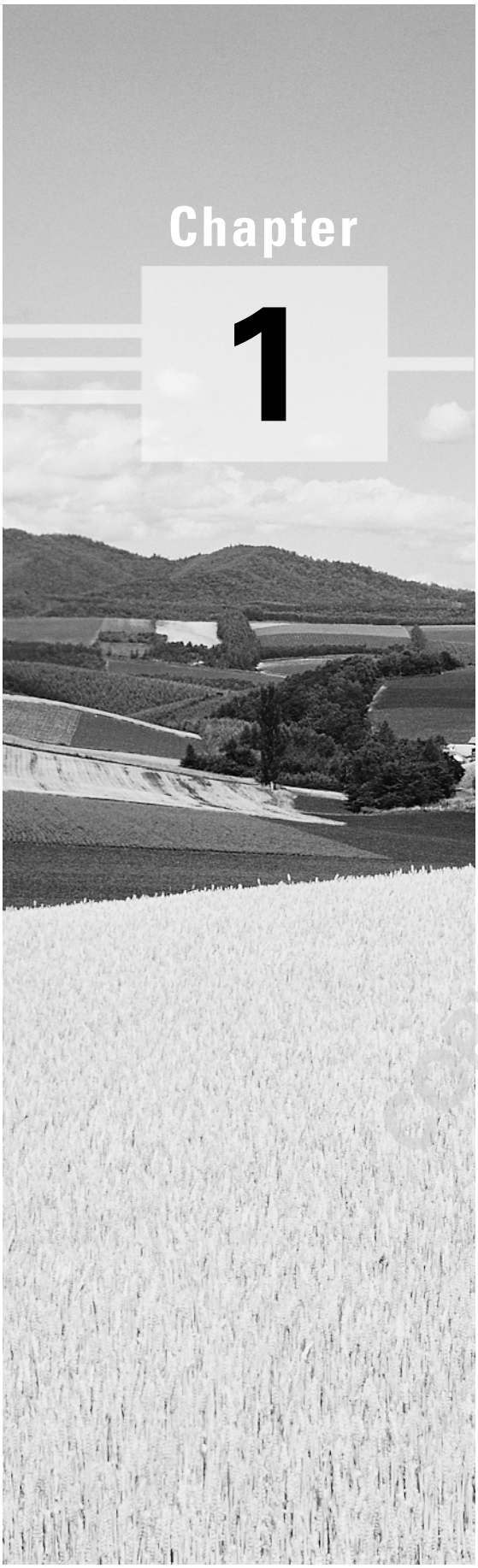


Chapter

1

Overview of the Active Directory

Managing users, computers, applications, and network devices can seem like a never-ending process. However, it's for this very reason that many of us (as systems administrators) have jobs in the first place! Nevertheless, there's a great need for organization, especially when it comes to some of the most fundamental yet tedious tasks we perform every day. That's where the concept of directory services comes in.

To truly appreciate the value of a directory service, let's first look at a real-world example of a situation without organization. Suppose we're trying to find an old friend from college. The first step we would take would probably be to look for their name in the local phone book. If we couldn't find it there, we might try searching in the phone books of a few other cities or on the Internet. If none of those methods were successful, we'd probably resort to calling friends who might have kept in touch with others.

As you can see, this is not an exact science! We could search forever without finding our old friend's telephone number. Part of the problem is due to the lack of a single central repository of phone number information. Without knowing where the information is stored, perseverance and luck are one's strongest tools. Clearly, this is a problem. Yet, it's the way a lot of networks are managed in the real world. That is, information is scattered throughout the organization, and finding what you need may take several phone calls and database searches.

If you've heard about the *Active Directory*, there's a good chance that you already have an idea of its purpose. Microsoft's Active Directory technology is designed to store information about all of the objects within your network environment, including hardware, software, network devices, and users. Furthermore, it is designed to increase capabilities while it decreases administration through the use of a hierarchical structure that mirrors a business's logical organization. In other words, it forms the universal "phone book" we so badly need in the network world!

You've probably also heard that a great deal of planning and training is required to properly implement the Active Directory's many features. We're not talking about a few new administrative tools or check boxes here! In order to reap the true benefits of this new technology, you must be willing to invest the time and effort to get it right. And you'll need buy-in from the entire organization. From end users to executive management, the success of your directory services implementation will be based on input from the entire business. All of these statements about the Active Directory are true.

There's no excuse for poor planning when it comes to the Active Directory. If you're not sure how to configure the directory services for your environment, you'll probably benefit very little from its implementation. In fact, you could make your network more difficult to manage if you improperly implement Windows 2000. It's not a "one size fits all" type of feature. Once you have a good idea for the logical organization of your business and technical environment, however, you will have made much progress toward successfully installing and configuring the Active Directory. That's where the content of this book—and the Microsoft exam for which it will prepare you—come in.

It's a difficult task to cover the various aspects of Windows 2000's most important feature—the Active Directory—even in a whole book. As we briefly mentioned in the introduction, Microsoft's main goal in *Exam 70-217: Implementing and Administering a Microsoft Windows 2000 Directory Services Infrastructure* is to test your ability to *implement* the various features of the Active Directory. The problem is that it doesn't make much sense to begin implementing the Active Directory until you understand the terms, concepts, and goals behind the Active Directory and this big change in the network operating system.

Planning an entire directory services architecture that conforms to your business and technical requirements is beyond the scope of this book. The topic is considerably complex and requires a thorough understanding of all the ramifications for your organization. You must take into account, for example, business concerns, the geographic organization of your company, and its technical infrastructure. In fact, it's such an important topic that Microsoft has decided to test those concepts under a separate exam: *Exam 70-219: Designing a Microsoft Windows 2000 Directory Services Infrastructure*. You can study for that exam using another Sybex book, *MCSE: Windows 2000 Directory Services Design Study Guide*. It would be difficult to overemphasize the importance of planning for Windows 2000 and the Active Directory.

Planning, however, is just one part of the process. Once you have determined exactly *what* your Active Directory should look like, it's time to find out *how* to implement it. And that's what we'll cover throughout this book. Specifically, we'll talk about the various methods for implementing the tools and features of Windows 2000 based on your company's business and technical requirements. Despite the underlying complexity of the Active Directory and all of its features, Microsoft has gone to great lengths to ensure that implementation and management of the Active Directory are intuitive and straightforward, for no technology is useful if no one can figure out how to use it.

In this chapter, we'll take a look at some of the many benefits of using a directory services system and, specifically, Microsoft's Active Directory. We'll cover basic information regarding the various concepts related to Microsoft's Active Directory. The emphasis will be on addressing why the entire idea of directory services came about and how it can be used to improve operations in your environment. We'll then move on to looking at the various logical objects created in the Active Directory and the ways in which you can configure them to work with your network environment. Finally, we'll cover the details related to mapping your organization's physical network infrastructure to the directory services architecture. The goal is to describe the framework on which the Active Directory is based.

With that goal in mind, let's get started!

No specific exam objectives are covered in this chapter, but a basic understanding of how the Active Directory is structured and why it was created are essential for performing well on the exam. If you've had little previous exposure to the Active Directory, or if you want to know how Active Directory is different from NT's domain model, you should definitely read this chapter! Also, be sure to see the appendix, "Planning the Active Directory," for more information on designing a directory services environment.

The World before the Active Directory

The title of this section hints of a time long past. However, the overwhelming majority of networks today run without any single unified directory service. Almost all companies—from small businesses to global enterprises—store information in various disconnected systems. For example, a company might record data about its employees in a human resources database while

network accounts reside on a Windows NT 4 domain controller. Other information—such as security settings for applications—reside within various other systems. And there's always the classic: paper-based forms. The main reason for this disparity is that no single flexible data storage mechanism was available. But implementing and managing many separate systems is a huge challenge for most organizations. Before we look at some potential solutions, let's examine the problem further.

The Benefits of Windows NT 4

Microsoft designed the Windows 2000 operating system platform to succeed its highly successful Windows NT 4 Workstation and Server products. Therefore, it's important to understand the basics of Windows NT before diving into the new features that are available with the Active Directory, a completely new technology introduced with Windows 2000.

The goal of using a network operating system (NOS) is to bring security, organization, and accessibility to information throughout a company's network. In contrast to a peer-to-peer network, properly configured file and print servers allow users and systems administrators to make the most of their resources.

For many years, the realm of network and systems management was one that was controlled by administrators who often worked with cryptic command-line interfaces. That is, only specialists normally managed information systems. Newer network operating systems, such as Novell NetWare and Windows NT, started bringing ease of administration into the network computing world so that network administration no longer needed to be a task delegated to only a few individuals. For example, by bringing the intuitive graphical user interface (GUI) to the world of systems and network administration, Windows NT 4 opened up the doors to simplifying management while still providing the types of security required by most businesses. With these tools, managers and non-technical staff could perform basic systems management functions.

Windows NT Server and Workstation computers offered many benefits, including reliability, scalability, performance, and flexibility. In many cases, companies saw Windows NT 4 as a much more cost-effective solution than their existing client-server solutions. Other benefits of Windows NT included its compatibility with a large installed base of current software products. Application developers could, with a minimal amount of effort, develop programs that would run properly on various Windows-based platforms.

The purpose of this introduction is to provide an overview of the functionality of Windows NT 4. For more details about the product, see `www.microsoft.com/ntserver`.

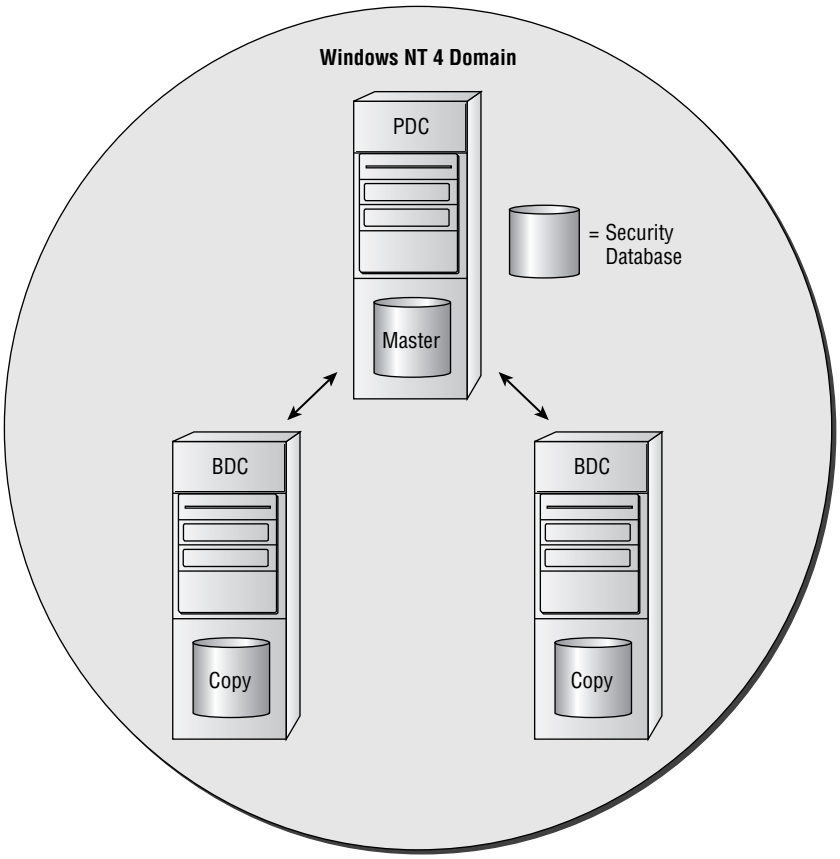
A major design goal for the Windows NT 4 operating system was to provide for a secure yet flexible network infrastructure. A few years ago, few technical and business professionals would have imagined that personal computers would make inroads into corporate server rooms and data centers. For many reasons, including cost-efficiency and price-performance ratios, they have done just that. With these characteristics in mind, we have set the stage for discussing the model used by Windows NT to organize users and secure resources and some of its shortcomings.

The Domain Model in Windows NT 4

The Windows NT 4 platform has met many of the challenges of the network world. However, like any technical solution, it has its limitations. First and foremost, questions regarding the scalability of its rudimentary directory services prevented some potential inroads into corporate data centers. Windows NT uses the concept of a domain to organize users and secure resources. A Windows NT domain is essentially a centralized database of security information that allows for the management of network resources.

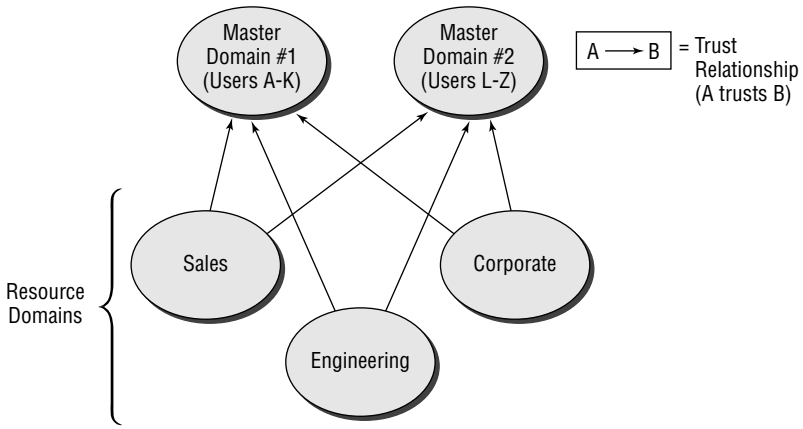
Domains are implemented through the use of Windows NT Server computers that function as *domain controllers*. Every domain has exactly one Primary Domain Controller (PDC) and may have one or more Backup Domain Controllers (BDCs). All network security accounts are stored within a central database on the PDC. To improve performance and reliability in distributed environments, this database is replicated to BDCs. Although BDCs can help distribute the load of network logon requests and updates, there can be only one master copy of the accounts database. This primary copy resides on the PDC, and all user and security account changes must be recorded by this machine and transmitted to all other domain controllers. Figure 1.1 provides an example of such a topology.

FIGURE 1.1 A Windows NT 4 domain topology using PDCs and BDCs



In order to meet some of these design issues, several different Windows NT domain models have been used. Figure 1.2 provides an example of a multiple-master domain topology. In this scenario, user accounts are stored on one or more master domains. The servers in these domains are responsible primarily for managing network accounts. BDCs for these user domains are stored in various locations throughout the organization. Network files, printers, databases, and other resources are placed in resource domains with their own PDC and BDCs. These domains may be created and managed as needed by the organization itself and are often administered separately. In order for resources to be made available to users, each of the resource domains must trust the master domain(s). The overall process places all users from the master domains into global *groups*. These global groups are then granted access to network resources in the resource domains.

FIGURE 1.2 A multiple-master domain topology



The Windows NT domain model works well for small- to medium-sized organizations. It is able to accommodate several thousands of users fairly well, and a single domain can handle a reasonable number of resources. Above these guidelines, however, the network traffic required to keep domain controllers synchronized and the number of trust relationships to manage can present a challenge to network and systems administrators. As the numbers of users grow, it can get much more difficult for the domains to accommodate large numbers of changes and network logon requests.

The Limitations of Windows NT 4

The Windows NT 4 domain model has several limitations that hinder its scalability to larger and more complex environments. We already alluded to one earlier—it can't accommodate the number of users supported by large organizations. Although multiple domains can be set up to ease administration and network constraint issues, administering these domains quickly becomes quite complicated and management-intensive. For example, trust relationships between the domains can quickly grow out of control if not managed properly, and providing adequate bandwidth for keeping network accounts synchronized can be a costly burden on the network.

Domains, themselves, are flat entities used to organize and administer security information. They do not take into account the structure of businesses and cannot be organized in a hierarchical fashion (using subdomains for administrative purposes). Therefore, systems administrators are forced to place users into groups. As groups cannot be nested (that is, have

subgroups), it is not uncommon for many organizations to manage hundreds of groups within each domain. Setting permissions on resources (such as file and print services) can become an extremely tedious and error-prone process.

As far as security is concerned, administration is often delegated to one or more users of the IT department. These individuals have complete control over the domain controllers and resources within the domain itself. This poses several potential problems—both business and technical. As the distribution of administrator rights is extremely important, it would be best to assign permissions to certain areas of the business. However, the options available in the Windows NT operating system were either difficult to implement or did not provide enough flexibility. All of this leads to a less-than-optimal configuration. For example, security policies are often set to allow users far more permissions than they need to complete their jobs.

If you have worked with Windows NT 4 domains in a medium- to large-sized environment, you are probably familiar with many of the issues related to the domain model. Nevertheless, Windows NT 4 provides an excellent solution for many businesses and offers security, flexibility, and network management features unmatched by many of its competitors. As with almost any technical solution, however, there are areas in which improvements can be made.

Now that we've gone over the basics of Windows NT 4 and its directory structure, let's move on and examine how Windows 2000's Active Directory addresses some of these challenges.

The Benefits of the Active Directory

Most businesses have created an organizational structure in an attempt to better manage their environments. For example, companies often divide themselves into departments (such as Sales, Marketing, and Engineering), and individuals fill roles within these departments (such as managers and staff). The goal is to add constructs that help coordinate the various functions required for the success of the organization as a whole.

The Information Technology (IT) department in these companies is responsible for maintaining the security of the company's information. In modern businesses, this involves planning for, implementing, and managing various network resources. Servers, workstations, and routers are common

tools that are used to connect users with the information they need to do their jobs. In all but the smallest environments, the effort required to manage each of these technological resources can be great.

That's where Windows 2000 and Microsoft's Active Directory come in. In its most basic definition, a directory is a repository that records information and makes it available to users. The overall design goal for the Active Directory was to create a single centralized repository of information that securely manages a company's resources. User account management, security, and applications are just a few of these areas. The Active Directory is a data store that allows administrators to manage various types of information within a single distributed database, thus solving one of the problems we stated earlier. This is no small task, but there are many features of this directory services technology that allow it to meet the needs of organizations of any size. Specifically, the Active Directory's features include the following:

Hierarchical Organization In sharp contrast to the flat structure of the Windows NT 4 domain model, the Active Directory is based on a hierarchical layout. Through the use of various organizational components, a company can create a network management infrastructure that mirrors its business organization. So, if a company has 10 major divisions, each of which has several departments, the directory services model can reflect this structure through the use of various objects within the directory. This structure can efficiently accommodate the physical and logical aspects of information resources, such as databases, users, and computers. In addition to the hierarchical organization of objects within the Active Directory, the integration of network naming services with the *Domain Name System* (*DNS*) provides for the hierarchical naming and location of resources throughout the company and on the public Internet.

Extensible Schema One of the foremost concerns with any type of database is the difficulty encountered when trying to accommodate all types of information in one storage repository. That's why the Active Directory has been designed with extensibility (i.e., the ability to add to and change the schema) in mind. In this case, extensibility means the ability to expand the directory schema. The *schema* is the actual structure of the database in terms of data types and location of the attributes. This is important because it allows applications to know where particular pieces of information reside. You cannot delete any portion of the schema, even the pieces that you may add. The information stored within the structure of the Active Directory can be expanded and customized through the use of various

tools. One such tool is the Active Directory Services Interface (ADSI), which is available to Windows developers. ADSI provides objects and interfaces that can be accessed from within common programming languages, such as Visual Basic, Visual C++, and Active Server Pages (ASP). This feature allows the Active Directory to adapt to special applications and to store additional information as needed. It also allows all of the various areas within an organization (or even between them) to share data easily based on the structure of the Active Directory.

Centralized Data Storage All of the information within the Active Directory resides within a single, yet distributed, data repository. This allows users and systems administrators to easily access the information they need from wherever they may be within the company. The benefits of the centralized data storage include reduced administration requirements, less duplication, greater availability, and increased organization of data.

Replication If server performance and reliability were not concerns, it might make sense to store the entire Active Directory on a single server. In the real world, however, accessibility and cost constraints require the database to be replicated throughout the network. The Active Directory provides for this functionality. Through the use of replication technology, the data store can be distributed between many different servers in a network environment. The ability to define sites allows systems and network administrators to limit the amount of traffic between remote sites while still ensuring adequate performance and usability. Reliable data synchronization allows for multimaster replication—that is, all domain controllers can update information stored within the Active Directory and can ensure its consistency at the same time.

Ease of Administration In order to accommodate various business models, the Active Directory can be configured for centralized or decentralized administration. This gives network and systems administrators the ability to delegate authority and responsibilities throughout the organization while still maintaining security. Furthermore, the tools and utilities used to add, remove, and modify Active Directory objects are available from all Windows 2000 domain controllers. They allow for making companywide changes with just a few mouse clicks.

Network Security Through the use of a single logon and various authentication and encryption mechanisms, the Active Directory can facilitate security throughout an entire enterprise. Through the process of *delegation*, higher-level security authorities can grant permissions to other administrators. For ease of administration, objects in the Active Directory tree inherit

permissions from their parent objects. Application developers can take advantage of many of these features to ensure that users are identified uniquely and securely. Network administrators can create and update permissions as needed from within a single repository, thereby reducing chances of inaccurate or outdated configuration.

Client Configuration Management One of the biggest struggles for systems administrators comes with maintaining a network of heterogeneous systems and applications. A fairly simple failure—such as a hard disk crash—can cause hours of work in reconfiguring and restoring a workstation or server. Hours of work can also be generated when users are forced to move between computers and they need to have all of their applications reinstalled and the necessary system settings updated. Many IT organizations have found that these types of operations can consume a great deal of IT staffers' time and resources. New technologies integrated with the Active Directory allow for greatly enhanced control and administration of these types of network issues. The overall benefit is decreased downtime, a better end user experience, and reduced administration.

Scalability and Performance Large organizations often have many users and large quantities of information to manage. The Active Directory was designed with scalability in mind. Not only does it allow for storing up to millions of objects within a single domain, it also provides methods for distributing the necessary information between servers and locations. These features relieve much of the burden of designing a directory services infrastructure based on technical instead of business factors.

Searching Functionality One of the most important benefits of having all of your network resources stored in a single repository is the ability to perform accurate searches. Users often see network operating systems as extremely complicated because of the naming and location of resources. But it shouldn't be that complicated. For example, if we need to find a printer, we should not need to know the name of the domain or print server for that object. Using the Active Directory, users can quickly find information about other users or resources, such as printers and servers, through an intuitive querying interface.

We'll cover the technical aspects of how Windows 2000 addresses all of the above within the technical chapters of this book. For now, keep in mind the various challenges that the Active Directory was designed to address. The scope of this chapter is limited to introducing only the technical concepts on which the Active Directory is based. In order to better

understand this topic, let's now discuss the various areas that make up the logical and physical structure of the Active Directory.

The Active Directory's Logical Structure

Database professionals often use the term schema to describe the structure of data. A schema usually defines the types of information that can be stored within a certain repository and special rules on how the information is to be organized. Within a relational database or Microsoft Excel spreadsheet, for example, we might define tables with columns and rows. Similarly, the Active Directory schema specifies the types of information that are stored within a directory. By default, the schema supports information regarding usernames, passwords, and permissions information. The schema itself also describes the structure of the information stored within the Active Directory data store. The Active Directory data store, in turn, resides on one or more domain controllers that are deployed throughout the enterprise. In this section, we'll take a look at the various concepts that are used to specify how the Active Directory is logically organized.

Components and Mechanisms of the Active Directory

In order to maintain the types of information required to support an entire organization, the Active Directory must provide for many different types of functionality. These include the following:

Data Store When you envision the Active Directory from a physical point of view, you probably imagine a set of files stored on the hard disk that contain all of the objects within it. The term *data store* is used to refer to the actual structure that contains the information stored within the Active Directory. The data store is implemented as just that—a set of files that reside within the file system of a domain controller. This is the fundamental structure of the Active Directory.

The data store itself has a structure that describes the types of information it can contain. Within the data store, data about objects is recorded and made available to users. For example, configuration information about the domain topology, including trust relationships (which we'll cover later in this chapter), are contained within the Active Directory. Similarly, information about users, groups, and computers that are part of the domain are also recorded.

Schema The Active Directory schema consists of rules on the types of information that can be stored within the directory. The schema is made up of two types of objects: attributes and classes. Attributes define a single granular piece of information stored within the Active Directory. First Name and Last Name, for example, are considered attributes, which may contain the values of Bob and Smith. Classes are objects that are defined as collections of attributes. For example, a class called Employee could include the First Name and Last Name attributes.

It is important to understand that classes and attributes are defined independently and that any number of classes can use the same attributes. For example, if we create an attribute called Nickname, this value could conceivably be used to describe a User class and a Computer class. By default, Microsoft has included several different schema objects. In order to support custom data, however, applications developers can extend the schema by creating their own classes and attributes. As we'll see in Chapter 5, "Installing and Managing Trees and Forests," the entire schema is replicated to all of the domain controllers within the environment to ensure data consistency between them.

The overall result of the schema is a centralized data store that can contain information about many different types of objects—including users, groups, computers, network devices, applications, and more.

Global Catalog The *Global Catalog* is a database that contains all of the information pertaining to objects within all domains in the Active Directory environment. One of the potential problems with working in an environment that contains multiple domains is that users in one domain may want to find objects stored in another domain, but they may not have any additional information about those objects.

The purpose of the Global Catalog is to index information stored in the Active Directory so that it can be more quickly and easily searched. In order to store and replicate all of this information, the Global Catalog can be distributed to servers within the network environment. That is, network and systems administrators must specify which servers within the Active Directory environment should contain copies of the Global Catalog. This decision is usually made based on technical considerations (such as network links) and organizational considerations (such as the number of users at each remote site). You can think of the Global Catalog as a universal phone book. Such an object would be quite large and bulky, but also very useful. Your goal (as a systems administrator) would be to

find a balance between maintaining copies of the phone book and making potential users of the book travel long distances to use it.

This distribution of Global Catalog information allows for increased performance during companywide resource searches and can prevent excessive traffic across network links. Since the Global Catalog includes information about objects stored in all domains within the Active Directory environment, its management and location should be an important concern for network and systems administrators.

Searching Mechanisms The best-designed data repository in the world is useless if users can't access the information stored within it. The Active Directory includes a search engine that can be queried by users to find information about objects stored within it. For example, if a member of the Human Resources department is looking for a color printer, they can easily query the Active Directory to find the one located closest to them. Best of all, the query tools are already built into Windows 2000 operating systems and are only a few mouse clicks away.

Replication Although it is theoretically possible to create a directory service that involves only one central computer, there are several problems with this configuration. First, all of the data is stored on one machine. This server would be responsible for processing all of the logon requests and search queries associated with the objects that it contained. Although this scenario might work well for a small network, it would create a tremendous load on servers in larger environments. Furthermore, clients that are located on remote networks would experience slower response times due to the pace of network traffic. Another drawback is that the entire directory would be stored in only one location. If this server became unavailable (due to a failed power supply, for example), network authentication and other vital processes could not be carried out. To solve these problems, the Active Directory has been designed with a replication engine. The purpose of *replication* is to distribute the data stored within the directory throughout the organization for increased availability, performance, and data protection. Systems administrators can tune replication to occur based on their physical network infrastructure and other constraints.

Each of these components must work together to ensure that the Active Directory remains accessible to all of the users that require it and to maintain the accuracy and consistency of its information. Now that we've seen the logical structure and features of the Active Directory, let's move on to looking at organizational concepts.

An Overview of Active Directory Domains

In Windows 2000 Active Directory, a domain is a logical security boundary that allows for the creation, administration, and management of related resources. You can think of a domain as a logical division, such as a neighborhood within a city. Although each neighborhood is part of a larger group of neighborhoods (the city), it may carry on many of its functions independently of the others. For example, resources such as tennis courts and swimming pools may be made available only to members of the neighborhood, while resources such as electricity and water supplies would probably be shared between neighborhoods. So, think of a domain as a grouping of objects that utilizes resources exclusive to its domain, but keep in mind that those resources can also be shared *between* domains.

Although the names and fundamental features are the same, Active Directory domains vary greatly from those in Windows NT. As we mentioned earlier, an Active Directory domain can store many more objects than a Windows NT domain. Furthermore, Active Directory domains can be combined together into *forests* and *trees* to form hierarchical structures. This is in contrast to Windows NT domains, which treat all domains as peers of each other (that is, they are all on equal footing and cannot be organized into trees and forests). Before going into the details, let's discuss the concept of domains.

Within most business organizations, network and systems administration duties are delegated to certain individuals and departments. For example, a company might have a centralized IT department that is responsible for all implementation, support, and maintenance of network resources throughout the organization. In another example, network support may be largely decentralized—that is, each department, business unit, or office may have its own IT support staff. Both of these models may work well for a company, but implementing such a structure through directory services requires the use of logical objects.

Domains are composed of a collection of computers and resources that share a common security database. An Active Directory domain contains a logical partition of users, groups, and other objects within the environment. Objects within a domain share several characteristics, including the following:

Group Policy and Security Permissions Security for all of the objects within a domain can be administered based on one set of policies. Thus, a domain administrator can make changes to any of the settings within the domain. These settings can apply to all of the users, computers, and objects within the domain. For more granular security settings, however, permissions can be granted on specific objects, thereby distributing

administration responsibilities and increasing security. Domains are configured as a single security entity. Objects, permissions, and other settings within a domain do not automatically apply to other domains.

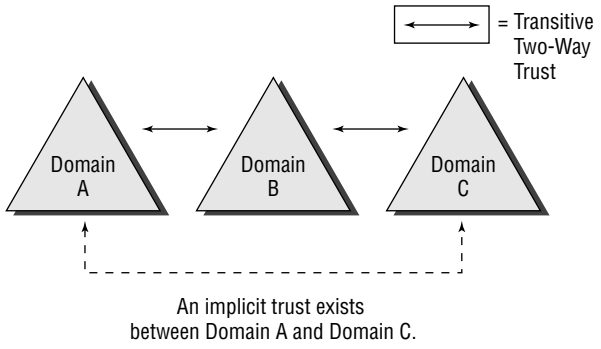
Hierarchical Object Naming All of the objects within an Active Directory container share a common namespace. When domains are combined together, however, the namespace is hierarchical. For example, a user in one department might have an object name called `janedoe@engineering.microsoft.com` while a user in another department might have one called `johndoe@sales.microsoft.com`. The first part of the name is determined by the name of the object within the domain (in these examples, the username). The suffix is determined by the organization of the domains. The hierarchical naming system allows each object within the Active Directory to have a unique name. For more information on naming Active Directory objects, see the appendix.

Hierarchical Properties Containers called *organizational units* (OUs) (described later) can be created within a domain. These units are used for creating a logical grouping of objects within the Active Directory. The specific user settings and permissions that are assigned to these objects can be inherited by lower-level objects. For example, if we have an organizational unit for the North America division within our company, we can set user permissions on this object. All of the objects within the North America object (such as the Sales, Marketing, and Engineering departments) would automatically inherit these settings. This makes administration easier, but inheritance is an important concept to remember when implementing and administering security since it results in the implicit assignment of permissions. The proper use of hierarchical properties allows systems administrators to avoid inconsistent security policies (such as a minimum password length of six characters in one object and a minimum password length of eight characters in another).

Trust Relationships In order to facilitate the sharing of information between domains, trust relationships are automatically created between them. Additionally, the administrator can break and establish trust relationships based on business requirements. A trust relationship allows two domains to share security information and objects, but does not automatically assign permissions to these objects. This allows users who are contained within one domain to be granted access to resources in other domains. To make administrating trust relationships easier, Microsoft has made transitive two-way *trusts* the default relationship between

domains. As shown in Figure 1.3, if Domain A trusts Domain B and Domain B trusts Domain C, Domain A implicitly trusts Domain C.

FIGURE 1.3 Transitive two-way trust relationships



Generally, triangles are used to represent Active Directory domains (thereby indicating their hierarchical structure), and circles are used to represent flat domains (such as those in Windows NT).

Overall, the purpose of domains is to ease administration while providing for a common security and resource database.

Using Multiple Domains

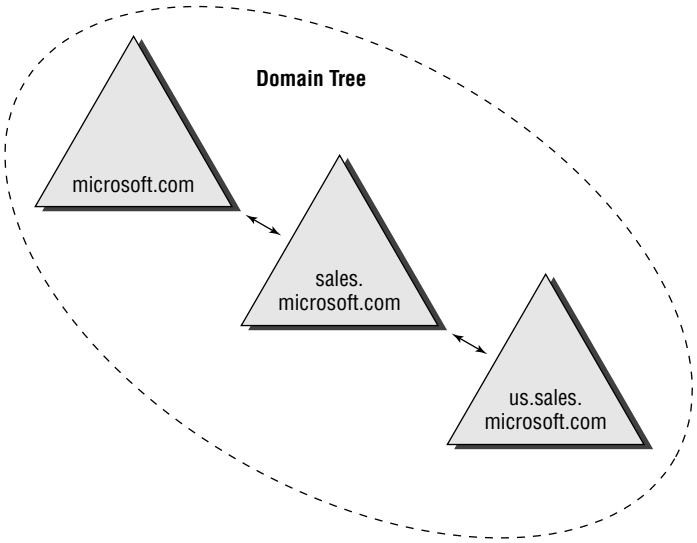
Although the flexibility and power afforded by the use of an Active Directory domain will meet the needs of many organizations, there are reasons for which companies might want to implement more than one domain. We'll cover these planning issues in the appendix. For now, however, it is important to know that domains can be combined together into domain trees.

Domain trees are hierarchical collections of domains that are designed to meet the organizational needs of a business (see Figure 1.4). Trees are defined by the use of a contiguous namespace. For example, the following domains are all considered part of the same tree:

- `microsoft.com`
- `sales.microsoft.com`
- `research.microsoft.com`
- `us.sales.microsoft.com`

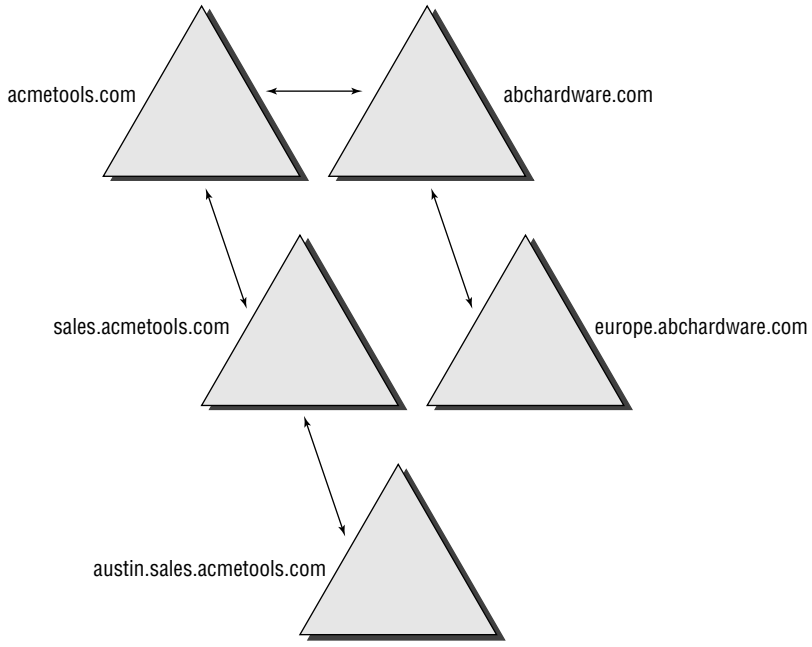
Notice that all of these domains are part of the `microsoft.com` domain. Domains within trees still maintain separate security and resource databases, but they can be administered together through the use of trust relationships. By default, trust relationships are automatically established between parent and child domains within a tree.

FIGURE 1.4 A domain tree



Although single companies will often want to configure domains to fit within a single namespace, noncontiguous namespaces may be used for several reasons. We'll look at several of these reasons in Chapter 5. When domain trees are combined together into noncontiguous groupings, they are known as forests (see Figure 1.5). Forests often contain multiple noncontiguous namespaces consisting of domains that are kept separate for technical or political reasons. Just as trust relationships are created between domains within a tree, trust relationships are also created between trees within a forest so resources can be shared between them.

FIGURE 1.5 An Active Directory forest

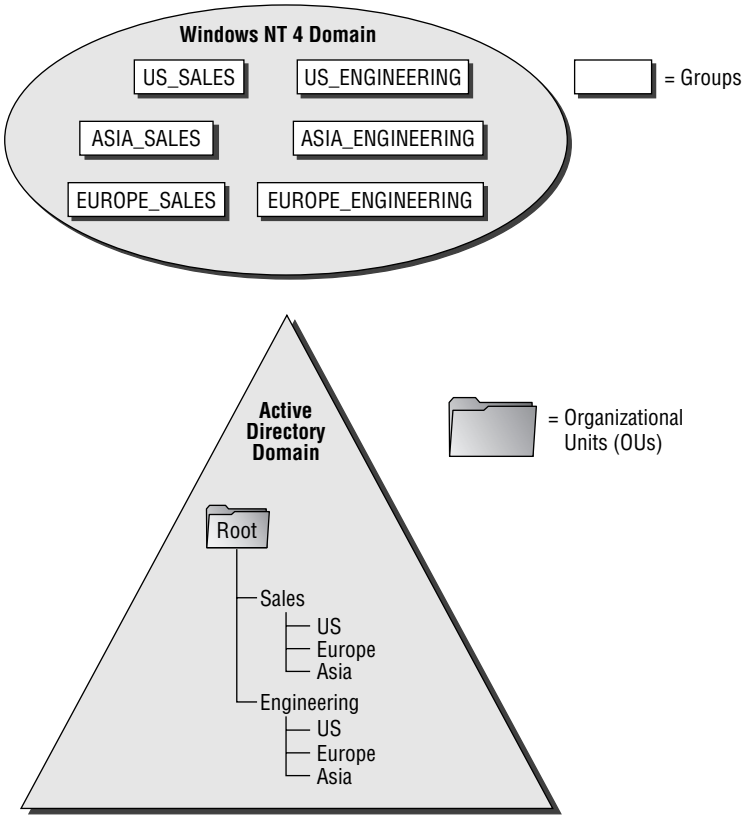


Physically, domains are implemented and managed by the use of domain controllers. We'll cover this topic later in this chapter.

Creating a Domain Structure with Organizational Units

As we mentioned earlier, one of the fundamental limitations of the Windows NT 4 domain organization is that it consists of a flat structure. All users and groups are stored as part of a single namespace. Real-world organizations, however, often require further organization within domains. For example, we may have three thousand users in one domain. Some of these should be grouped together in an Engineering group. Within the Engineering group, we might also want to further subdivide users into other groups (for example, Development and Testing). The Active Directory supports this kind of hierarchy. Figure 1.6 provides a depiction of the differences between the structure of a Windows NT 4 domain and that of an Active Directory domain.

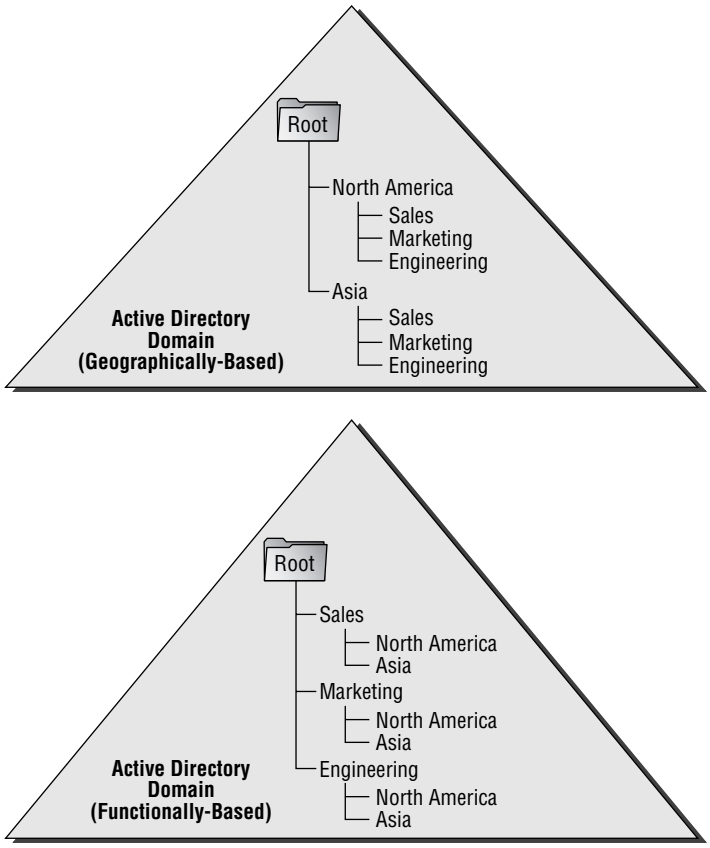
FIGURE 1.6 Windows NT 4 vs. Active Directory domains



The fundamental unit of organization within an Active Directory domain is the organizational unit (OU). OUs are container objects that can be hierarchically arranged within a domain. Figure 1.7 provides an example of a typical OU setup. OUs can contain other objects such as users, groups, computers, and even other OUs. The proper planning and usage of OUs are important because they are generally the objects to which security permissions and Group Policies are assigned. A well-designed OU structure can greatly ease the administration of Active Directory objects.

OUs can be organized based on various criteria. For example, we might choose to implement an OU organization based on the geographic distribution of our company's business units.

FIGURE 1.7 Two different OU hierarchy models



We'll look at various planning issues for OUs in the appendix.

Active Directory Object Names

A fundamental feature of a directory service is that each object within the directory should contain its own unique name. For example, our organization may have two different users named John Smith (who may or may not be in different departments or locations within the company). There should be some unique way for us to distinguish these users (and their corresponding user objects).

Generally, this unique identifier is called the *distinguished name*. Within the Active Directory, each object can be uniquely identified using a long

name that specifies the full path to the object. Following is an example of a distinguished name:

```
/O=Internet/DC=Com/DC=MyCompany/DC=Sales
 /CN=Managers/CN=John Smith
```

In the above name, we have specified the following several different types of objects:

Organization (O) The company or root-level domain. In this case, the root level is the Internet.

Domain Component (DC) A portion of the hierarchical path. DCs are used for organizing objects within the directory service. The DCs specify that the user object is located within the `sales.mycompany.com` domain.

Common Name (CN) Specifies the names of objects in the directory. In this example, the user John Smith is contained within the Managers container.

When used together, the components of the distinguished name uniquely identify where the user object is stored. Instead of specifying the full distinguished name, we might also choose to use a relative distinguished name. This name specifies only part of the path above and is relative to another object. For example, if our current context is already the Managers group within the `sales.mycompany.com` domain, we could simply specify the user as `CN=John Smith`.

Note that if we change the structure of the domain, the distinguished name of this object would also change. A change might happen if we rename one of the containers in the path or move the user object itself. This type of naming system allows for flexibility and the ability to easily identify the potentially millions of objects that might exist in the Active Directory.

User, Computer, and Group Objects

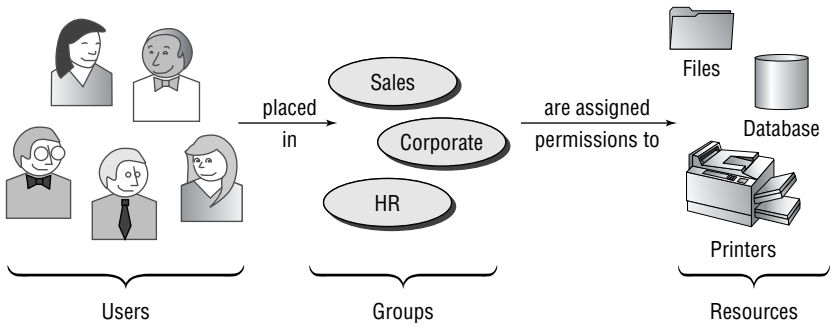
The real objects that you will want to control and manage with the Active Directory are the users, computers, and groups within your network environment. These are the types of objects that allow for the most granular level of control over permissions and allow you to configure your network to meet business needs.

User accounts are used to enforce the security within the network environment. These accounts define the login information and passwords that are used to receive permissions to network objects. Computer objects allow systems

administrators to configure the functions that can be performed on client machines throughout the environment. Both User accounts and Computer objects enable security to be maintained at a granular level.

Although security can be enforced by placing permissions directly on User and Computer objects, it is much more convenient to combine users into *groups*. For example, if there are three users who will require similar permissions within the Accounting department, we could place all of them in one group. If users are removed or added to the department, we could easily make changes to the group without having to make any further changes to security permissions. Figure 1.8 shows how groups can be used to easily administer permissions.

FIGURE 1.8 Using groups to administer security



There are two main types of groups within the Active Directory: security groups and distribution groups. *Security groups* are used for the administration of permissions. All members of a security group will receive the same security settings. *Distribution groups*, on the other hand, are used only for sending e-mail and other messages to several different users at once. They do not involve the maintenance of security permissions but can be helpful in handling multiple users.

Overall, the proper use of groups assists greatly in implementing and managing security and permissions within the Active Directory.

The Active Directory's Physical Structure

So far, we have focused our attention on the logical units that make up the Active Directory. That is, the ideas presented so far are designed to

bring organization to the structure of the network. What we haven't discussed is exactly *how* domains, trees, forests, and the Active Directory itself are created and managed. In this section, we'll see how various servers and network devices can be used to implement and manage the components of the Active Directory.

Server Roles within the Active Directory

The Active Directory data store is stored on one or more computers within an organization's network environment. All editions of the Windows 2000 Server platform are able to participate in Active Directory domains under the following roles:

Domain Controllers The heart of the Active Directory's functionality resides on domain controllers. These machines are responsible for maintaining the Active Directory data store, including all of its objects, and for providing security for the entire domain. Although an Active Directory configuration may involve only one domain controller, it is much more likely that organizations will have more servers in order to increase performance and establish fault-tolerance. All of the information that resides within the Active Directory is synchronized between the domain controllers, and most changes can be made at any of these servers. This functionality is referred to as multimaster replication and is the basis through which Active Directory information is distributed throughout an organization.

Member Servers Often, you will want to have servers that function as part of the domain but are not responsible for containing Active Directory information or authenticating users. Common examples include file/print servers and Web servers. A Windows 2000 Server computer that is a member of a domain but is not a domain controller itself is referred to as a *member server*. By using member servers, systems administrators can take advantage of the centralized security database of the Active Directory without dedicating server processing and storage resources to maintaining the directory information.

Stand-Alone Servers It is possible to run Windows 2000 Server computers in a workgroup environment that does not include Active Directory functionality at all. These machines are known as stand-alone servers. They maintain their own security database and are administered independently of other servers, as no centralized security database exists. Stand-alone servers might be used for functions such as public Web servers or in situations in which only a few users require resources from a machine and the administrative overhead for managing security separately on various machines is acceptable.

A major benefit in the Windows 2000 Server operating system is the ability to easily promote and demote domain controllers after the operating system has been installed. Unlike the situation with Windows NT 4, reinstallation of the entire operating system is no longer required to change the role of a server. Furthermore, by properly promoting and demoting domain controllers, you can effectively move them between domains, trees, and forests.

In addition to the various types of server roles that the Windows 2000 Server platform can take on within the Active Directory domains, the Active Directory requires systems administrators to assign specific functionalities to other servers. In discussing replication, certain servers might be referred to as masters. Masters contain copies of a database and generally allow both read and write operations. Some types of replication may allow multiple masters to exist, while others specify that only a single master is allowed. Certain tasks within the Active Directory work well using multimaster replication. For example, the ability to update information at one or more of the domain controllers can speed up response times while still maintaining data integrity through replication. Other functions, however, better lend themselves to being defined centrally. These operations are referred to as single-master operations because the function only supports modification on a single machine in the environment. These machines are referred to as Operations Masters servers. The role of these servers is to handle operations that are required to ensure consistency within an Active Directory environment. Some of these are unique within a domain, and others are unique within the tree or forest. The changes made on these machines are then propagated to other domain controllers, as necessary. The various roles for Operations Masters servers within the Active Directory include the following:

Schema Master As we mentioned earlier, one of the benefits of the Active Directory schema is that it can be modified. All changes to the schema, however, are propagated to all domain controllers within the forest. In order for the information to stay synchronized and consistent, it is necessary for one machine within the entire tree or forest to be designated as the Schema Master. All changes to the schema must be made on this machine. By default the first domain controller installed in the tree or forest is the Schema Master.

Domain Naming Master When creating, adding, or removing domains, it is necessary for one machine in the tree or forest to serve as a central authority for the Active Directory configuration. The Domain Naming Master ensures that all of the information within the Active Directory forest is kept consistent and is responsible for registering new domains.

Within each Active Directory domain, the following roles can be assigned to domain controllers:

Relative ID Master A fundamental requirement of any directory service is that each object must have a unique identifier. All users, groups, computers, and other objects within the Active Directory, for example, are identified by a unique value. The Relative ID (RID) Master is responsible for creating all of these identifiers within each domain and for ensuring that objects have unique IDs between domains by working with RID Masters in other domains.

Primary Domain Controller (PDC) Emulator In order to support Windows NT, Windows 2000 Server must have the ability to serve as a Windows NT PDC. Microsoft has made a conscious decision to allow networks to work in a mixed mode of Windows NT domains and Active Directory domains in order to facilitate the migration process (and encourage more people to buy Windows 2000!). As long as there are computers in the environment running Windows NT 4, the PDC Emulator will allow for the transmission of security information between domain controllers. This provides for backward compatibility while an organization moves to Windows 2000 and the Active Directory.

Infrastructure Master Managing group memberships is an important role fulfilled manually by systems administrators. In a potentially distributed Active Directory environment, though, it is important to make sure that group and user memberships stay synchronized throughout the network. In order to understand how information might become inconsistent, let's look at an example using two domain controllers named DC1 and DC2. Suppose we make a change to a user's settings on DC1. At the same time, suppose another systems administrator makes a change to the same user account but on DC2. There must be some way to determine which change takes precedence over the other. More important, all domain controllers should be made aware of these changes so that the Active Directory database information remains consistent. The role of the Infrastructure Master is to ensure consistency between users and their group memberships as changes, additions, and deletions are made.

If there is more than one domain controller in the domain, the Global Catalog should not reside on the same server as the Infrastructure Master. This would prevent it from seeing any changes to the data and would result in replication not occurring between the various domain controllers.

It is important to note that the above assignments are *roles* and that a single machine may perform multiple roles. For example, in an environment in which only a single domain controller exists, that server will assume all of the above roles by default. On the other hand, if multiple servers are present, these functions can be distributed between them for business and technical reasons. By properly assigning roles to the servers in your environment, you'll be able to ensure that single-master operations are carried out securely and efficiently.

Accessing the Active Directory through LDAP

In order to insert, update, and query information from within the Active Directory, Microsoft has chosen to employ the worldwide Internet Engineering Task Force (IETF) standard protocol called the *Lightweight Directory Access Protocol* (*LDAP*). LDAP is designed to allow for the transfer of information between domain controllers and to allow users to query information about objects within the directory.

As LDAP is a standard, it also facilitates interoperability between other directory services. Furthermore, communications can be programmed using objects such as the Active Directory Services Interface (ADSI). For data transport, LDAP can be used over TCP/IP, thus making it an excellent choice for communicating over the Internet, as well as private TCP/IP-based networks.

Managing Replication with Sites

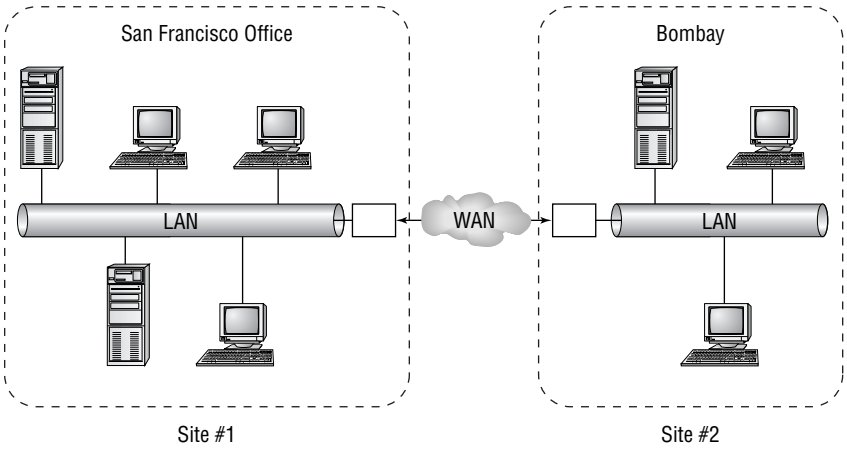
A common mistake made in planning the Active Directory is to base its structure on the technical constraints of a business instead of on business practices. For instance, a systems administrator might recommend that a separate domain be placed at each of a company's three remote sites. The rationale for this decision is understandable—the goal is to reduce network traffic between potentially slow and costly remote links. However, the multidomain structure may not make sense for organizations that have a centralized IT department and require common security settings for each of the three locations.

In order to allow the Active Directory to be based on business and political decisions while still accommodating network infrastructure issues, Windows 2000 supports the concept of *sites*. Active Directory sites are designed to define the physical layout of a company's network by taking into account multiple subnets, remote access links, and other network factors. When performing vital functions between domain controllers, for example, you might want to limit bandwidth usage across a slow link. However, within your

local area network (LAN) environment, you will want replication to occur as quickly as possible to keep machines synchronized.

Sites are usually defined as locations in which network access is quick and inexpensive. Windows 2000 uses sites to determine when and how information should be replicated between domain controllers and other machines within the environment. Figure 1.9 provides an example of how a distributed company might choose to implement sites.

FIGURE 1.9 A typical site configuration



It is important to understand the distinction between logical and physical components of the Active Directory. When planning your objects and domains, you will want to take into account the business requirements of your organization. This will create the logical structure of the directory. In planning for the implementation of the Active Directory, however, you must take into account your network infrastructure—the physical aspects. Sites provide a great way to isolate these two requirements.

Active Directory Names and DNS

The Domain Name System (DNS) is a distributed database built upon an Internet standard that is used to resolve friendly, hierarchical names to TCP/IP network addresses. Systems administrators who have to remember many server IP addresses will easily recall the need for DNS—it can be quite a difficult and error-prone process to remember all of these numbers. For example, if we have

a server on the Internet with an IP address of `24.133.155.7`, we may want to give it a friendly name, such as `sales.mycompany.com`. Instead of typing the IP address every time we need to access the resource, we could specify the fully-qualified name of the machine and leave it to the DNS servers on the Internet to resolve the address.

Understanding TCP/IP is vital to understanding the use of almost any modern network operating system. If you're planning to deploy a Windows 2000 environment, be sure you take the time to learn the details of working with TCP/IP.

The Windows 2000 Active Directory relies on DNS for finding DCs and naming and accessing Active Directory objects. Windows 2000 includes a DNS server service that can be used for automatically updating records that store machine name to IP address mappings. DNS offers many advantages. First, it is the primary name resolution method used on the Internet. Therefore, it has widespread support in all modern operating systems and works well between various operating system platforms.

Second, DNS is designed with fault-tolerance and distributed databases in mind. If a single DNS server does not have the information required to fulfill a request for information, it automatically queries another DNS server for this information. Systems administrators are only responsible for maintaining the DNS entries for their own machines. Through the use of efficient caching, the load of performing worldwide queries on large networks can be minimized.

The various technical details related to DNS are well beyond the scope of this section, but we will cover them later in Chapter 2, "Integrating DNS with the Active Directory."

Real World Scenario

Upgrading Windows NT Domains to Active Directory

You are a consultant doing work for an organization that has decided to move its environment to the Active Directory. However, before the upgrade can begin, you must first design a suitable Active Directory. You have several choices that need to be made and many considerations to take into account. Factors that should affect your decision include the following:

Political Issues How does the current business operate—as single, independent business units, or as a centralized environment? Who will be responsible for administering portions of the network?

Network Issues What types of network connections are present between your remote offices? How reliable are these connections? Also, what are the domain name requirements for this environment?

Organizational Structure How are various areas of the business structured? For example, do the departments operate individually, with separate network administrators for each department? Or is the environment much more centralized?

Based on the answers to these questions, you might choose to implement only a single domain. This method provides for simple administration and should meet most requirements. You may, however, have other concerns (such as the need to support multiple DNS namespaces). In any case, the best solution will be based on the specific needs of the environment.

Summary

In this chapter, we took a high-level overview of the concepts related to the Active Directory. Specifically, we discussed the following:

- The benefits of implementing the Active Directory
- How the Active Directory compares to Windows NT's domain model
- How and why multiple Active Directory domains can be created
- The logical components of an Active Directory environment
- The naming of Active Directory objects
- The physical components that make up an Active Directory environment

Exam Essentials

Understand the problems that Active Directory is designed to solve. The creation of a single, centralized directory service can make network operations and management much simpler. The Active Directory solves many shortcomings in Windows NT's domain model.

Understand Active Directory design goals. The Active Directory should be structured to mirror an organization's logical structure. Understand

the factors that you should take into account, including business units, geographic structure, and future business requirements.

Understand features of Active Directory. Understand how and why Microsoft has included features that allow for extensibility, centralized data storage, replication, ease of administration, security, and scalability.

Remember the Operations Master server roles that are required in an Active Directory environment. Operations Master roles are vital to the proper operations of the Active Directory. Some of these roles must be present in each Active Directory domain while others require only one for the entire Active Directory environment.

Understand the basic domain structure for an Active Directory environment. An Active Directory environment can consist of only a single domain, or it can include multiple domains that form a tree. Multiple trees can be combined into a forest.

Key Terms

Before you take the exam, be certain you are familiar with the following terms:

Active Directory	Lightweight Directory Access Protocol (LDAP)
delegation	member server
distinguished name	organizational units (OUs)
Domain Name Systems (DNS)	replication
Distribution group	schema
domain	Security group
domain controllers	sites
forests	trees
Global Catalog	trusts
groups	

Review Questions

1. Which of the following is not a feature of the Active Directory?
   - **A.** The use of LDAP for transferring information
   - **B.** Reliance on DNS for name resolution
   - **C.** A flat domain namespace
   - **D.** The ability to extend the schema
2. Domains provide which of the following functions?
   - **A.** Creating security boundaries to protect resources and ease of administration
   - **B.** Easing the administration of users, groups, computers, and other objects
   - **C.** Providing a central database of network objects
   - **D.** All of the above
3. Which of the following types of servers contain copies of the Active Directory database?
   - **A.** Member servers
   - **B.** Domain controllers
   - **C.** Stand-alone servers
   - **D.** None of the above
4. Which of the following objects are used for creating the logical structure within Active Directory domains?
   - **A.** Users
   - **B.** Sites
   - **C.** Organizational units (OUs)
   - **D.** Trees
   - **E.** None of the above

5. Which of the following is *false* regarding the naming of Active Directory objects?

   A. The Active Directory relies on DNS for name resolution.

   B. Two objects can have the same relative distinguished name.

   C. Two objects can have the same distinguished name.

   D. All objects within a domain are based on the name of the domain.

6. Which of the following are *true* regarding Active Directory trust relationships?

   A. Trusts are transitive.

   B. By default, trusts are two-way relationships.

   C. Trusts are used to allow the authentication of users between domains.

   D. All of the above.

7. Which of the following protocols is used to query Active Directory information?

   A. LDAP

   B. NetBEUI

   C. NetBIOS

   D. IPX/SPX

8. Which of the following is not true regarding the Windows NT domain namespace?

   A. Windows NT domains have a hierarchical namespace.

   B. Windows NT domains allow thousands of users.

   C. Windows NT domains can be implemented as master domains.

   D. Windows NT domains can be implemented as resource domains.

   E. All of the above.

9. Which of the following is a possible role for a Windows 2000 Server?

   A. Member server

   B. Primary Domain Controller

   C. Backup Domain Controller

   D. Stand-alone server

   E. Both A and D

10. Which of the following statements is *true* regarding domain controllers?

    A. All Active Directory domain controllers are automatically configured as Windows NT domain controllers.

    B. Windows NT domain controllers can host a copy of the Active Directory database.

    C. Windows 2000 domain controllers can be configured to provide the functionality of Windows NT domain controllers.

    D. None of the above.

11. Which of the following is not a characteristic of DNS?

    A. Built-in redundancy

    B. Reliance on proprietary technologies

    C. Scalability

    D. Distributed databases

12. An organization uses 12 Active Directory domains in a single forest. How many Schema Masters must this environment have?

    A. 0

    B. 1

    C. 12

    D. More than 12

    E. None of the above

13. An organization has three remote offices and one large central one. How many sites should this environment contain?

A. 0

B. 1

C. 3

D. 4

E. Not enough information

14. Which of the following features of the Active Directory allows information between domain controllers to remain synchronized?

A. Replication

B. The Global Catalog

C. The schema

D. None of the above

15. Jane is a systems administrator for a large, multidomain, geographically distributed network environment. The network consists of a large, central office and many smaller remote offices located throughout the world. Recently, Jane has received complaints about the performance of Active Directory–related operations from remote offices. Users complain that it takes a long time to perform searches for network resources (such as Shared Folders and Printers). Jane wants to improve the performance of these operations. Which of the following components of the Active Directory should she implement at remote sites to improve the performance of searches conducted for objects in *all* domains?

A. Data store

B. Global Catalog

C. Schema

D. None of the above

Answers to Review Questions

1. C. The Active Directory uses a hierarchical namespace for managing objects.

2. D. All of these options are features of domains and are reasons for their usefulness.

3. B. Only domain controllers contain a copy of the Active Directory database. Member servers rely on the Active Directory but do not contain a copy of the database, and stand-alone servers do not participate in the Active Directory at all.

4. C. OUs are used for creating a hierarchical structure within a domain. Users are objects within the directory, sites are used for physical planning, and trees are relationships between domains.

5. C. The distinguished name of each object in the Active Directory must be unique, but the relative distinguished names may be the same. For example, we might have a User object named Jane Doe in two different containers.

6. D. Trusts are designed for facilitating the sharing of information and have all of the above features.

7. A. LDAP is the IETF standard protocol for accessing information from directory services. It is also the standard used by the Active Directory.

8. A. The Windows NT namespace is a flat model because groups cannot contain other groups and there is no hierarchical structure within a domain. The components of Active Directory domains, on the other hand, allow the use of organizational units (OUs) in order to create a manageable hierarchy within a domain.

9. E. Primary Domain Controllers and Backup Domain Controllers are only used in Windows NT domains.

10. C. Through the use of the PDC Emulator functionality, Windows 2000 domain controllers can provide services for Windows NT domains.

11. B. DNS is a worldwide standard that is widely supported in all modern operating systems.

12. B. Only one Schema Master is allowed in an Active Directory environment, regardless of the number of domains.

13. E. The site topology is completely independent from domain architecture—a domain can span many sites, and many domains can be part of the same site. The fact that the organization has four locations does not necessarily mean that it should use a specific number of sites. Rather, this determination should be made based on physical network characteristics.

14. A. Replication ensures that information remains synchronized between domain controllers.

15. B. The Global Catalog contains information about multiple domains, and additional Global Catalog servers can greatly increase the performance of operations such as searches for shared folders and printers. The other options are features of the Active Directory, but they are not designed for fast searching across multiple domains.