WHAT IS ACTIVE DIRECTORY?

Active Directory is a directory service. The term directory service refers to two things — a directory where information about users and resources is stored and a service or services that let you access and manipulate those resources. Active Directory is a way to manage all elements of your network, including computers, groups, users, domains, security policies, and any type of user-defined objects. It melds several NT services and tools that have functioned separately so far — User Manager for Domains, Server Manager, Domain Name Server — and provides additional functions beyond these services and tools.

Active Directory is built around Domain Name System (DNS) and lightweight directory access protocol (LDAP) — DNS because it is the standard on the Internet and is familiar, LDAP because most vendors support it. Active Directory clients use DNS and LDAP to locate and access any type of resource on the network. Because these are platform-independent protocols, Unix, Macintosh, and other clients can access resources in the same fashion as Windows clients.

The Microsoft Management Console (MMC) is used to implement and manage Active Directory. The goals of Active Directory are the same as those we identified in the discussion of domain models (Chapter 4). The two most important are

- Users should be able to access resources throughout the domain using a single logon.
- Administrators should be able to centrally manage both users and resources.

Active Directory allows central control and decentralized administration of mixed NT 4.0 and 2000 Server domains. Clients can be 2000 Server workstations and servers, Windows 95, Windows 98, or any other system that has the Active Directory add-on installed.

Because Active Directory is a Microsoft product, most of this discussion focuses on 2000 Server’s implementation of Active Directory. Where applicable, we include information about how Unix can integrate with Active Directory.

FUNDAMENTALS OF ACTIVE DIRECTORY

In the world of Active Directory, clients and servers interact in the following manner:

1. If a client wants to access a service or a resource, it does so using the resource’s Active Directory name. To locate the resource, the client sends a standard DNS query to a dynamic DNS server by parsing the Active Directory name and sending the DNS part of the name as a query to the dynamic DNS server.
2. The dynamic DNS server provides the network address of the domain controller responsible for the name. This is similar to the way static DNS currently operates — it provides an IP address in response to a name query.
3. The client receives the domain controller’s address and uses it to make an LDAP query to the domain controller. The LDAP query finds the address of the system that has the resource or service that the client requires.
4. The domain controller responds with the requested information. The client accepts this information.
5. The client uses the protocols and standards that the resource or service requires and interacts with the server providing the resource.

Benefits of Using Active Directory in an Enterprise Environment

Active Directory is one of the main features that distinguishes 2000 Server from NT 4.0 and previous versions. This newest implementation of the directory services is a response to the often-stated, and possibly warranted, criticism that NT is not designed to be an enterprise-wide solution. In an enterprise environment, Active Directory provides the following benefits:

- More fine-grained administration is possible. Instead of having many administrators with sweeping and widespread rights over all directories (as in User Manager for Domains), you
may have administrators who have a great deal of authority over a particular directory or group of directories but few, if any, rights over other directories. Rights can be granted down to the attribute (object property) level.

- By using the global catalog, you can query various attributes of objects. For example, a particular object — a user name — can be located by querying one of its attributes — say, last name.
- Global groups and local groups have gone the way of the dodo bird (in a pure 2000 Server environment). Instead, you can create nested groups that can contain many levels of users with various individual rights and privileges.
- You can create new object types. Standard object definitions include users, groups, computers, domains, organizational units, and security policies. Ten million objects per domain are allowed.
- Trusts are, by default, transitive. If domain A trusts domain B and domain B trusts domain C, then domain A will trust domain C. However, as an administrator, you can deliberately not allow domain A to trust domain C or allow a one-way trust only.
- Kerberos security is implemented for network authentication, which allows for greater security than the clear text or encrypted logons previously used.
- Fault tolerance is greater. Each controller maintains a copy of the directory database and the replication topology is in a ring structure so that there are always two possible paths for replication.
- Active Directory’s Class Store and Group Policy Editor (GPE) let users access and download applications to which they are entitled, regardless which machine they are sitting at. Active Directory’s Microsoft Installer (MSI) lets developers package applications for use with Active Directory.
- A domain controller can be moved to another site or to another domain without having to reinstall 2000 Server.

Active Directory’s beauty is that it can scale up or down and functions equally well providing simple directory services or more complex levels of administration. Besides supporting LDAP, Active Directory supports HTTP.

The next section focuses on the structure of Active Directory and how it differs from NT 4.0 Directory Services.

Logical Domain Structure of Active Directory
Active Directory is best understood from bottom to top. As an organization becomes larger and more complex, bottom-level units can be joined to make higher-level units. For example, domains can be joined in a hierarchical way to make domain trees, and domain trees can be joined with trusts to make domain forests. We look at the hierarchy of units (from bottom to top) in the following sections.

Simple Objects
Simple objects include computers, groups, users, security policies, and user-defined objects. Objects have attributes, some of which are mandatory and some of which are optional. To view objects in 2000 Server Active Directory, click Start and select Programs, Administrative Tools, Directory Management. Then select Advanced on the View menu to bring up a window similar to Figure 7.1.

In Figure 7.1, note that seven objects are under the domain object, acernt5dom, and eight objects are under the highlighted object Builtin. To find the properties and attributes of any object, simply highlight and right-click the object and select Properties. The properties of Builtin are shown in Figure 7.2.

As Figure 7.2 shows, the object pathname is acernt5dom.foretell.ca/Builtin, and the object class is builtinDomain. The creation date is 9/4/98 5:25:24 p.m. (the time and date that Active Directory for 2000 Server was configured), and the modification date is 9/14/98 4:35:37 P.M. (the last time that we looked at Builtin’s subdirectories). USN stands for Update Sequence Number; Active Directory uses it to keep track of the most recent copy of an object. (USNs are covered in more detail in “Replication” later in this chapter.)

Organizational Units
Organizational units (OUs) are a new object type within 2000 Server’s Active Directory. They
are designed to reduce the number of domains in an organization. OUs are often used to replace domains and subdomains on systems migrating to Active Directory. Under NT 4.0, different departments in an organization are often structured as separate domains, but using Active Directory, these domains can and should be restructured as OUs, thereby flattening the domain structure.

OUs can also be nested, so that a section within a department can have its own OU within the department OU. Objects within OUs must be contained within one domain.

To create an OU within 2000 Server Active Directory, open the Active Directory Manager, Select View, Advanced Features, then select Domain Controllers, Action, New, Organizational Unit. Type the name of the OU as shown in Figure 7.3. Click OK to create the OU object.

**Special Note:** Note that we've appended the domain name, acernt5dom.foretell.ca, to the OU name testorg. This procedure is in keeping with DNS naming conventions and is normal practice when creating OUs. (An OU that is a subunit of another OU would append the name of the OU parent.)

To configure the properties of this object, right-click the newly created object in the right pane of the Active Directory Manager window. A dialog box similar to Figure 7.4 appears.

As the OU's creator, you can add information about the unit. By switching to the Managed By tab, you can also specify the person who will manage the OU.

Switch to the Security tab to see that default permissions have been given to certain groups. Anyone with permission has complete discretion to change these permissions, down to the property level, to fit the organization. Configuring object security is covered in more detail in the “Access Rights — Delegation and Distribution” later in this chapter.

**Special Note:** An important point to note is that the DNS namespace doesn’t show OUs — domains are the smallest units recognized within the DNS Manager. See “Microsoft DNS and Active Directory” later in this chapter for further information.

**Domains**

Domains group network objects and OUs into a unit with a security boundary. By default, security policies and settings don’t flow from one domain to another.

**Domain Trees**

Domain trees are a hierarchical way to group domains. All domains have, by default, a two-way trust arrangement with all other domains. The domains share a hierarchical naming structure with the child domain name appended to the parent name in the fully-qualified domain name.

**Domain Forests**

Domain forests are groups of domain trees. The domain trees within the domain forest don’t share a naming structure, but a two-way transitive trust is created among the root (top-level) domains in each domain tree. Because the domains within the domain trees are all joined with two-way trusts, in effect, resources become available to any user within the domain forest. Domain trees within the domain forest also share a global catalog.

The domain forest structure is particularly useful in this era of company mergers. If one company buys another company or if two companies decide to merge, an administrator doesn’t have to spend hours renaming and restructuring the domains within an organization. By typing DCPromo from the Run command on a 2000 Server and running the wizard, the administrator has the option of joining an existing domain to another domain to form a domain tree or joining an existing domain tree to another to form a domain forest. To see the domains, domain trees, and domain forests in the Active Directory namespace, click Start and select Programs, Administrative Tools, Domain Tree Management.
Physical Domain Structure
The logical domain structure and the physical domain structure don’t have to be the same. Active Directory lets you create sites to mirror the physical domain structure. Computers linked with high-speed, reliable network links can be grouped in one site, while computers linked with lower speed lines can be partitioned into separate sites.

Suppose that you have a department that contains several local subdepartments and several remote subdepartments connected with a slow link. A good logical partitioning would have the department as a domain and each subdepartment represented by an OU. The physical partitioning, however, would group the local domain and local OUs into one site and leave each remote subdepartment as its own separate site.

The physical domain structure is managed with the Active Directory Sites and Services Manager, which you access by clicking Programs, Administrative Tools. Figure 7.5 shows a site that has been automatically configured on a 2000 Server domain controller.

The site has the default name Default-First-Site-Name. To create a new site, highlight Sites and select Action, New Site. Type the new site name (the name must not include spaces), as shown in Figure 7.6.

Highlight a site link name — in this case, there’s only one choice, DEFAULTIPSITELINK — then click OK. You’ve created a new site.

Replication
Active Directory renders obsolete the notion of primary and backup domain controllers. With Active Directory, each controller has a copy of the directory and any controller can initiate and replicate changes. To avoid conflicts, each controller maintains a table of update sequence numbers (USNs) in its database. The tables identify the latest copy of the directory received from every other controller.

Replication can occur at several levels:

- Within a domain
- Within a site (intrasite)
- Between sites

Replication within a domain is automatic and transparent to the administrator. USNs determine the most recent copy of the directory database, and timestamps are used only to break a tie.

You can configure intrasite replication by selecting the NTDS settings of a server (domain controller) as shown in Figure 7.7.

1. Right-click NTDS Settings, then select New NTDS Connection from the drop-down menu.
2. From the list of all the domain controllers within the site, select the one to which you wish to replicate, then click OK.
3. Name the connection in the next box — it’s a good idea to use a naming scheme that involves the name of the other domain controller — and click OK. The new connection appears in the right pane of the Active Directory Sites and Services Manager.
4. A default replication schedule is set automatically. To change the schedule, highlight the connector, and select Properties, Schedule to see a window similar to Figure 7.8.
5. To change the replication frequency for a time period, highlight the time period, then select one of the four options — None, Once per Hour, Twice per Hour, or Four Times per Hour. As an administrator, you may choose to replicate data between nonvolatile sites (sites where changes are infrequent) less often than the default of four times per hour. On the other hand, if changes on the site are frequent and bandwidth is plentiful, two to four times per hour may be appropriate. You may also stagger peak replication times for different connections — for example, making one connection replicate frequently between noon and 4:00 P.M. and another connection replicate frequently between 4:00 P.M. and 8:00 P.M. The possibilities are endless.
**Special Note:** Replication doesn’t automatically take place between domains that are not part of the same site, even if they belong to the same domain tree or domain forest. Replication between sites takes place automatically only under the following two conditions:

- A second domain controller is installed at site A and then is moved to site B.
- A third domain controller is installed at site B, which creates a replication link between the first domain controller at site A and the third domain controller at site B.

Replications are tracked by the *up-to-date vector*, a number that represents the number of originating writes received from each controller within the domain. *(An originating write is a change that has not been replicated, but has originated at the controller itself.)* Each controller keeps a copy of the up-to-date vector within its database, along with the USN.

To provide fault tolerance, interdomain replication (within a site) is configured in a ring topology, ensuring there are two possible paths by which a domain controller can receive its directory database. A mechanism known as *propagation dampening* is used to cut down on the amount of network traffic this topology could generate. To understand this mechanism, let’s consider an example.

Let’s say you have three controllers — Server A, Server B, and Server C. On Server A, you make three password changes; the USN for this property change is 3 and because these property changes originated from the server itself, it is considered an originating write. The up-to-date vector count is therefore 3.

Let’s say, that Server A now replicates to Server C. Server C’s USN for the property change (password change on Server A) is incremented by 1. *(Let’s say that originally this USN was 0, so now it is 1.)* However, because the up-to-date vector for this property change is 3 on Server A, the up-to-date vector for the same property change becomes 3 on Server C. The following table summarizes this first replication:

<table>
<thead>
<tr>
<th>Property change</th>
<th>Server A</th>
<th>Server B</th>
<th>Server C</th>
</tr>
</thead>
<tbody>
<tr>
<td>USN</td>
<td>Password change Server A</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Up-to-date vector</td>
<td>Password change Server A</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

Let’s say we now have two more password changes on Server A — the USN becomes 5 while the up-to-date vector becomes 5 as well. A replication now occurs between Server A and Server B. Server B’s USN for this property change (password change on server A) becomes 1 (assuming again that the USN was 0 before). The up-to-date vector, however, becomes 5 for this property change because it is 5 on Server A. The following table summarizes this second replication:

<table>
<thead>
<tr>
<th>Property change</th>
<th>Server A</th>
<th>Server B</th>
<th>Server C</th>
</tr>
</thead>
<tbody>
<tr>
<td>USN</td>
<td>Password change Server A</td>
<td>5</td>
<td>0 —&gt;1</td>
</tr>
<tr>
<td>Up-to-date vector</td>
<td>Password change Server A</td>
<td>5</td>
<td>0 —&gt;5</td>
</tr>
</tbody>
</table>

Now another replication occurs between Server A and Server C. Server C’s USN becomes 2, while the up-to-date vector for the property change becomes 5. The following table summarizes this third replication:

<table>
<thead>
<tr>
<th>Property change</th>
<th>Server A</th>
<th>Server B</th>
<th>Server C</th>
</tr>
</thead>
</table>
Finally, replication occurs between Server C and Server B — Server C’s USN is greater — 2 rather than 1; but because the up-to-date vector is the same (5), replication of that property change (again, password change on Server A) doesn’t occur. However, Server B’s USN is incremented by 1 (which makes it 2). The following table summarizes this last replication:

<table>
<thead>
<tr>
<th></th>
<th>Property change</th>
<th>Server A</th>
<th>Server B</th>
<th>Server C</th>
</tr>
</thead>
<tbody>
<tr>
<td>USN</td>
<td>Password change Server A</td>
<td>5</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Up-to-date vector</td>
<td>Password change Server A</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Even though this mechanism seems extraordinarily complex for one property change, it reduces a large amount of network traffic when you consider the number of property changes that can occur frequently on an enterprise-level network.

Because domain controllers in 2000 Server are peers, there is always the possibility of collisions, which occur when the same object or item is changed at the same time from two different domain controllers. To avoid this, Active Directory uses property version numbers. When a property on an object is changed, the property version number is incremented. The server to which the change is being replicated looks at the property version number in its local database and makes the change only if its locally stored property version number for that property is lower than the new property version number. If the property version numbers are the same, replication of this property occurs only if the timestamp of the update is later than the timestamp of the existing property version number.

**ACTIVE DIRECTORY SNAP-INS AND RELATED SERVICES**

Active Directory on the 2000 Server platform interfaces with so many services that it's difficult to describe Active Directory in isolation. In many ways, Active Directory is the 2000 Server platform or at least the service with which most, if not all, other services communicate. In fact, unlike NT 4.0, 2000 Server is installed first as a member server (if it’s being installed on a brand new partition) and then is made a domain controller by configuring Active Directory. Realistically, it is impossible to have a 2000 Server network without Active Directory. Although you could still use the server as a standalone, what would be the point?

**Microsoft Management Console (MMC) Snap-Ins**

Below, we describe three of the MMC snap-ins that are central to Active Directory management: Active Directory Manager, Active Directory Tree Manager, and Active Directory Sites and Services Manager.

**Active Directory Manager**

Active Directory Manager is the primary interface for Active Directory. In some ways this snap-in is NT 4.0’s User Manager for Domains and Server Manager plus a whole lot more. To open this snap-in, perform the following steps:

1. Click Start and select Programs, Administrative Tools, Directory Management.
2. Highlight Active Directory Manager at the top of the tree and select View, Advanced Features. An interface similar to Figure 7.9 appears.

You see several subdirectories under acernt5dom, which is an object of the domain class:

- Built-in — This subdirectory of the object class builtinDomain contains the usual NT 4.0 local groups such as Account Operators, Administrators, and Printer Operators, but now they are
called security groups. These groups are automatically added when Active Directory is configured.

- Computers — This subdirectory contains client computers or workstations on the domain.
- ForeignSecurityPrincipals — This subdirectory is essentially an object container created for storing security principal objects within a trusted domain.
- System — This subdirectory contains several services particular to the system and Microsoft DNS (see Figure 7.10). It also contains additional configurable objects, such as Default Domain Policy, Policies, and IP Security, which a domain administrator may use to configure policies particular to the domain. The System directory is of primary importance to you as an administrator.
- Users — This subdirectory contains the normal default users, such as Administrator and guest accounts, as well as some NT 4.0 global and local groups (renamed “Security Group — Global” and “Security Group — Domain Local”) and other groups, such as Certificate Publishers, that are new.
- Domain Controllers — This subdirectory, by default, contains the name of the domain controller(s) for the domain.

**Active Directory Tree Manager**
The Active Directory Tree Manager snap-in becomes important when your network has multiple domains arranged in a domain tree. To view this snap-in, select Administrative Tools, Domain Tree Management. A window similar to Figure 7.11 appears.

Figure 7.11 shows only one domain — acernt5dom.foretell.ca — of object type domainDNS. Active Directory uses DNS as its locator service; therefore, DNS must be present in the domain.

If you right-click the domainDNS object and select Properties, you will see a dialog box similar to Figure 7.12.

In this dialog box, you perform the final transformation from mixed mode to native 2000 Server/Active Directory mode (if and when your system is entirely upgraded to 2000 Server).

**Caution:** As the warning in the lower portion of Figure 7.12 indicates, you can’t reverse the transformation from mixed mode to native mode. Once you have changed to a 2000 Server network, you can’t go back and install NT 4.0 computers or clients that aren’t enabled for Active Directory.

To manage individual domains in the domain tree, follow these steps:

1. Highlight the domain in the Active Directory Tree Manager.
2. Select Action, Manage to open the Active Directory Manager.
3. Click Advanced on the Action menu to open a window similar to that shown Figure 7.9.

**Active Directory Sites and Services Manager**
Using the Active Directory Sites and Services Manager snap-in, it’s possible to create new sites and move domain controllers from one site to another. The Active Directory Sites and Services Manager also contains a Services subdirectory, with sub-subdirectories such as RAS, Windows NT, and Eap Entries. You can alter the security properties on these services. After selecting the desired service, you have two options. You can select Delegate Control from the Action menu and select the objects within the container for which certain user groups have permissions. Or you can select Properties, Security and assign permissions to objects or the properties of objects within the container.

**Microsoft DNS and Active Directory**
A discussion of Active Directory isn’t complete without a discussion of DNS because Active Directory uses DNS as its locator service and can’t function without it. Accordingly, one of the most worthwhile things you can do before migrating to Active Directory is to standardize your network on DNS and rename your NT domains to correspond to DNS names.

To solve name resolution queries, Active Directory uses dynamic DNS, a new form of DNS that maintains and updates zone files dynamically. (Dynamic DNS is a powerful new service
Depending on your requirements, there are many different ways of configuring the Active Directory domains and domain trees to use DNS. For example, a domain namespace can be contained in one DNS zone or in many zones. Generally speaking, a decentralized organization will have more zones, while a centralized organization will have fewer zones. More zones in a decentralized organization reduces the amount of traffic because DNS queries for a particular domain will be accessing a smaller zone file, which would ideally be located geographically close.

**Special Note:** Because Active Directory won’t work without DNS, when you configure your 2000 Server machine as a domain controller, you are asked whether you want to install the DNS service if a DNS server can’t be located. The release notes for 2000 Server suggest that it’s preferable to install the DNS service before upgrading your machine to a domain controller rather than selecting Y when prompted to install DNS. We chose to install DNS while configuring the machine as a domain controller — so far with no ill effect!

To add an Active Directory-integrated DNS zone to a domain, perform the following steps:

1. Open Microsoft DNS Manager.
2. Highlight Forward Lookup Zone or Reverse Lookup Zone and right-click.
3. Select Create a New Zone to open the wizard.
4. Select Active Directory Integrated so that the zone will show up in the Directory Manager snap-in. Note that to be integrated with Active Directory, the zone must be a primary zone.
5. Click Next, then Finish. An example of a forward lookup zone created in DNS Manager is shown in Figure 7.13.

The same zone, acernt5do1.foretell.ca, is shown within the Active Directory Manager in Figure 7.14.

**Querying Active Directory**

Active Directory queries are resolved in two ways:

- Using the global catalog, which contains information about the entire domain forest of domain trees
- Searching the domain tree from which the user is performing the search

The method used depends on the object attribute for which the user is searching. To avoid becoming too large and unwieldy, the global catalog retains only some of the attributes of objects, even though all objects themselves are indexed.

If the attribute upon which the query is based is present in the global catalog, the entire domain forest may be searched. If the attribute is not present, only the domain tree from which the search has been initiated may be searched. Therefore, it is prudent to limit your network to one domain tree if at all possible.

Performing searches within Active Directory is simple. First, access the Active Directory Manager, then select a subdirectory (for our example, Users). Right-click Users, then select Find from the drop-down menu. A dialog box similar to Figure 7.15 appears.

As you can see, you can perform two types of searches — “Users, Contacts and Groups” and Advanced. Figure 7.16 shows the results of a search for the user named Administrator within the domain acernt5dom.

More complex, SQL-type searches are possible using the Advanced tab. For example, Figure 7.17 shows the search for a group that begins with the letter A but isn’t Administrators.

The Find screen is very versatile — whichever kind of search you choose, you can browse for the container in which you want to perform the search and select whether you wish to search the domain or the entire directory. Interestingly, it isn’t possible to locate ordinary files using
this method. To locate files, you use Windows Explorer, as you do in earlier releases of NT.

Security and Active Directory
Security policy is much more flexible under 2000 Server than it is on previous versions of NT. It’s possible to have different security policies for different domain controllers. It’s also possible to define security policies that apply to the whole domain tree and not just the domain.

Policy Storage
Under 2000 Server, domain policy is no longer stored in the SAM, and local policy is no longer stored in the LSA. Instead, these policies are stored in Active Directory under Domain Policy Objects and Local Policy Objects, respectively. To see the policy objects for a domain, from the Active Directory Manager Advanced view select System, Policies. Domain Policy Objects store the following information:

- Password policy — sets password policy for user accounts
- Lockout policy — enables or disables a lockout policy for user accounts
- Security descriptor for domain operations — specifies who can change Domain Policy property settings
- Kerberos logon policy
- Encrypting file system data recovery policy
- Listing of the trusted Certificate Authorities

Local Policy Objects store the following information:

- Auditing policy
- Privilege policy or Administrative roles — specifies which privileges are associated with which accounts
- Local roles — specifies which accounts receive domain SIDs in their token
- User rights policy — sets local user and group rights

By default, all machines in a domain use the default local policy on the domain controller unless they have their own local policy.

Logon and Authentication
Three authentication protocols are supported in 2000 Server — Kerberos version 5, SSL/TLS (Public key authentication), and Windows NT Lan Manager. The three protocols are supported by the Security Support Provider Interface (SSPI) API.

Kerberos 5 is the primary protocol for secure authentication. It is installed by default as a service on 2000 Server domain controllers, and it is installed by default as a client on 2000 Server member servers and workstations. The advantages of Kerberos are many:

- To log onto a remote server within a domain, a 2000 Server server doesn’t have to requery Kerberos. It can perform its own authentication procedure instead.
- One server can authenticate to another server on behalf of a client. For example, a server located in one domain can send security credentials for a domain client to a server in another domain.
- Kerberos 5 is used in many platforms, including some Unix platforms. 2000 Server’s version of Kerberos complies with RFCs 1510 and 1964.
- Both client and server are authenticated during a Kerberos session.

Under 2000 Server, a domain logon using Kerberos takes place as follows:

1. C1, the client program, contacts the Local Security Authority Sub System (LSASS) using the winlogon sequence (Ctrl+Alt+Delete). (LSASS and the SSPI are integrated.)
2. Password and identification (the session key) are sent from LSASS to the Kerberos authentication library.
3. The library requests a ticket-granting ticket (TGT) from the nearest Active Directory Key distribution center (KDC). The TGT is encrypted with the hashed password from step 2.
4. If the request is granted, a TGT is returned to the library. The library uses that ticket to obtain a ticket to the client workstation at which the user is sitting and then extracts SIDs from the ticket to pass back to the LSASS.

5. LSASS then creates the access token, which includes SIDs and privileges. All processes that the user executes use this access token as in NT 4.0 and earlier.

6. If the client wishes to access a network service, Client Runtime checks the ticket cache in the library for a ticket to that server. If found, the ticket is sent to the server. If not found, C1 sends the original TGT (obtained from the login process) back to the KDC and receives in return a new ticket. This ticket is appended with the session key and stored in the KDC for future use.

Local logon takes place using the older mechanism, MSV1_0, also known as NTLM.

Kerberos is not, by itself, a public key authenticator. Instead, it is a three-sided shared-secret authenticator. The three sides are the client (C1), the KDC, and the server (S1). Microsoft plans to expand Kerberos to allow public/private key encryption. Under this model, the client (user) may obtain a ticket only if it has an X.509 version 3 certificate stored in its own User Object within the Active Directory. The user requests a ticket by querying the Active Directory for a public key certificate (by using his or her private key). If Kerberos locates the public key certificate, the Kerberos service issues one ticket for the user, and that ticket is used to issue the initial ticket granting ticket (TGT) described above.

When the client wishes to access another server on the same domain, the server can verify the client ticket without accessing Kerberos. Kerberos gives the server encryption key to the new server, which the server can use to decrypt the ticket. When the client wishes to access another server outside the domain, the authenticating servers contact the Kerberos service on that domain on behalf of the client using the following steps:

1. Client requests access to a resource in another domain.
2. Kerberos service in the requesting domain issues a referral ticket for the new domain to the client.
3. The Kerberos client sends the ticket to the Kerberos service on the new domain.
4. The Kerberos service on the new domain issues a new ticket valid for the new domain but identifying the requesting domain.
5. If the original client is a non-Windows client hosting Kerberos version 5, it can be authenticated by a 2000 Server, but some values may be lost. If this happens, the Kerberos service still attempts to match the name on the ticket with an NT user account or a specialized account set up for this purpose.

For Internet connection and smart cards, the SSL/TLS protocol is used. Based on public key authentication, X.509 version 3 certificates are either given by a Certificate authority such as Verisign or issued by the network’s own certificate server.

Special Note: A certificate consists of a key pair — a public key and a private key. Each key can decrypt messages encrypted by the other. Normally, the mechanism is as follows:

1. The message is encrypted with the sender’s (i.e., the logged-on user’s) private key to create a digitally signed message.
2. To make sure that the receiver is legitimate, the original message must also be encrypted with the receiver’s public key (previously sent to the trusted sender). Then only the receiver, who holds the private key, is able to decrypt the message.
3. Once the receiver receives the message, it decrypts the public key wrapper using its own private key and decrypts the private key wrapper using the trusted sender’s public key.

For machines running NT 4.0 or earlier, Windows NT LAN Manager was the authentication protocol. It will continue to provide authentication to or from computers running these earlier releases as long as they are present in the network.

Access Rights — Delegation and Distribution
Under 2000 Server, Active Directory stores not only system policy information but also all
security information related to the objects contained within Active Directory. The security information, which is contained within a property known as the security descriptor, includes:

- The object's owner
- Group memberships
- Access Control List (ACL)

**Special Note:** All objects, including attribute objects themselves, must have a security descriptor. The security descriptor is an object represented by the OID (object identifier) 2.5.5.15. OIDs are discussed further in “Object Identifier” later in this chapter.

Within the ACL are Access Control Entries, each containing the security principal (which is a user, group, or computer), and the type of access allowed. The ACL can contain both ACEs that refer to the whole object and ACEs that refer only to the properties of the object. In this way, permissions and rights can be assigned on a more refined (even property-by-property) basis.

Under 2000 Server, child objects within the domain inherit access permissions from their parent objects unless otherwise configured. This inheritance can be both a blessing and a curse. Alistair G. Lowe-Norris makes a very important point regarding inheritance in his article “Managing Permissions for NT 5.0’s Active Directory” (Windows NT Magazine, July 1998). In the article, he explains that if you deny read access to a container object (let's say an OU) for a particular group of users and accidentally leave the “Apply these permissions down the tree” box checked (the default), all objects in this OU aren't visible to this group. To undo the damage, each OU must be examined to reinstate this permission — a daunting task if the OU contains many objects.

**Special Note:** Users and groups are the only non-container objects for which it is possible to assign permissions.

To see the rights and permissions for a particular object, open the Active Directory Service Manager within 2000 Server, select the object (let's say a domain) from the tree in the left pane, then select Properties from the drop-down menu. In the Properties dialog box, switch to the Security tab, as shown in Figure 7.18.

Highlight the group or user for which you wish to see permissions. Several permissions show up on this page; you can see the rest by clicking Advanced. Figure 7.19 shows the permissions you can set using the Advanced button.

To see further information, highlight the group or user on this page — let's use ForeTell\Authenticated Users — then click View/Edit. The permission property dialog box shown in Figure 7.20 appears, showing two tabs, Object and Properties.

The Object tab displays permissions for the group or user on the object itself. The Properties tab displays permissions for the group or user on the properties of the object (and any child objects). If these permissions are changed, the changes are, by default, propagated down the tree (but within the domain).

Like NTFS permissions and user rights under NT 4.0, permissions under Active Directory are cumulative, except for the Deny permission (like No access under NT 4.0), which overrides other permissions. If a user belongs to a group that has permission on a resource and the user is nested within a group that doesn't have permission on the same resource, the user is denied access to that resource.

**MIGRATING TO WINDOWS 2000 SERVER**

In the following sections, we look first at the tasks you should complete before you migrate...
from a earlier version of Windows or NT to 2000 Server. Then, we look at the migration process itself for a single domain model, a master and multiple-master/resource domain model, and a complete trust model.

Before the Migration
Preparing for 2000 Server is primarily the task of preparing for Active Directory. Sean Daily’s excellent article, “10 Steps to Prepare for NT 5.0 Now” (Windows NT Magazine, February 1998), contains a great deal of useful information about preparing your organization for 2000 Server and the Active Directory. We touch on the highlights here.

Domain Flattening
One of the first suggestions is to flatten your organization — into one NT domain if possible. Many of the reasons for having more than one NT domain, such as the master domain/resource domain model, in which the resource domain trusts the master domain but the master domain trusts no other domains, disappears under 2000 Server. For example, instead of having two domains — a master and a resource domain — you could reconfigure the domains into OUs with exactly the same rights and permissions as those in your domain model.

Of course, flattening your organization domain structure during the period just before receiving 2000 Server is a little easier said than done. Recall from Chapters 3 and 4, that under NT 4.0 a PDC or BDC can’t be reconfigured as a PDC or BDC for another domain. NT must be completely reinstalled to make the change. Member servers and NT workstations can be transferred to a new domain, but even that is a headache if you have more than a few workstations (and fewer administrators than you need).

Special Note: There are tools to facilitate the migration process. One such tool is the Phoenix Domain Reconfiguration Tool, available from FastLane Technologies. The documentation for the Phoenix Tool describes two main tools. One, called Phoenix Domain Reconfiguration tool, is essentially a manual migration; the other, called DR Distributor, performs a scheduled automated migration. This tool also comes with a bundled resource kit consisting of NT Reporter, AdminChecker, DC Mover, Exchange Updater, and Copy Password, all of which are designed to assist in your migration. This product works on NT networks 3.51 and beyond. For more information, see www.fastlanetech.com.

Standardizing on TCP/IP and DNS
As we mentioned before, Active Directory uses DNS as its locator service, and where there’s DNS, there must be TCP/IP. Ideally, TCP/IP should be your only protocol, but if that’s impossible, it should at least be the default. Your organization should aim to eliminate NetBIOS applications completely — another task that’s easier said than done. Fortunately, until the network has completely migrated to 2000 Server and Active Directory, it is still possible to support NetBIOS applications. Until your migration is complete, however, administrators must continue to use LMHOSTS and/or WINS to support those applications that use NetBIOS.

Object Identifier
Active Directory has an extensible schema. This means that as an administrator, with appropriate permissions, you may create new object classes and attributes specific to your organization. To do so, however, you need an Object Identifier (OID). An OID is equivalent in function to a CLSID (or GUID) for an ActiveX control. Each one must be unique, and therefore they must be issued by a standards organization — in the United States, it’s the American Nation Standards Institute (ANSI). Organizations receive (for a fee) a root OID that they then divide into branch IDs — a process similar to receiving a network number and subnet mask that a company uses to assign IP addresses. More information about fee schedules and registration can be found at www.ansi.org/public/services/reg_org.html

An example root OID given by an issuing authority is 1.2.3.4. The organization then divides the OID by appending a period and any number from 0 through $2^{28} - 1$ to this number. For example, 1.2.3.4.5 could be the number that the organization assigns to a division within its
company. This division could then append any number from 0 through $2^{28-1}$ to 1.2.3.4.5 (such as 1.2.3.4.5.1) to create a root OID for objects that it creates. Any objects created by this division would then have an OID 1.2.3.4.5.1.x, where x is from 0 through $2^{28-1}$.

It's quite possible that your company has no desire whatsoever to create new objects. Even so, it's a good idea to get an OID just in case.

PC Upgrades
Migrating to 2000 Server and Active Directory has a downside — more memory. To install 2000 Server on a server, Microsoft recommends a Pentium Pro with at least 128 MB of RAM and a 2 GB hard drive. The upside is that all workstations and servers don’t have to migrate immediately, but some features won’t be present until all computers are enabled for the Active Directory. (These missing features are discussed further in “Mixed Environments” later in this chapter.)

Clean up SID Duplication
Under normal circumstances, all security identifiers used to identify objects in an NT 4.0 domain are unique. However, in “10 Steps to Prepare for NT 5.0 Now,” Sean Daily mentions that some disk duplication utilities that clone NT systems can cause duplicate SIDs.

The best medicine is prevention. Don’t practice cloning if you’re planning to migrate to 2000 Server and Active Directory. However, if you’ve cloned in the past, it’s possible to use a product like SidChanger from PowerQuest (www.powerquest.com), GHOSTWalker from Ghost Software (www.ghostsoft.com), or NTSID freeware by Mark Russinovich and Bryce Cogswell (www.ntinternals.com/ntutil.htm) to reassign unique SID numbers to each object (computer, group, or user) in an NT network.

Migration
Each domain model has its own migration issues. Below we look at the most important ones for each model.

Single Domain
For many organizations, the single domain model is more than sufficient. If your network is currently configured as one NT domain, your job is basically complete. Once you receive 2000 Server, you may wish to subdivide your organization into OUs to provide more detailed control of your rights and permissions.

You can perform a single domain migration to 2000 Server and Active Directory (according to the Microsoft white paper Planning and Migrating to Active Directory) in the two ways:

- Upgrade the PDC to 2000 Server first. Install Active Directory on this PDC, then install 2000 Server and Active Directory on at least one other domain controller (for fault tolerance). Upgrade other servers and workstations to 2000 Server and/or the Active Directory client software. To add an extra level of security, before migrating, synchronize one BDC with a PDC, then take the BDC off line. If problems occur during migration, you can bring the BDC back on line and promote it to an NT 4.0 PDC. You can then start again.

- Take the PDC off line, bring it into a Lab and build the 2000 Server/Active Directory network from there, adding workstations and servers as the configuration stabilizes. Until you tell the system that the migration to 2000 Server is complete, there is backward compatibility; NT 4.0 clients can continue to be added.

Master and Multiple-Master/Resource Domain
As we mentioned earlier, OUs can replace many of the functions of domains. Where possible, you should group the OUs into one domain for the following reasons:

- Fewer domains make for a lighter administrative burden.
- Querying is faster and more efficient. The global catalog can be used only for those attributes that are kept within the global catalog database. Therefore, if you are searching for an object located in another domain tree within the same domain forest and the attribute is not in the global catalog, it won’t be found.
Sometimes the reason for a multiple-master/resource domain arrangement is that the number of users within one NT 4.0 domain is limited to approximately 20,000 users. Active Directory, on the other hand, supports up to 10 million objects (which may be computers, users, groups, or OUs).

Microsoft’s white paper entitled Planning and Migrating to Active Directory contains a step-by-step approach to migrating from a master/resource model to a single domain with OUs. The following procedure is based on the section “Merging Second-Tier Domains into Master Domains.”

1. Migrate the Master Domain to 2000 Server and install Active Directory on at least one PDC in this domain.
2. Create OUs based on the existing domain structure. The master domain will become one OU, while the resource domain will become another OU. Further divide these OUs if desired.
3. Upgrade resource domain PDCs to 2000 Server and Active Directory. Don’t move these PDCs from the resource domain yet.
4. Move member servers from the resource domains to OUs in the master domain. The administrator of the master domain can use both NetBIOS and Active Directory publishing to advertise a resource’s existence.
5. Move the resource domain local groups to the master domain. Normally, permissions are not preserved when domain local groups are moved to a new domain. However, Active Directory actually copies security principals that retain the old SID and append the new SID. When the logon service builds the access token, both SIDs are added to the token, thereby ensuring a link between the SID in the ACL and the new SID.
6. Move workstations on the resource domain to the master domain. If the workstations are to participate in Active Directory, they must first be upgraded to 2000 Server (if an NT workstation) or have special client software installed (if Windows 95/98).
7. Delete the resource domain either by turning off domain controllers in the resource domain or by moving them to the master domain. Domain controllers in 2000 Server, unlike those in NT 4.0, can be moved from one domain to another as long as both domains are Active Directory domains (which means that Active Directory is installed on at least one domain controller in each domain).

Complete Trust
The complete trust model, in which a two-way trust is set up among all domains, is the most flexible but also the most difficult to administer. Everyone potentially has access to everything, and this access has advantages and disadvantages. It is up to you as the administrator to set the individual file and directory permissions so that highly confidential information (directors’ salaries, for example) can’t be accessed by all. Of course this is true with any network, but having the one-way trust can sometimes add an extra layer of protection above and beyond user rights and permissions.

Migrating a complete trust domain doesn’t differ greatly from migrating a master/resource domain. If possible, everything should be compiled into one domain or at the very least a domain tree. By pretending that one of the domains is the master domain and the rest are resource domains, the resource domains can be collapsed into the master domain in the same manner as outlined in the previous section.

Mixed Environments
It’s generally the case that the average enterprise networking environment consists of more than one operating system. It’s even more likely, as 2000 Server becomes available, that companies will choose to retain their NT 4.0 servers and workstations and integrate them gradually with the 2000 Server environment.

In the next sections, we focus on the how NT 4.0 and 2000 Server can coexist in a network
and some of the ways in which Unix systems can be made to integrate with the NT Active Directory.

**NT 4.0 and 2000 Server Mixed Environments**

For purposes of discussion, we’ve classified Active Directory installations as pure and mixed. Depending on the classification, certain features will or won’t be present.

- **Pure 2000 Server/Active Directory environments** — All domain controllers have 2000 Server and Active Directory installed. Workstations and clients may or may not be Active-Directory enabled. If they are, they will be able to participate fully in the Active Directory domain. If not, the controllers will appear as NT 4.0 controllers, and the data will appear as a flat store. Once all controllers have been upgraded, you must explicitly designate the network as a pure 2000 Server/Active Directory domain. From that point on, downlevel replication (NT 4.0 and below) ceases, and it’s impossible to add downlevel controllers. All controllers act as peers.

- **Mixed NT 4.0 and 2000 Server/Active Directory environments** — The PDC must have 2000 Server and Active Directory installed. BDCs may have either NT 4.0 or 2000 Server installed. Clients may or may not be Active-Directory enabled. Unfortunately, the mixed NT 4.0/2000 Server environment doesn’t support two important features of Active Directory: multimaster replication and nesting of groups. Multimaster replication lets you change user account information from any domain controller because the controllers act as peers under 2000 Server. A mixed environment still has at least one PDC with one or more BDCs, and user account changes can be made from the PDC only. The group nesting feature lets administrators create smaller groups within larger groups to any level they wish. Furthermore, in 2000 Server, the notion of local groups and global groups is gone. In a mixed environment, however, this system persists.

**Unix and Active Directory**

At the time of this writing, few tools help Unix machines integrate fully with Active Directory. This situation will change with the full release of 2000 Server. Cisco and Microsoft have announced a partnership to develop Active Directory tools that will be available on Unix platforms through the use of Cisco IOS software. The target date for shipping this product is 90 days after the release of the final production version of Active Directory (which would presumably coincide with the final release of 2000 Server.) Further information on this product is available at


Another tool that is used to integrate Active Directory with Unix systems is Active Directory Services Interface (ADSI). Actually, this is the tool that Cisco is using to develop Active Directory Services for Unix. Developers and administrators can use ADSI to write applications in Visual Basic and Visual C that can use information from the Active Directory to interface with Unix computers, among other things. The ADSI kit may be downloaded from

www.microsoft.com/win32dev/netwrk/adsi.htm

**CONCLUSION**

Active Directory, we’ve found, is a moving target. New features and improvements are being added all the time. Ultimately, Active Directory promises to be a tool that all administrators will appreciate, whether they manage a network of ten or several thousand computers.