

EXAM OBJECTIVES

This chapter kicks your preparation for the Directory Services exam into high gear. Although the basics covered in this chapter don't apply to any one specific exam objective, they're the fundamental cornerstones on which you'll build your Active Directory knowledge and skills.

Active Directory is new for Windows 2000 and has its own unique set of terminology and components. In this chapter, you'll master that terminology, as well as basic Active Directory concepts. With the basics firmly under your belt, you'll be ready to meet head-on the more advanced Active Directory topics covered in later chapters.

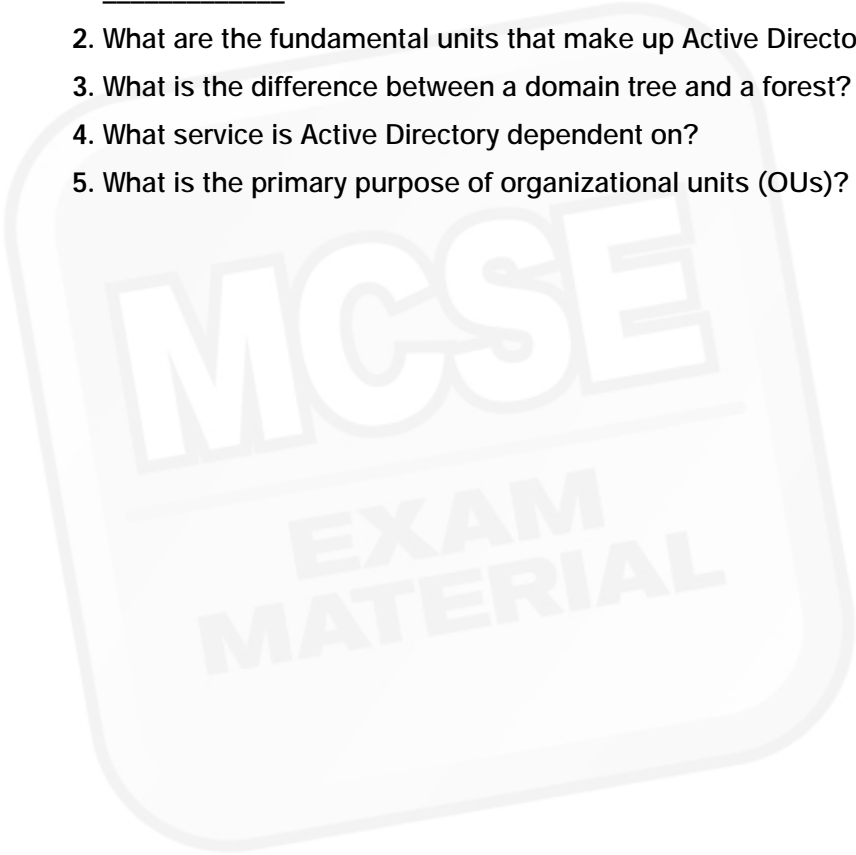
Overview of Active Directory

2

No overview of Windows 2000 would be complete without a discussion of Active Directory. In this chapter, I'll define what Active Directory is and discuss some of its important features. Next I'll explain in detail the structure of Active Directory, including its many components. I'll go over some of the practicalities of how Active Directory is implemented. Finally, I'll introduce some food for thought when planning for Active Directory on your Windows 2000 network.

Chapter Pre-Test

1. Two key features of Active Directory are _____ and _____.
2. What are the fundamental units that make up Active Directory?
3. What is the difference between a domain tree and a forest?
4. What service is Active Directory dependent on?
5. What is the primary purpose of organizational units (OUs)?



MCSE
EXAM
MATERIAL

What Is Active Directory?

Active Directory is the directory service used by Windows 2000. It is a core new feature of the Windows 2000 operating systems.

A *directory service* consists of two parts—a centralized, hierarchical database that contains information about users and resources on a network, and a service that manages the database and enables users of computers on the network to access the database. In Windows 2000, the database is called the Active Directory data store, or sometimes just the directory. The Active Directory data store contains information about various types of network objects, including printers, shared folders, user accounts, groups, and computers. Windows 2000 Server computers that have a copy of the Active Directory data store, and that run Active Directory are called *domain controllers*. In a Windows 2000 domain, a read/write copy of the Active Directory data store is physically located on each domain controller in the domain. A *domain* is a logical grouping of networked computers in which one or more of the computers has shared resources, such as a shared folder or printer, and in which all of the computers share a common Active Directory data store.

The three primary purposes of Active Directory are:

- To provide user logon and authentication services
- To enable administrators to organize and manage user accounts, groups, and network resources
- To enable authorized users to easily locate network resources, regardless of where they are located on the network



CROSS-REFERENCE

This chapter is basically a theoretical and planning discussion about Active Directory. Later chapters in this book address installing Active Directory (Chapter 7), administering and securing Active Directory (Chapter 8), and managing and optimizing Active Directory operations and replication (Chapter 22).

So why is Active Directory so cool? I'll answer that question in the next section by discussing some of the features of Active Directory.

Understanding the Features of Active Directory

Active Directory is a major step forward for the Windows NT/Windows 2000 operating systems. Just a few of the key features and benefits offered by Active Directory are:

- It provides fully integrated security.
- It provides ease of administration by using group policies.
- It makes resources easier to locate.
- It is scalable to any size network.
- It is flexible and extensible.

Because each of these features is fundamental to your understanding of Active Directory, I'll discuss them individually in the following sections.

Fully Integrated Security

When I say that Active Directory provides fully integrated security, I am actually expressing two important concepts. First, Active Directory (working in conjunction with the Windows 2000 Security subsystem, in which Active Directory resides) provides network security by managing the logon and authentication processes. Second, Active Directory (and the Security subsystem) provides security by controlling access to objects (such as user accounts, shared folders, and printers) in the directory data store. What makes this access control so powerful is that it can be defined precisely — not only on each and every object in the directory data store, but on each separate property of each object, as well.

Ease of Administration

The logical, hierarchical structure of Active Directory, in conjunction with group policies, makes for greater ease in administering a Windows 2000 Server network.

You can think of the structure of Active Directory as being like the hierarchical structure of a file system. When working with a file system, you can assign a particular user administrative rights to a folder and to all of that folder's contents. In Active Directory, you can delegate, to a particular

user, administrative rights to a specific part of Active Directory, and to all of that part's contents.

The hierarchical structure of Active Directory also lends itself to the application of group policies. A *group policy* is a policy that contains rules, settings, or both that are applied to all users or computers located in a specific part of Active Directory. For instance, a group policy can be used to define a set of programs that will appear on the desktops of the users whose user accounts are located in a particular part of Active Directory. This part could be a department, a floor in a building, a geographic location, or the entire company. In Active Directory, you can apply a group policy to your entire network, or to the largest unit of your organization to which you want the policy to apply. Administration is easier because you manage settings that apply to many users by implementing a small number of policies, rather than by manually configuring settings individually for a large number of users and computers.

Ease of Locating Resources

Because Active Directory stores information on all network resources in a centralized data store, it stands to reason that it should be easier for a user to locate a resource than if this information were distributed throughout numerous databases on the network. And it really is easier.

Active Directory also enables administrators and users alike to quickly locate an object anywhere on the network by searching for any known property of the object. For example, suppose I want to find the e-mail address of a particular user on my network. I can use the Search menu, My Network Places, or Active Directory Users and Computers to search for this user by the user's first or last name, telephone number, or other known property of that user account. Assuming I have the appropriate permissions to view the user's account information, all information about that user, including the user's e-mail address, will be displayed.

Scalability to Any Size Network

The hierarchical structure of Active Directory lends itself to scalability. Because Active Directory can include multiple domains, it is scalable to any size network.

Flexibility and Extensibility

Active Directory can evolve as your business does. It is not a static structure that, once implemented, can never be changed.

Active Directory is said to be *extensible*. This means that new classes of objects can be added, and new attributes can be added to classes of objects already present.

Now that you have a basic understanding of what Active Directory is and an awareness of some of its key benefits, it's time to wade in a little deeper to the actual structure of Active Directory and its many components.

Understanding the Structure of Active Directory

To review: Active Directory has a hierarchical, tree-like structure. Information about network users and resources is stored in the *Active Directory data store*, which is a structured, centralized database. A read/write copy of the Active Directory data store is physically located on each domain controller in a Windows 2000 domain. This data store is commonly referred to as the directory.



EXAM TIP

If you don't have a solid understanding of the structure of Active Directory and its components, don't even *think* of taking the Directory Services exam! Nailing down these concepts is vitally important to your success on this exam.

In order to talk in greater depth about the structure of Active Directory, I need to introduce and define several new terms. Many of these terms are components of Active Directory, and some of the terms are used to define relationships between the components. In the following sections I'll discuss objects and classes, schema, the global catalog, and the hierarchical structure of Active Directory, including domains, organizational units, trees, trust relationships, and forests. I'll also discuss Active Directory names and naming conventions, as well as security. When I'm finished, you'll have a much better picture of how Active Directory is structured.

Objects and Classes

An Active Directory *object* is a record in the directory that is defined by a distinct set of attributes. The attributes of an object are the same as the object's properties. The terms are synonymous; however, the term *properties* is more prevalent throughout the Windows 2000 user interface.

The specific attributes that an object can have are defined by the object's class. A *class* is simply a template that is used to define the attributes of an object when it is created. A class defines the required and optional attributes of the objects that are instances of that class. For example, the Computer class contains a list of the required and optional attributes that are used when a computer object is created. All computer objects will be created using the same Computer class definition.

There are many classes of Active Directory objects. Some of the classes are:

- Computer
- Contact
- Group
- Organizational Unit
- Domain
- Printer
- User
- Shared Folder

Schema

In Active Directory terminology, the *schema* is a formal definition — a set of rules, if you wish — of all of the classes of objects and their attributes that are stored in the directory. The schema governs the structure of the directory, including how various objects in the directory fit into the directory's hierarchical structure.

The schema is what makes Active Directory extensible. As organizations change, it may be necessary to add or modify object attributes, or even to create new classes. The use of certain applications, in particular, may require these kinds of modifications. Microsoft anticipates that application vendors will provide the means to modify the schema when necessary to support their application's specific requirements.

Windows 2000 Server includes a tool to modify the schema. It is a Microsoft Management Console (MMC) snap-in that is only available after installing the Windows 2000 Administration Tools (ADMINPAK) on a Windows 2000 computer. The name of the snap-in is Active Directory Schema.



CROSS-REFERENCE

For information on Installing the ADMINPAK, see the sidebar titled "Installing the ADMINPAK" in Chapter 8.

Because the schema is the heart of Active Directory, it's important that it be protected from accidental or unauthorized modification. For this reason, Microsoft created a special Security Group for Windows 2000 called Schema Admins. Only users with this permission can run programs that will modify the schema.

Global Catalog

The *global catalog* is a master, searchable index that contains information about every object in every domain in a forest. For now, you can think of a *forest* as all of the domains that make up a company's network. Forests will be covered in more technical detail later in this chapter.

The global catalog, in conjunction with various search tools, is what enables administrators and users to search for and quickly locate an object, regardless of where the object is located on the network.

Windows 2000 automatically creates, by default, a global catalog on the first domain controller that is installed in a forest. You can configure other domain controllers to maintain a copy of the global catalog, as well. The global catalog contains a full copy, or replica, of all objects in its host domain, and a partial replica of all objects in all other domains in the forest. A partial replica includes the most common properties of every object, but not all of the properties of every object.

Hierarchical Structure

By now you've read the term "hierarchical structure" a zillion times. But what does it mean, exactly? A *hierarchical structure* refers to a manner of organizing a group of interrelated elements in which the elements are ranked or stacked, one above the other. An example of a hierarchical

structure that you are probably familiar with is an organizational chart. Figure 2-1 shows an organizational chart for ABC Bank.

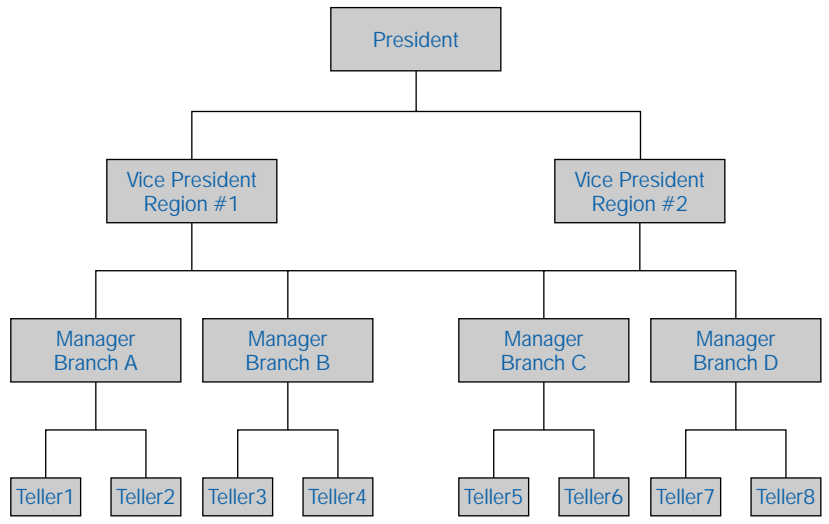


FIGURE 2-1 Organization chart of ABC Bank

In the organizational chart of this regional bank, the President is at the top of the chart, and beneath the President is a level consisting of two Vice Presidents. After the Vice President level is a layer of management staff, and beneath this layer is a level that represents the numerous bank tellers. The hierarchical structure typically has at the top a single element, which branches into lower layers that contain progressively more elements the farther down you go.

The key building blocks in the Active Directory hierarchical structure are domains, which are the focus of the next section.

Domains

Domains are the fundamental units that make up Active Directory. As stated previously, a *domain* is a logical grouping of networked computers in which one or more of the computers has shared resources, such as a shared folder or printer, and in which all of the computers share a common Active Directory data store that contains user account, resource, security, and other information. Active Directory consists of one or more domains.

A domain is a natural security boundary in a Windows 2000 network. Users from other domains cannot pierce this boundary to access shared resources unless trust relationships are created between the domains to

permit user access. More information about trust relationships is provided later in this chapter.

A domain can span several geographic locations of a company, or a domain can be created for every location. Sometimes the needs of the departments, divisions, or subsidiaries of an organization determine the number and structure of the domains needed to effectively manage the organization's network.



TIP

When possible, I recommend using a single domain, because this greatly simplifies the administration of your network.

The domains that make up Active Directory usually correspond to the network's DNS domains, and typically use the same FQDN naming convention used by DNS servers. FQDN stands for *fully qualified domain name*, and is the naming convention used on the Internet. The format for an FQDN is *server_name.domain_name.root_domain_name*. I'll discuss names and naming conventions in a bit more detail later in this chapter.

Domains contain objects, and can also contain organizational units, which are discussed in the next section.

Organizational Units

Organizational units are a type of Active Directory object, and are sometimes called container objects. They contain objects and other organizational units from their own domain. Organizational units are often called by their abbreviated name (OUs).

An organizational unit is used to organize related objects and other organizational units in Active Directory in much the same way that a folder is used to organize related files and other folders in a volume. Also, the organizational unit is the smallest container component of Active Directory to which you can delegate administrative authority or assign group policy. The primary purpose of an organizational unit, then, is the organization of related objects and other organizational units to simplify administration.

For example, suppose an administrator wants to delegate network administration of the Sales department to an assistant administrator. The administrator decides to group together all of the objects associated with the Sales department (including users, computers, printers, shared folders, and groups). Then the administrator creates an organizational unit and

places all of the objects associated with the Sales department into this organizational unit. Completing these steps enables the administrator to delegate administration for the Sales department by assigning the assistant administrator the permissions required to administer the organizational unit and its contents.

Trees

In Active Directory terminology, a *domain tree* is a hierarchical grouping of one or more domains that must have a single root domain, and may have one or more child domains. In a domain tree, the root domain is the domain at the top (or root) of the tree.

Domains in a domain tree are often spoken of in terms of parent domains and child domains. A *parent domain* is any domain that is above another domain in the domain tree hierarchy. A *child domain* is any domain that is below another domain in the tree. A domain can be a parent to a domain below it and a child to the domain above it. In a multidomain tree, the root domain is always a parent domain. Figure 2-2 illustrates a domain tree. Notice that there is only one root domain in the tree, but that the tree contains more than one child domain.

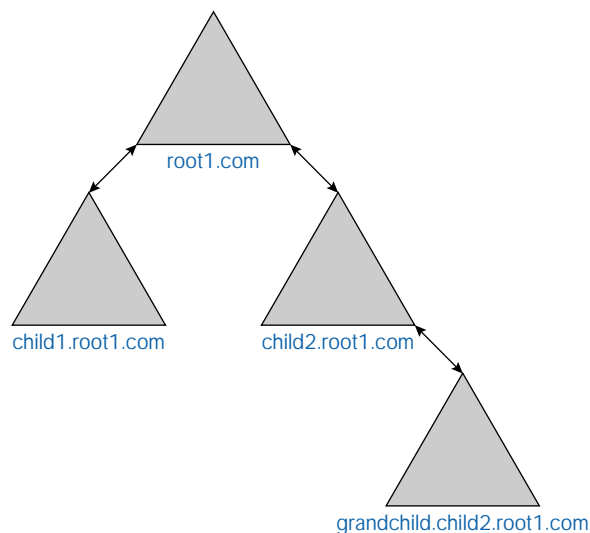


FIGURE 2-2 A domain tree

Also notice the naming structure used in Figure 2-2. In a domain tree, the domains that make up the tree have contiguous DNS domain names. The root domain's name forms the basis of (and will be a part of) the

FQDNs of all of the other domains in the tree. A child domain's FQDN is created by appending the name of the parent domain to its own NetBIOS name by using the *child_domain.parent_domain.root_domain.com* format. For example, in Figure 2-2, the domain with a NetBIOS name of child1 appends the name of its parent domain, root1.com., to its own name, resulting in an FQDN of child1.root1.com. The root domain in a domain tree also takes its name in this way, by appending the name of the first-level DNS domain that it is a member of to its own NetBIOS name. In Figure 2-2, the root domain with a NetBIOS name of root1 appends the name of the first-level DNS domain, com, to its own name, resulting in an FQDN of root1.com.

In organizations that require multiple domains, a domain tree enables any permitted user in any domain in the tree to access shared resources in any domain in the tree. This user access is made possible by the special trust relationships that exist between the domains in the tree.

Trust Relationships

To manage the interaction between multiple domains, trust relationships are necessary. A *trust relationship*, or *trust*, is an agreement between two domains that enables users in one domain to be authenticated by a domain controller in another domain, and therefore to access shared resources in the other domain.

The terminology used to discuss trusts is sometimes confusing, so a good portion of this section is dedicated to explaining and clarifying these terms. Once you've mastered the terminology, trust concepts are much easier to understand.

Trusting Domain vs. Trusted Domain Two terms are commonly used to refer to a trust between two domains: trusting domain and trusted domain. The *trusting domain* is the domain that has resources to share with user accounts in the trusted domain. The trusting domain trusts the trusted domain. The *trusted domain* is the domain that contains the user accounts that want to access the shared resources in the trusting domain. The trusted domain is trusted by the trusting domain.

A trust relationship between two domains is depicted in diagrams by using an arrow to point from the trusting (resource) domain to the trusted (user accounts) domain. Figure 2-3 illustrates a trust relationship between the west.com domain and the east.com domain. The west.com domain

is the trusting domain, and the `east.com` domain is the trusted domain. Notice that the arrow points toward the domain with the user accounts.

This trust relationship enables users from the `east.com` domain to access shared resources located in the `west.com` domain.

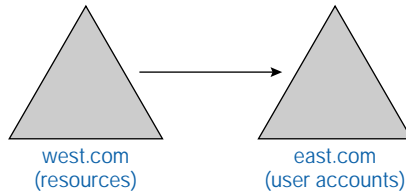


FIGURE 2-3 The `west.com` domain trusts the `east.com` domain

Intransitive and Transitive Trusts An *intransitive trust* is a trust relationship between two domains that does not extend beyond these two domains to other domains. An intransitive trust is a *one-way trust*, meaning that a single trust relationship exists between the two domains.

Suppose that the `a.com` domain trusts the `b.com` domain. Further suppose that the `b.com` domain trusts the `c.com` domain. Figure 2-4 shows these trust relationships.

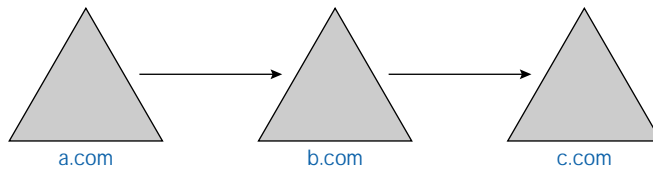


FIGURE 2-4 Intransitive trusts

At first glance, it might appear that the user accounts in the `c.com` domain are able to access resources in the `a.com` domain, but this is not the case. A trust relationship does not exist between the `a.com` domain and the `c.com` domain. Therefore, users in the `c.com` domain can't access resources in the `a.com` domain.

It is possible to establish a two-way trust relationship between two domains by creating two, one-way trusts between those domains. In a *two-way trust relationship*, two domains trust each other.

A *transitive trust* is a trust relationship between two Windows 2000 domains in the same domain tree (or forest) that can extend beyond these two domains to other trusted domains within the same domain tree (or forest). A transitive trust is always a two-way trust, meaning that both of

the domains trust each other. By default, all Windows 2000 trusts within a domain tree (or forest) are transitive trusts.

Transitive trusts are depicted in diagrams by a single line with an arrow at each end. Figure 2-5 illustrates transitive trusts in a Windows 2000 domain tree.

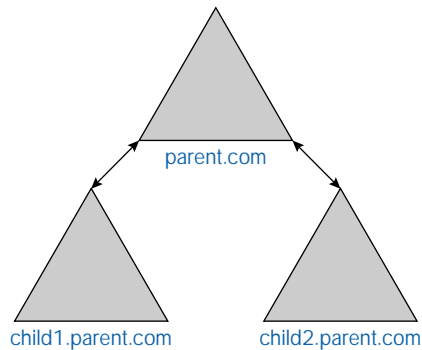


FIGURE 2-5 Transitive trusts

Notice that in Figure 2-5, transitive trust relationships exist between each child domain and the parent domain, but that no trust relationship exists directly between the two child domains. Nonetheless, the transitive trust relationships make it possible for users in the `child1.parent.com` domain to access resources located in both the `parent.com` domain and in the `child2.parent.com` domain. Likewise, users in the `child2.parent.com` domain can access resources located in both the `parent.com` domain and in the `child1.parent.com` domain because of the transitive trusts that connect the three domains.

In Windows 2000 domain trees, Windows 2000 Server automatically creates two-way, transitive trust relationships between a parent domain and a child domain when the child domain is created in the domain tree. The presence of transitive trust relationships between all of the domains in a Windows 2000 domain tree makes it possible for a user in one domain to access a shared resource located in any domain in the tree, regardless of how many domains separate the user and the shared resource.

Windows NT Server 4.0 doesn't support transitive trusts—it only supports intransitive trusts. This means that the only type of trust relationship possible between a Windows 2000 domain and a Windows NT domain is an intransitive trust.

Explicit Trusts An *explicit trust* is a trust that an administrator creates, versus a trust that is automatically created by Windows 2000. An explicit trust can be either transitive or intransitive. Explicit trusts are sometimes used when you need to manage trusts between a Windows 2000 domain and a Windows NT domain. Explicit trusts are also used in large, multidomain forests to shorten the path between two domains to shorten the time required for authentication and logon.

Forests

Earlier in this chapter I said you could think of a forest as being all of the domains that compose a company's network. A more technically accurate definition of a forest is a group of one or more domain trees, linked by transitive trusts, that shares a common schema and global catalog.

A forest begins with one domain and one domain tree. It's kind of a difficult concept to grasp, but when you install Active Directory on the first domain controller on your network, Windows 2000 creates a domain, a domain tree, and a forest all at the same time. So, even though you've only installed Active Directory on one computer, you've got all of these big-picture elements created and ready to go. Now the forest can grow as you add additional domains and domain trees.



IN THE REAL WORLD

There are no tools that enable you to work on two forests at the same time. If you have a multi-domain network, it can be good to have a root domain that only contains the administrator account so that changes to domain structures can be easily made.

Figure 2-6 illustrates a forest that consists of two domain trees. Notice that this forest contains two root domains, each of which forms the basis for its own domain tree. Also notice that a single, transitive trust connects the two domain trees.

Take another look at Figure 2-6, and notice the domain names. By definition, the domains in a domain tree have contiguous DNS domain names. In this example, `rootA.com` is contained in the name of every domain in its tree. Likewise, `rootB.org` is contained in the name of every domain in its tree. However, that's as far as it goes. The two domain trees themselves do not have contiguous DNS domain names, even though they have been joined together in a forest.

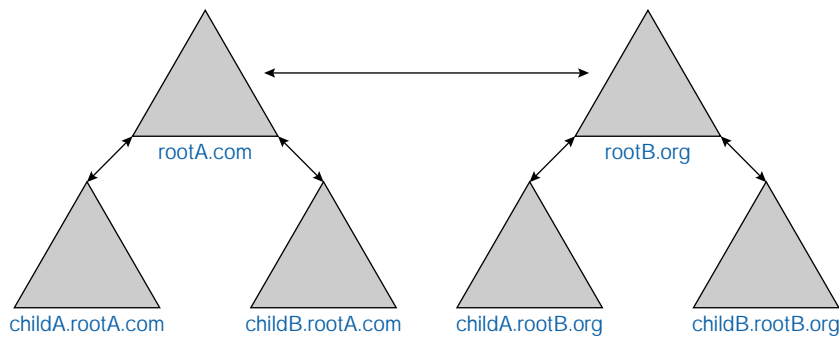


FIGURE 2-6 A forest

A forest takes its name from the root tree, which is the first tree created in the forest.

Names and Naming Conventions

Names are of critical importance in Active Directory. In this section, I'll explain the types of names and naming conventions used by Active Directory.

Within Active Directory, each object has a name. When you create an object in Active Directory, such as a user or a computer, you assign the object a name. This name must be unique within the domain—you can't assign an object the same name as any other object (regardless of its type) in that domain. If you have a user named AlanC, for example, you can't create a computer account in the domain that is also named AlanC.

For more information on developing names for domains, organizational units, users, groups, and computers, see the section titled "Planning Naming Conventions" later in this chapter.

At the same time that you create an object, not only do you assign a name to the object, but Active Directory also assigns identifiers to the object. Active Directory assigns every object a globally unique identifier (GUID), and assigns many objects a security identifier (SID). A *GUID* is typically a 32-digit hexadecimal number that uniquely identifies an object within Active Directory. A *SID* is a unique number created by the Windows 2000 Security subsystem that is assigned only to *security principal objects* (users, groups, and computers) when they are created. Windows 2000 uses SIDs to grant or deny a security principal object access to other objects and network resources.

Active Directory uses a hierarchical naming convention that is based on Lightweight Directory Access Protocol (LDAP) and DNS standards.

Objects in Active Directory can be referenced by using one of three Active Directory name types:

- Relative distinguished name (RDN)
- Distinguished name (DN)
- User principal name (UPN)

A *relative distinguished name* (RDN) is the name that is assigned to the object by the administrator when the object is created. For example, when I create a user named AlanC, the RDN of that user is AlanC. The RDN only identifies an object — it doesn't identify the object's location within Active Directory. The RDN is the simplest of the three Active Directory name types, and is sometimes called the common name of the object.

A *distinguished name* (DN) consists of an object's RDN, plus the object's location in Active Directory. The DN supplies the complete path to the object. An object's DN includes its RDN, the name of the organizational unit(s) that contains the object (if any), and the FQDN of the domain. For example, suppose that I create a user named AlanC in an organizational unit called US in a domain named `Exportsinc.com`. The DN of this user would be:

```
AlanC@US.Exportsinc.com
```

A *user principal name* (UPN) is a shortened version of the DN that is typically used for logon and e-mail purposes. A UPN consists of the RDN plus the FQDN of the domain. Using my previous example, the UPN for the user named AlanC would be:

```
AlanC@Exportsinc.com
```

Another way you can think of a UPN is as a DN stripped of all organizational unit references.

Security

As mentioned previously, Active Directory resides in the Windows 2000 Security subsystem. Together, Active Directory and the Security subsystem protect Active Directory against unauthorized access. The Active Directory/Security subsystem team uses access control lists (ACLs) to determine who can access (and/or modify) an object. An ACL is a list of SIDs and the associated access privileges assigned to each SID. Each object and network resource has an ACL associated with it.

Another security feature of Active Directory is delegation. As stated previously, you can delegate administrative privileges for a container object (such as an organizational unit or domain) and all of its contents to a user or group. This feature enables you to distribute administrative tasks among several employees without having to give each of these employees administrative privileges to the entire network.

Understanding How Active Directory Is Implemented

My goal in this section is to explain how Active Directory is actually implemented on a network. I know you're up to your eyeballs in theory right now, but hang in there just a little longer and I think you'll begin to understand how the pieces fit together.

In the next sections I'll talk a bit about what happens when Active Directory is installed, how the global catalog server works to provide Active Directory information about every object in every domain in a forest, and what Active Directory replication is and how it's implemented on a Windows 2000 network. I'll also introduce the concept of flexible single master operations (FSMO) and discuss the various roles that domain controllers can play. Finally, I'll explain how Active Directory is dependent on DNS and cover some of the limitations that non-Windows 2000 client computers have in terms of using the features of Active Directory.

Installing Active Directory

You have to install Active Directory on all Windows 2000 Server computers that you want to function as domain controllers for a domain — Active Directory is not installed by default. When you first install Active Directory in a domain, Windows 2000 performs two tasks: First, it promotes the computer on which you're installing Active Directory to a domain controller; and second, it creates the Active Directory data store (for that domain) on the newly created domain controller. When you install Active Directory on an additional Windows 2000 Server computer in that domain, Windows 2000 promotes that computer to a domain controller and copies a read/write replica of the Active Directory data store from one of the other existing domain controllers in the domain. A Windows 2000 Server

computer can be a domain controller for only one domain; however, a domain can have multiple domain controllers..

Active Directory is installed by using the Active Directory Installation Wizard that is available in the Configure Your Server tool in the Administrative Tools folder. You can also start this wizard by selecting Start ⇨ Run, typing `Dcpromo.exe`, and pressing Enter.



CROSS-REFERENCE

I'll give detailed instructions for installing Active Directory in Chapter 7. However, I strongly recommend that you read the rest of this chapter, including the "Planning for Active Directory on Your Network" section, before you install Active Directory.

Global Catalog Server

A *global catalog server* is a domain controller that has an additional duty — it maintains a global catalog. You may recall that a global catalog is a master, searchable database that contains information about every object in every domain in a forest. The global catalog contains a complete replica of all objects in Active Directory for its host domain, and contains a partial replica of all objects in Active Directory for every other domain in the forest.

A global catalog server performs two important functions:

- Provides group membership information during logon and authentication
- Helps users locate resources in Active Directory

I'll discuss both of these functions in the next section. A global catalog server provides universal group membership information during a user's logon and authentication process. (A universal group is one that can contain users and other groups from any domain in the forest.) Determining group membership information is a critical part of the logon process, because the groups a user is a member of help determine that user's rights and permissions. The global catalog server provides this group membership information in a highly efficient manner because a global catalog server can respond to the request, instead of having to query a domain controller from each domain in the forest. If a global catalog server is not available, the user will not be able to log on to the domain unless that user is a member of the Domain Admins group.

A global catalog server helps users locate Active Directory objects, regardless of which domain in the forest contains the object. Users can browse the global catalog for available services and resources. In addition, because the global catalog contains information on every object in every domain in the forest, users can search the global catalog for a specific object or resource. For example, suppose I want to locate the e-mail address of another employee in my company. I know the employee's name, so I can query the global catalog by the employee's name, and, assuming I have the appropriate permissions, view pertinent data about the employee I'm searching for, including his or her e-mail address.

A global catalog server, then, provides unified Active Directory information across all domains in the forest; whereas domain controllers only contain information about objects in their own domain. Global catalog servers are a critical part of a multidomain network.

Each domain maintains its own global catalog server. And, by default, there is only one global catalog server in each domain. Normally this is a good idea, but occasionally there may be a valid reason for having more than one global catalog server in a domain.



CROSS-REFERENCE

For more information on optimizing global catalog servers, see Chapter 22.

By default, the first domain controller established in a domain serves as the global catalog server. To move the global catalog to a different server or to add an additional global catalog server, you can use the Active Directory Sites and Services tool. This tool, which is contained in the `Administrative Tools` folder, is available after Active Directory is installed.

Replication

The term *replication*, as applied to Active Directory, refers to the process of copying information and information updates from the Active Directory data store on one domain controller to other domain controllers. The purpose of replication is to synchronize Active Directory data among the domain controllers in the domain and forest. Several types of Active Directory information get replicated:

- **The schema:** The schema is replicated to all domain controllers in the forest.

- **Configuration data:** This data, which includes high-level forest/tree/domain structure, trust, and configuration information, is replicated to all domain controllers in the forest.
- **Domain data:** This complete, detailed information about every object in the domain is replicated only to the domain controllers within this domain.

Replication of Active Directory data is usually partial, meaning that only updated information (versus a complete copy of the Active Directory data store) is copied from one domain controller to other domain controllers. Typically the only time a complete replication is performed is when you install a new domain controller on the network.

Windows 2000 automatically performs replication. Many administrators of small to medium-sized networks will never have to configure replication. That said, replication uses a fair amount of network bandwidth, so sometimes it is beneficial to manage when and how replication takes place, particularly over slow WAN links. You can manage Active Directory replication by using sites, and by using Active Directory Sites and Services.

Sites

A *site* consists of one or more TCP/IP subnets, which are specified by an administrator. Additionally, if a site contains more than one subnet, the subnets should be connected by high-speed, reliable links. Sites do not correspond to domains: You can have two or more sites within a single domain, or you can have multiple domains in a single site. A site is solely a grouping based on IP addresses. Figure 2-7 shows two sites connected by a slow WAN link.

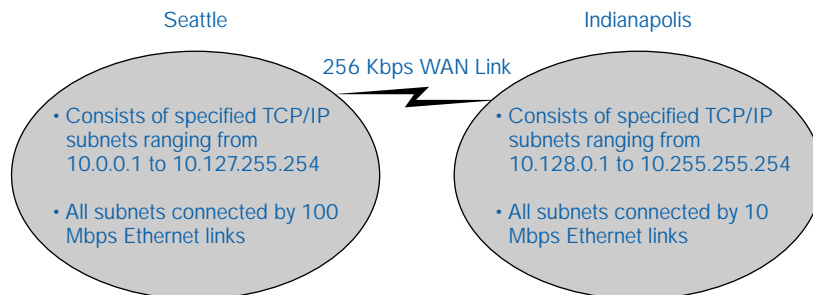


FIGURE 2-7 Two sites

The purpose of sites is to enable servers that regularly copy data to other servers (such as Active Directory replication data) to distinguish between servers in their own site (which are connected by high-speed links) and servers in another site (which are connected by slower-speed WAN links).

Replication between domain controllers in the same site is fast, and typically administrators can permit Windows 2000 to automatically perform this task. Replication between a domain controller in one site and domain controllers in other sites is slower (because it takes place over a slow WAN link) and often should be scheduled by the administrator so that use of network bandwidth for replication is minimized during the network's peak-activity hours.

Sites and Active Directory replication can be configured by using Active Directory Sites and Services.



CROSS-REFERENCE

For detailed information on how to manage and optimize Active Directory replication, including how to use sites, see Chapter 22.

Flexible Single Master Operations

When Microsoft designed Windows 2000, its goal was to have every domain controller equal; instead of having a primary domain controller (PDC) and backup domain controllers (BDCs) like Windows NT 4.0 had, Microsoft wanted to have one class of domain controller that could perform every domain controller-related task. The advantages of this design would be the distribution of server load; the elimination of the need to connect (sometimes across multiple WAN links) to a specific server for the creation of users, change of passwords, and so on; and the elimination of the need to have one server always available to all users.



TIP

When more than one domain controller is able to perform a specific task, that task is referred to as a *multiple master operation*. When only one domain controller can perform a specific task, that task is called a *single master operation*.

So, things were going along pretty smoothly, and then, when Microsoft implemented Active Directory and all of its associated processes, it discovered that a purely multiple master design just wasn't going to work for Windows 2000. Although most domain controller-related tasks can be

performed by any domain controller, a few critical tasks had to be limited to one domain controller in a domain, or to one domain controller in a forest, or utter mayhem and havoc would result. What we've ended up with, then, is a largely multiple master design, with some restricted single master operations. These operations are called *flexible single master operations* (FSMO). The term *flexible* refers to the fact that an administrator can choose which domain controller will perform each restricted single master operation.

There are five different types of flexible single master operations roles (sometimes called master roles) that a domain controller can perform: schema master, domain naming master, PDC emulator, relative ID master, and infrastructure master. Each of these roles defines a specific set of flexible single master operations that only the domain controller assigned to that role can perform.

When you first install Active Directory on the first domain controller in the forest, that domain controller automatically assumes all five of the flexible single master operations roles. As you add domain controllers, you can manually reassign or transfer these master roles to other domain controllers as needed.

Schema Master

The *schema master* is the only domain controller that can make changes to the schema. When you, as an administrator, use an application to change the schema, you don't necessarily need to sit down at the schema master to run this application, nor do you need to know which computer is functioning as the schema master. Windows 2000 seamlessly connects the application to the schema master (across the network) in order to make the desired change.

Because the schema is identical throughout the forest, there can be only one schema master in a forest.

If the computer that is functioning as the schema master is not available when you want to make a change to the schema, you won't be able to change the schema until the schema master becomes available, or until you assign the schema master role to a different domain controller in the forest.

Domain Naming Master

The *domain naming master* is the only domain controller that can add or remove domains to or from the forest. The primary reason for isolating these tasks is to ensure that when a domain is created, its name is unique within the forest.

When you create a new domain in an existing forest by installing Active Directory on the first domain controller in the new domain, the new domain controller contacts the domain naming master to verify that the new domain name is not already in use in the forest, and then, once the domain naming master determines that the domain name is unique, it adds the new domain to the forest.

There can be only one domain naming master in a forest.

If the computer that is functioning as the domain naming master is not available when you want to add or remove a domain, you won't be able to add or remove the domain until the domain naming master becomes available, or until you assign the domain naming master role to a different domain controller in the forest.

PDC Emulator

The *PDC emulator* performs one of two different roles, depending on how Active Directory is implemented on your network.

When Active Directory is configured to interact with Windows NT 4.0 backup domain controllers (BDCs), or to interact with computers that don't have Windows 2000 Directory Service Client software, Active Directory is said to be operating in mixed-mode. When Active Directory operates in mixed-mode, the PDC emulator acts like a Windows NT primary domain controller (PDC). In this situation, the PDC emulator synchronizes user account information (such as user names and passwords) with the existing Windows NT 4.0 BDCs. In addition, when administrators or users of computers that don't run Windows 2000 client software need to make a user account change, that computer must contact the PDC emulator to make the desired change.

When Active Directory is configured to interact only with Windows 2000 domain controllers and computers that run Windows 2000 Directory Service Client software, Active Directory is said to be operating in native-mode. When Active Directory operates in native-mode, the PDC emulator receives password changes more quickly than other domain controllers in the domain. When this occurs, the PDC emulator is said to receive preferential treatment for replication of password changes. Because of this preferential treatment, the PDC emulator is the domain controller that is most likely to have the most current version of a user's password. Therefore, if another domain controller fails to authenticate a user due to an apparently incorrect password, it will forward the user's authentication request to the PDC emulator, and then

convey the PDC emulator's authentication response (either accept or deny) to the user.

There can be only one PDC emulator in each domain in a forest.

If the computer that is functioning as the PDC emulator is not available when you want to perform tasks that require it, you won't be able to perform these tasks until the PDC emulator becomes available, or until you assign the PDC emulator role to a different domain controller in the domain.

Relative ID Master

You may recall that when security principal objects (users, groups, and computers) are created, Active Directory assigns each of these objects a security identifier, or SID. An SID consists of two parts: a domain SID and a relative ID. The *domain SID* identifies the domain in which the object is created, and is the same for all objects created in the domain. The *relative ID* identifies the object in the domain, and is unique for each object created in the domain.

The *relative ID master* (sometimes called the RID master) is the domain controller in the domain that assigns a range of relative IDs to each domain controller in the domain for use in creating SIDs. Because of this assignment by the relative ID master, the potential for domain controllers issuing duplicate SIDs to newly created security principal objects is eliminated.

There can be only one relative ID master in each domain in a forest.

If the computer that is functioning as the relative ID master is not available when a domain controller exhausts its assigned range of relative IDs, that domain controller won't be able to issue SIDs until the relative ID master becomes available, or until you assign the relative ID master role to a different domain controller in the domain.

Infrastructure Master

The *infrastructure master* is the domain controller in the domain that updates group membership information when group members (who are users from other domains) are renamed or moved. For example, say that you have a group named Accounting in your domain. PatL, a user from another domain, is a member of the Accounting group. PatL recently changed her name due to marriage, so you change her user name to PatC. The infrastructure master is responsible for updating the Accounting group membership information to reflect the change in the user's name. (I know the description of this master role sounds bizarre, but this is really how it works.)

There can be only one infrastructure master in each domain in a forest.

If the computer that is functioning as the infrastructure master is not available when you perform tasks that require it, group membership information won't be correctly updated until the infrastructure master becomes available, or until you assign the infrastructure master role to a different domain controller in the domain.



CROSS-REFERENCE

For information on optimizing flexible single master operations and master roles, see Chapter 22.

The Domain Name System

On the Internet, host names are stored in various domains and subdomains that form a hierarchical tree structure called the *Domain Name System (DNS)*. A Windows 2000 computer that has the Domain Name System service installed on it is referred to as a *DNS server*. (More detail about the DNS service appears later in this chapter.)

Active Directory uses the same hierarchical naming conventions as DNS. Because of this, client computers use DNS servers to locate Active Directory domain controllers. Without DNS, Active Directory couldn't function, because client computers wouldn't be able to locate domain controllers.

The Domain Name System (DNS) service that ships with Windows 2000 supports the dynamic update of the DNS database — this feature was not supported by the Microsoft DNS Server service in Windows NT Server 4.0. When I use the phrase “dynamic update of the DNS database,” what I mean is that client computers and servers can dynamically register their host names and IP addresses with the DNS server, without administrator intervention. Previous versions of DNS required the administrator to manually enter host names and their associated IP addresses for each computer on the network.

You're probably getting the idea about now that Active Directory is dependent on DNS. In fact, Active Directory requires a DNS server that supports SRV (service) resource records (RFC 2052). If you have an existing DNS server on your network that meets these requirements, you can use this DNS server for your Windows 2000 network. If you don't have an existing DNS server on your network (or have a DNS server but it doesn't meet these requirements), you can either install a Windows 2000 stand-alone server with the Domain Name System (DNS) service installed

on it to function as your network's DNS server; or you can choose to install the Domain Name System (DNS) service during the installation of Active Directory on the first domain controller on your Windows 2000 network, and thereby make that domain controller into your Windows 2000 network's DNS server.



CROSS-REFERENCE

I'll cover DNS in more depth, including detailed installation instructions, in Chapter 7.

Using Clients with Active Directory

When Active Directory is implemented on a Windows 2000 network, different types of client computers have varying degrees of functionality with Active Directory.

In the most ideal of situations (from Microsoft's point of view, at least), you would have a pure Windows 2000 network, consisting entirely of Windows 2000 Server computers and client computers that run Windows 2000 Professional. In this scenario, all of the client computers (assuming their users had the appropriate permissions) would be able to log on to the domain; locate shared printers, files, and folders; and browse and search Active Directory for available resources. In addition, the administrator could efficiently manage the users and computers of this network by using group policies — one of the key benefits of Active Directory.

However, in reality, a lot of network administrators aren't going to upgrade all of their computers to Windows 2000 (at least not right away, and maybe never). In this case, Active Directory is configured to operate in mixed-mode, and client computers retain the same functionality they had in a Windows NT 4.0 network, but typically they aren't able to use many of the new features of Active Directory.

For example, when functioning as client computers on a Windows 2000 network, Windows NT 4.0 computers (both Server and Workstation) retain the same functionality they have always had. Assuming the users have the necessary permissions, users are still able to log on to the domain; access shared printers, files, and folders; and browse the network (but *not* browse or search Active Directory). There is currently no Directory Service Client software available for Windows NT 4.0 computers. Windows NT 4.0 computers on a Windows 2000 network can't be

managed by using group policy — they can only be managed by using Windows NT 4.0 system policy.

Windows 98 and Windows 95 computers retain the functionality they had when they composed a Windows NT 4.0 network, and in addition, when Directory Service Client software is installed (assuming the users have the appropriate permissions), users are able to browse and search Active Directory. However, Windows 98 and Windows 95 computers can't be managed by using group policy — they can only be managed by using Windows 98 or Windows 95 system policies.

What the whole client issue boils down to is this:

- If your network consists of only Windows 2000 Server and Windows 2000 Professional computers, all computers on the network can use the features of Active Directory and can be managed by using group policy.
- If your network consists of only Windows 2000 Server, Windows 2000 Professional, Windows 98, and Windows 95 computers, all computers on the network can use the features of Active Directory (provided Directory Service Client software is installed on the Windows 98 and Windows 95 computers). However, only the Windows 2000 computers can be managed by using group policy — the Windows 98 and Windows 95 computers can't.
- If your Windows 2000 network includes Windows NT 4.0 computers, the Windows NT 4.0 computers can't use the features of Active Directory — unless you upgrade them to Windows 2000. In addition, only the Windows 2000 computers can be managed by using group policy — the Windows NT 4.0 computers can't.

By not providing Directory Service Client software for Windows NT 4.0 computers, Microsoft appears to be saying that customers must upgrade Windows NT 4.0 computers to Windows 2000 if they want to take advantage of the benefits of Active Directory.



CROSS-REFERENCE

I'll cover installing the Directory Service Client in Chapter 4.

Planning for Active Directory on Your Network

Like any other major network change, implementing Active Directory on your network deserves an appropriate amount of careful planning to ensure a smooth transition.

There are several things you should consider when planning the implementation of Active Directory. Three important matters that should be carefully thought out are your domain design, the naming conventions you'll use, and how existing client computers will fit into your overall plan. I'll cover some planning considerations for each of these elements in the sections that follow.

Planning a Domain Design

Planning an Active Directory domain design for your network is one of the most important tasks you may ever be called on to perform. The domain design is where it all starts—and ultimately, the decisions you make here can have wide-ranging effects on your company.

When you plan a domain design, you not only determine the domain structure you want to use, but also you plan an organizational unit (OU) structure and you plan for the upgrade of any previous domains. The following sections discuss these topics.

Planning a Domain Structure

In most situations, in terms of ease of network administration, the best possible Active Directory domain structure for a Windows 2000 network is a single domain. When a single domain is used, nearly every administrative task is simpler than when multiple domains are used, and there are none of the additional administrative tasks associated with a multidomain structure, such as managing trust relationships. In addition, when a single domain is used, users can locate resources easier and more consistently because every object's attribute is present in the domain's global catalog, while not every attribute of every object is replicated to every global catalog server in a multidomain design.

A speaker at a conference I recently attended stated that anyone who didn't choose a single domain structure for the implementation of Windows 2000 Active Directory was stupid. While for the most part I agree with the

sentiment behind the statement, there are situations where a multiple domain structure deserves some consideration. You might find that the benefits of using multiple domains outweigh the disadvantages when:

- The management structure of your organization is very decentralized, and no one individual has control over the company's network design and implementation. Instead of a single individual managing the company's global network plan, many individuals in different departments plan for their own department, division, or location.
- A parent company has several subsidiaries and does not want to integrate the network or the management structure of the subsidiaries into the parent company's structure.
- Your organization has multiple divisions and it is likely that one or more of the divisions may be sold or spun off as an independent entity.
- You have multiple locations and some of the locations don't have a reliable WAN link to corporate headquarters. You must have a reliable WAN link to keep Active Directory updated when a single domain spans multiple locations.
- Your existing network uses a multiple domain structure, and you don't have the necessary time, money, or manpower to change the domain structure at this time.
- Your company is so large that you anticipate that a single domain in Active Directory would contain millions of objects.



TIP

Windows NT 4.0 had a functional limitation of about 40,000 objects (or 40MB of disk space) in a single domain. So, while there is still some limitation, Windows 2000 can accommodate many more objects in a single domain than Windows NT 4.0 could.

Planning an Organizational Unit Structure

Organizational units (OUs), in my opinion, are meant to serve exactly one purpose: to make network administration easier. They're not meant to mimic the company's organizational chart (although if you administer your network in this manner, then it makes perfect sense to structure OUs in the same way). Organizational units make it easy to delegate authority to assistant administrators and to administer the network in manageable

chunks. Therefore, organizational unit structure should be designed with delegation in mind.

If you administer your network floor-by-floor and building-by-building, then you would probably want your OUs to consist of floors and buildings. If you administer your network by departments and location, you might want to use OUs that consist of departments and geographic locations. I think you get the picture.



CAUTION

Naming an OU after a geographic location can be a risky thing—just think of all of the cities and countries that have changed their names in the past 10 years.

One last note on OUs. OUs are *not* security principal objects. Because of this, OUs can't be used in the same way that a group can be used to assign rights and permissions to users or groups contained within the OU. If you want to assign rights and permissions to multiple users, use a group.

Planning the Upgrade of Previous Domains

If your network is not a brand new Windows 2000 network, you may need to plan for upgrading previously existing Windows NT 4.0 domains to Windows 2000 domains.

In many cases, you may want to move from a single or multiple master domain model (that consists of several domains) to a single domain model. If this is your situation, you must decide where all of the objects from each existing domain will be placed in the new Windows 2000 domain.

For example, you might want to place all of the users, groups, and computers from each existing domain into OUs in the new Windows 2000 domain that correspond to the previous Windows NT 4.0 domains.

Or, since you're taking the time to redesign your network anyway, you might want to totally reorganize your network—one user, computer, and group at a time. Sometimes this is the best way to go, particularly if the network has been added to, patched, and otherwise monkeyed with by numerous administrators over the years.

Planning Naming Conventions

When planning the naming conventions you'll use on your Windows 2000 network, there are several decisions to be made. You'll need to plan how

domains, organizational units (OUs), users, groups, and computers will be named.



TIP

When planning and implementing any naming convention, I recommend that you attempt to keep all names intuitive, short, and simple. This will make everyone's life (especially yours) much easier.

Naming Domains

First of all, you'll need to name your Active Directory domain (or domains). When you do this, you'll need to consider if the name you assign to your domain will be the same when accessed by users on your company's intranet as it will when accessed by external users over the Internet.

As was previously stated, Active Directory domains use DNS domain names. Keep in mind that the maximum length of a fully qualified domain name (FQDN), including periods and all extensions, is 63 characters. Allowed characters include uppercase letters (A–Z), lowercase letters (a–z), numbers (0–9), and the hyphen (-).

If your domain name will be the same for both internal and external users, I recommend choosing a domain name that is as close to your company's name as possible so it will be easily recognized and located by Internet users and by Internet search engines.

If you decide you want to further isolate your company's private intranet behind a firewall, you might choose to use one domain name for internal use, and use a different domain name for external users on the Internet. In this case, the domain name for internal use can be anything you want, and the domain name that external users on the Internet use should be as close to your company's name as possible.

Before you choose your domain name, you should consider using an Internet search engine to determine if the name you want to use is already registered to someone else. Currently, DNS domain names are managed by InterNIC (which stands for Internet Network Information Center).

Finally, if your network is connected to the Internet, you must register your company's DNS domain name (as accessed by users over the Internet) with the appropriate naming authority (InterNIC in the United States). Your Internet service provider will usually perform this task for you.

Naming Organizational Units

After you've named your domain, you'll want to plan how to name your organizational units (OUs). The key point about working with OUs is that they should represent the portion of the organization that is being managed. For example, if you use OUs to manage the users, groups, and computers in your company that are located on a particular floor or in a certain building, the name of the OU should readily identify the floor or building being managed.

Depending on the number of OUs in your organization, you may not need a formal naming scheme for OUs. Instead, you can choose intuitive names for the OUs that represent the particular grouping of objects they contain.

Naming Users, Groups, and Computers

If you have more than a few people in your organization, you'll need to plan a naming convention to use for users, groups, and computers.

When you create user, group, and computer accounts, keep in mind a few rules for these names:

- **Length:** User logon names can be from one to 20 characters long. Computer names should be limited to 15 characters in length for backward compatibility with NetBIOS applications and older client operating systems, such as Windows 95, Windows 98, Windows NT Workstation 4.0, and so on.
- **Uniqueness:** Names created in a domain must be unique within the domain. If you have a user named AlanC, for example, you can't create a computer account in the domain that is also named AlanC.
- **Unacceptable characters:** The following characters can't be used in user and group account names:
“ / \ [] : ; | = , + * ? < >
In addition, a user or group account name can't consist entirely of spaces or periods.
- **Acceptable characters:** As stated previously, computer names can consist of only the following allowed characters: uppercase letters (A–Z), lowercase letters (a–z), numbers (0–9), and the hyphen (-).

There are probably as many naming schemes for users, groups, and computers as there are network administrators. Often, the overall length of a name is limited to eight characters so that the name is compatible with MS-DOS directory name limitations. This eight-character limitation is common, but certainly not mandatory, especially on most of today's networks.

A few common naming conventions for user names include:

- The first seven letters of the user's first name plus the first letter of the user's last name
- The first letter of the user's first name plus the first seven letters of the user's last name
- The user's initials plus the last four digits of the user's employee number
- Various hybrid combinations of the preceding schemes

Finally, you'll need to come up with a way to handle exceptions. It's quite common, for example, for two users to have the same first name and last initial, such as Mike Sinclair and Mike Saunders. If you choose to adopt the first naming scheme in the preceding list, you will need to have a way to resolve these potentially duplicate user names. You could resolve the problem by assigning Mike Sinclair the user account name of MikeS (assuming he was hired first), and assigning Mike Saunders the user account name of MikeSa.

Planning for Clients

Earlier in this chapter, I discussed the limitations that non-Windows 2000 client computers have in terms of their ability to use many of the new features of Active Directory. Well, it's important to consider how your existing client computers will fit into your overall Windows 2000 Active Directory implementation plan.

As I see it, you've basically got four options:

- To achieve optimum functionality of client computers with Active Directory, you can upgrade all of the client computers on your network to Windows 2000 Professional.

- To achieve moderate functionality of client computers with Active Directory, you can install the Directory Service Client on all Windows 98 and Windows 95 computers and upgrade all other client computers (including Windows NT 4.0 computers) to Windows 2000 Professional.
- To achieve limited functionality of client computers with Active Directory, you can install the Directory Service Client on all Windows 98 and Windows 95 computers, and upgrade all other client computers (including Windows NT 4.0 computers) to Windows 2000 Professional only as they are replaced.
- To achieve minimal functionality of client computers with Active Directory, you can do nothing to existing client computers now, and later upgrade client computers to Windows 2000 Professional as they are replaced.

The choice you make will depend on many factors, including management desires, the amount of funding you have to implement the project, the amount of support manpower available, and how important full Active Directory functionality is to you and your organization.



KEY POINT SUMMARY



This chapter introduced several key Active Directory terms and concepts:

- Active Directory is the directory service used by Windows 2000. In Windows 2000, the directory service database is called the Active Directory data store. A read/write copy of the Active Directory data store is physically located on each domain controller in a Windows 2000 domain.
- Active Directory has many key features. It provides fully integrated security, provides ease of administration by using group policies, makes resources easier to locate, is scalable to any size network, and is flexible and extensible.
- Numerous Active Directory terms and concepts were defined and discussed in this chapter:
 - ▶ **Object:** A record in the directory that is defined by a particular set of attributes

- ▶ **Class:** A template used to create a specific type of object
 - ▶ **Schema:** A formal definition of all of the classes of objects and their attributes stored in the directory
 - ▶ **Global catalog:** A master, searchable index that contains information about every object in every domain in a forest
 - ▶ **Hierarchical structure:** A manner of organizing a group of interrelated elements in which the elements are ranked or stacked, one above the other
 - ▶ **Domain:** A logical grouping of networked computers in which one or more of the computers has shared resources and in which all of the computers share a common Active Directory data store
 - ▶ **Organizational unit (OU):** A type of Active Directory object, sometimes called a container object, that can contain objects and other organizational units
 - ▶ **Domain tree:** A hierarchical grouping of one or more domains that must have a single root domain, and may have one or more child domains
 - ▶ **Trust relationship or trust:** An agreement between two domains that enables users in one domain to be authenticated by a domain controller in another domain, and therefore to access shared resources in the other domain
 - ▶ **Forest:** A group of one or more domain trees, linked by transitive trusts, that shares a common schema and global catalog
 - ▶ **Global catalog server:** A domain controller that maintains a global catalog
 - ▶ **Replication:** The process of copying information and information updates from the Active Directory data store on one domain controller to other domain controllers
 - ▶ **Site:** One or more TCP/IP subnets, specified by an administrator; if a site contains more than one subnet, the subnets should be connected by high-speed, reliable links
 - ▶ **Flexible single master operations:** Operations that can only be performed by one specific domain controller
- You must install Active Directory on all Windows 2000 Server computers that you want to function as domain controllers – Active Directory is not installed by default.

- Active Directory is dependent on the Domain Name System (DNS). Both Active Directory and DNS use the same hierarchical naming conventions. In addition, on a Windows 2000 network, client computers use DNS servers to locate Active Directory domain controllers.
- When Active Directory is implemented on a Windows 2000 network, different types of client computers have varying degrees of functionality with Active Directory.
- When getting ready to implement Active Directory on your Windows 2000 network, there are several elements you should consider planning for, including your domain design, naming conventions, and how your existing client computers will fit into your overall Windows 2000 Active Directory implementation plan.

STUDY GUIDE

This section contains several exam readiness questions designed to test your knowledge of Active Directory terms and concepts and help you prepare for the Directory Services exam. You can find the answers to these questions at the end of this chapter.

Assessment Questions

1. Which of the following is *not* a feature of Active Directory?
 - A. It is flexible and extensible.
 - B. It is scalable to any size network.
 - C. It provides ease of administration by utilizing group policies.
 - D. It eliminates the need for trust relationships between domains.
2. Which of the following are classes of Active Directory objects? (Choose all that apply.)
 - A. User
 - B. Group
 - C. Domain
 - D. Workgroup
 - E. Organizational Unit
3. What is the minimum number of domains that a domain tree can contain?
 - A. 1
 - B. 2
 - C. 3
 - D. 4
4. By default, what type of trust are all Windows 2000 trust relationships within a domain tree or forest?
 - A. Explicit trust
 - B. One-way trust
 - C. Transitive trust
 - D. Non-transitive trust

5. Which master role causes the domain controller that performs this role to be the only domain controller in the forest that can add a new domain to the forest?
 - A. PDC emulator
 - B. Schema master
 - C. Relative ID master
 - D. Infrastructure master
 - E. Domain naming master
6. For most large companies, in terms of ease of network administration, what is the optimum number of Active Directory domains to use on their Windows 2000 network?
 - A. 1
 - B. 2
 - C. 3
 - D. More than 3
7. Which of the following are true statements about organizational units (OUs)? (Choose all that apply.)
 - A. They are security principal objects.
 - B. They are sometimes called container objects.
 - C. They should mimic the company's organization chart.
 - D. They should be used to make network administration easier.
 - E. They can contain objects and other organizational units from their own domain.
8. You want to implement Active Directory on your Windows 2000 network. Your network consists of Windows 2000 Server computers, Windows 2000 Professional computers, Windows NT Workstation 4.0 computers, and Windows 98 computers. You want to achieve optimum functionality of all of the client computers with Active Directory. What should you do?
 - A. Install the Directory Service Client on all of the Windows 98 computers.
 - B. Upgrade all of the Windows 98 computers to Windows 2000 Professional.

- C. Upgrade all of the Windows NT Workstation 4.0 computers and Windows 98 computers to Windows 2000 Professional.
- D. Nothing. Windows 2000 will automatically detect all client computers and optimize them to function with Active Directory.

Answers to Chapter Questions

Chapter Pre-Test

1. The correct answer consists of any two of the following key Active Directory features:
 - ▶ It provides fully integrated security.
 - ▶ It provides ease of administration by utilizing group policies.
 - ▶ It makes resources easier to locate.
 - ▶ It is scalable to any size network.
 - ▶ It is flexible and extensible.
2. Domains are the fundamental units that make up Active Directory.
3. A domain tree is a hierarchical grouping of one or more domains that must have a single root domain, and may have one or more child domains. In contrast, a forest is a group of one or more domain trees, linked by transitive trusts, that shares a common schema and global catalog.
4. Active Directory is dependent on DNS.
5. The primary purpose of OUs is the organization of related objects and other organizational units to simplify administration.

Assessment Questions

1. **D.** Active Directory does *not* eliminate the need for trust relationships.
2. **A, B, C, E.** Of the items listed, only “Workgroup” is not a formal class of Active Directory objects.
3. **A.** A domain tree is a hierarchical grouping of one or more domains that must have a single root domain, and may have one or more child domains.

4. **C.** By default, all Windows 2000 trusts within a domain tree or forest are transitive trusts.
5. **E.** The domain naming master is the only domain controller that can add or remove domains to/from the forest.
6. **A.** Using a single domain greatly simplifies the administration of your network.
7. **B, D, E.** Organizational units (OUs), which are sometimes called container objects, can contain objects and other OUs from their own domain, and should be used primarily to make network administration easier. OUs are not security principal objects, and should not mimic the company's organizational chart unless that is how the network is administered.
8. **C.** To achieve *optimum* functionality of the client computers, you must upgrade all of them to Windows 2000. Upgrading some of them and/or installing the Directory Service Client will gain some functionality, but the question specifically states that "optimum functionality" is the required result.