid out, determine the answers to the following questions:

    a. How well will network traffic be handled?

    b. Where will the global catalog reside?

    c. How will the network structure map to physical sites?

    d. Are any one-way shortcut trusts required, and if so, where?

    e. What is the replication plan throughout the network?

    f. Are any changes necessary in the decisions you made in earlier steps?

Project Summary

If you have a very simple network, the preceding steps all but answer themselves. With a large, complex network it will be a significant task to answer all the questions. The importance of answering the questions is almost directly proportional to the difficulty of doing that. Planning out your AD lays the foundation of your network. If it is done well, you will reap the benefits for some time.

Module 9 Mastery Check

1. What is a domain?
2. What is the directory structure used within Windows Server 2003, and how is it used?
3. What is an FQDN, and how is it used in Windows Server 2003?
4. What kind of structure does DNS use to relate computer names to IP addresses?
5. What is the information that AD stores for a particular directory entry?
6. Is there a PDC in AD and how are the functions of a PDC handled?
7. What is a schema, and how is it used in AD? What part does a GUID play in the AD schema?
8. How are multiple domains in AD helpful?
9. What is a forest, and how is used in AD?
10. In working with domains, what is a trust relationship, and in particular, what is a transitive trust?
11. What is meant by AD replication, and how is it used?