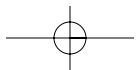
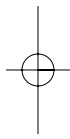
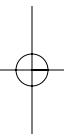


PART II

# Core Skills



## CHAPTER 3

# Managing OUs, Users, and Groups

The most visible part of Active Directory administration is managing objects with the Users and Computers snap-in. This snap-in enables you to create organizational units (OUs) to set up an OU tree in a domain. You also use this snap-in to populate the OU tree by creating objects of the following eight classes in the OUs you want:

- Users (or alternatively, inetOrgPersons, new in AD2003)
- Contacts
- Computers
- Groups
- MSMQ queue aliases (new in AD2003)
- Shared folders
- Printers

This chapter covers managing OUs and the first five classes in the list. We will proceed as follows:

- First, we describe the contents of your Active Directory domain right after installation.
- Second, we explore how to manage OUs and objects of each of the five other classes (i.e., users, inetOrgPersons, contacts, computers, and groups).
- Finally, we discuss some additional features of the Users and Computers snap-in, and we discuss or list additional tools for managing objects.

This chapter focuses on the Users and Computers snap-in. If you have to create many objects, other tools you can use include DSAdd (new in AD2003), LDIFDE, CSVDE, scripting, or some Resource Kit

## 140 Chapter 3 Managing OUs, Users, and Groups

---

tools. DSAdd we cover at the end of this chapter; the other tools we just list there, but cover later in the book.

---

**NOTE** Behind the scenes, a domain object can contain objects of 35 classes (23 in AD2000), and an OU can contain objects of 59 classes (35 in AD2000). However, with the Users and Computers snap-in, you can normally create and see objects of only the nine classes just listed.

---

**NOTE** MSMQ Queue Alias is listed as an alternative when you create new objects in the Users and Computers snap-in. You can, however, create and use such objects only if you first install Microsoft Message Queuing 3.0 (MSMQ). Because such aliases (corresponding to the `msmq-Custom-Recipient` class) reside in the directory, they help applications to locate message queues. Further discussion of the topic is outside the scope of this book. Also, we don't mention MSMQ queue aliases later in this chapter, even though they would appear in some lists, and so on.

---

### Active Directory after Installation

---

After you have created your first domain by installing Active Directory on a server (i.e., promoting it to a domain controller), there are certain users, computers, groups, and containers already in place (see Figure 3.1). You see these objects with the Users and Computers snap-in, which you start by clicking the Start button and selecting Administrative Tools, Active Directory Users and Computers.

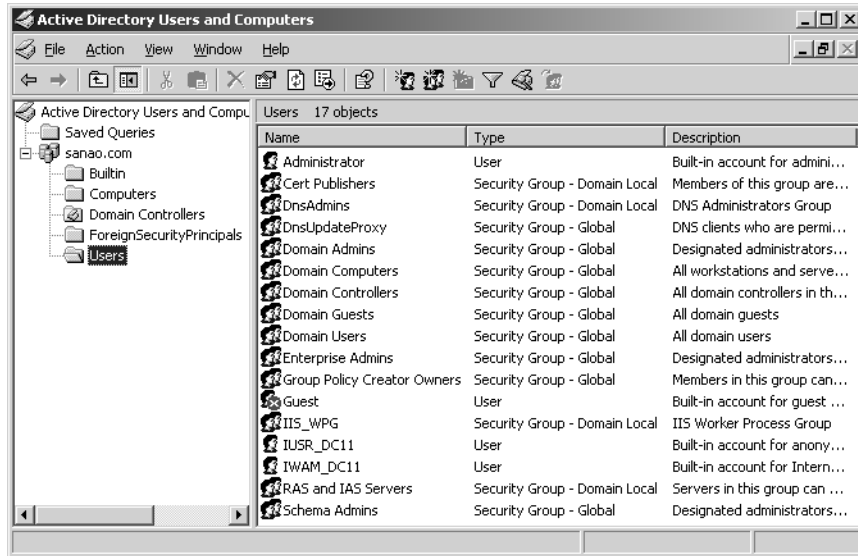
---

**TIP** Another way to start the Users and Computers snap-in is to click the Start button, select Run, type `dsa.msc`, and press Enter.

---

You will see the following predefined objects in the snap-in:

- Five containers, one of which is an OU
- Some user objects (or user accounts)
- Some group objects (sometimes referred to as group accounts)
- One computer object (or computer account) for your domain controller



**Figure 3.1** A newly installed domain (`sanao.com` in the figure), which is the root domain of a forest

**NOTE** Active Directory contains only objects. Users, groups, and computers, however, are often called *accounts* instead of objects. You could also argue that an account is something that can authenticate (user or computer), so a group is not an account, but “just” a group of accounts.

**NOTE** If you upgrade a Windows NT domain, you will see the users, groups, and computers of that domain in Active Directory.

### Predefined OUs and Other Containers

The objects in a domain should reside in containers instead of at the domain level, just as files on disk should reside in folders instead of in the root folder. Accordingly, the predefined objects are stored in containers below the domain level. Table 3.1 describes the five predefined containers. Because they have extra system-flags protection, you cannot rename, move, or delete them—they are always there (unless you redirect some of them, as explained a little later in the section “Redirecting the Users and Computers Containers to OUs”).

**142 Chapter 3 Managing OUs, Users, and Groups****Table 3.1** The Predefined Containers in Active Directory

Container	OU	Purpose	Possible Contents
Builtin	No	This is a container for the predefined built-in local security groups (you cannot create them yourself).	Computer, group, user, inetOrgPerson
Computers	No	This is a default container for computer objects corresponding to Windows NT/2000/XP workstations and member servers in this domain.	Computer, contact, group, printer, user, inetOrgPerson, shared folder
Domain Controllers	Yes	This is a default container for computer objects corresponding to domain controllers of this domain.	Computer, contact, group, OU, printer, user, inetOrgPerson, shared folder
Foreign Security Principals	No	This is a container for placeholders that represent group members from domains external to the forest. This includes well-known security principals, such as Authenticated Users, if they are members of some group in the domain.* Objects in this container are visible only when the snap-in's Advanced Features are turned on.	Computer, contact, group, printer, user, inetOrgPerson, shared folder
Users	No	This is a default container for users and groups.	Computer, contact, group, printer, user, inetOrgPerson, shared folder

\* We discuss well-known security principals in Chapter 4 and foreign security principals in Chapter 6.

---

**NOTE** In Table 3.1, the Possible Contents column lists the object types (that is, classes) that you can create in the corresponding container using the Users and Computers snap-in. With an “under-the-hood” tool, such as ADSI Edit, you could create other types of objects. However, there is no need to use the predefined containers for anything but what is described in the table.

---

You shouldn't use the Builtin container for anything, even though it is possible to create computers, groups, and users in it. Likewise, you

could create users in the Computers container or computers in the Users container, but there is no point in doing so. Putting such things together is comparable to placing your cookbooks and music CDs on the same shelf. It is possible, but why do it?

If you want, you can keep your users in the Users container and computers in the Computers container. If you do so, however, you can neither create OUs in them nor assign Group Policy for them, because these containers are not OUs. If you have more than 20 users, for example, and you want to delegate some administration, you will probably end up creating new OUs for your users and computers (i.e., outside the Users or Computers containers). We will come back to this issue in the “Administering OUs” section later in this chapter.

The Domain Controllers container is an OU, and therefore you can create OUs in it and assign Group Policy(ies) for it. This OU already has a Default Domain Controllers Policy Group Policy object (GPO) assigned, which affects the security and other settings of your domain controllers. You are likely to keep the computer objects for your domain controllers in this container and other OUs that you create below it.

### ***Why These Containers?***

It may seem that the way these predefined containers were chosen is odd. Why are most of them not OUs? Some explanation is given by the fact that these containers ease the upgrade from Windows NT to Active Directory. During the upgrade process, the old user accounts and groups are migrated to the Users container, old workstation and member server accounts are migrated to the Computers container, and old domain controller accounts are migrated to the Domain Controllers container.

In addition to migration, the default Users and Computers containers are also used whenever users, groups, and computers are created using downlevel tools. Such tools are Windows NT User Manager and Server Manager, Net commands (Net User, Net Group, Net Localgroup, Net Computer), Windows Support Tools NetDom Add command without the /OU switch, and any tool that uses the old Windows NT APIs to create these objects. Also, when a Windows NT/2000/XP workstation joins a domain and there is no precreated computer account, the account will be created in the default Computers container.

In Windows NT, built-in local groups were internally stored separately from other groups, users, and computer accounts. This separation was brought over to Active Directory in the form of the Builtin container.

## 144 Chapter 3 Managing OUs, Users, and Groups

---

So why are these three containers not OUs? One explanation could be that this way you are intentionally discouraged from using them in the long run and you must create new OUs instead.

### ***Redirecting the Users and Computers Containers to OUs***

If your domain is on the Windows Server 2003 functional level, you can redirect the predefined Users container and the Computers container to OUs you have created. After the redirection, these new OUs will be the default containers for users, groups, and computers. For example, you can create the Employees OU and the Workstations OU, and then perform the redirection using the following commands:

```
redirusr OU=Employees,DC=Sanao,DC=com  
redircmp OU=Workstations,DC=Sanao,DC=com
```

During the redirection, the system-flags protection is moved from the old default containers to the new ones. Therefore, you could rename or delete the old containers.

The predefined Users container and Computers container don't allow you to apply group policies to them, but if you perform the redirection to a normal OU, you can apply group policies. On the other hand, if your users and computers are already in normal OUs and you don't use any tools that would create new users or computers in the default containers, there is not much advantage in the redirection.

For more information, see the Microsoft Knowledge Base article 324949 at <http://www.microsoft.com>.

### **Predefined Users**

Two user objects are always present: Administrator and Guest. You cannot delete either of them, but you can rename them at will. Renaming Administrator offers some extra protection because a potential network intruder would need to guess the new name in addition to the password. However, if you have a large network and many administrative personnel, it may be confusing for the Administrator account to have a different name.

---

**NOTE** The default permissions of Active Directory allow any user of the forest to see the names of administrative accounts, so renaming them is really minimal "protection." You can think of it as adding a small extra

hurdle in a potential intruder's path. If you chose permissions compatible with pre-Windows 2000 servers, anonymous users can also see this information.

Active Directory has predefined user accounts besides Administrator and Guest, depending on what services are installed. Table 3.2 lists the typical predefined user accounts in Active Directory.

If you enable the Guest account, be careful about the permissions you give to it or the Everyone group (and in AD2000 also to the Domain Users or Users groups). After all, anyone who "walks in the door" can use the Guest account. There are two ways to use the Guest account.

**Table 3.2** The Predefined User Accounts in Active Directory

Name	Present	Description
Administrator	Always (although could be renamed); cannot be disabled in AD2000 but can be disabled in AD2003	The only user account you can use when you log on for the first time. The Administrator account of the first domain in a forest has the widest possible administrative permissions on Active Directory and the domain controllers in the same forest. You can create other user accounts with permissions as wide. The Administrator accounts of the later domains in a forest have the widest possible administrative permissions for their own domains.
Guest	Always (although could be renamed); disabled by default	If someone doesn't have a user account, he can use the Guest account (if the account is enabled). (See the discussion in the text.)
IUSR- _servername	One for each domain controller that has IIS installed	If IIS allows anonymous access (e.g., by Web browsers), anonymous users use permissions of this user account.
IWAM- _servername	One for each domain controller that has IIS installed	IWAM stands for IIS Web Application Manager. The IISWAM. OutofProcess Pool component (part of IIS) uses this user account. <i>(continued)</i>

**146** Chapter 3 Managing OUs, Users, and Groups**Table 3.2** The Predefined User Accounts in Active Directory (*cont.*)

Name	Present	Description
krbtgt	Always; disabled by default; hidden by default; cannot be enabled or renamed	The Kerberos key distribution center (KDC) uses this account. “Krbtgt” is part of the KDC’s service principal name (SPN). Also, a symmetric key is derived from the password of krbtgt, and this key is used to encrypt and decrypt TGTs. Only the KDC knows this password and it changes the password periodically.
TsInternetUser	For Windows 2000 Terminal Services	When an optional Internet Connector license is enabled, Terminal Services clients are not prompted with a logon dialog box. Instead, they are logged on automatically with the TsInternetUser account.

- If your workstation is a member of a domain of the forest where the Guest account is enabled (in some of its domains), you just type “guest” in the logon dialog box, select the correct domain, and start using the workstation and the network.
- If your workstation is in a workgroup or in a different forest from the one in which the Guest account is enabled (in some of its domains), you first need to log on with some other user account. When you connect to the resources of the domain where the Guest account is enabled, you are granted access based on that Guest account’s permissions. You never type “guest” anywhere—you just use “Jack,” for example. When the server doesn’t recognize “Jack,” it switches to use “Guest” automatically. The catch is that if there is another Jack, who most likely uses a different password, you are denied access. The server just thinks that someone is trying to crack Jack’s account and doesn’t use Guest at all.

## Predefined Groups

Active Directory includes predefined security groups. Some of them reside in the Builtin container and the rest reside in the Users container, as follows:

- *Builtin*: Built-in local security groups
- *Users*: Mostly global security groups

The primary purpose of most of these predefined groups is to be the means by which administrative rights and permissions are assigned. To be anything more than an end user in the network, a user needs one or more of the following types of permission or rights:

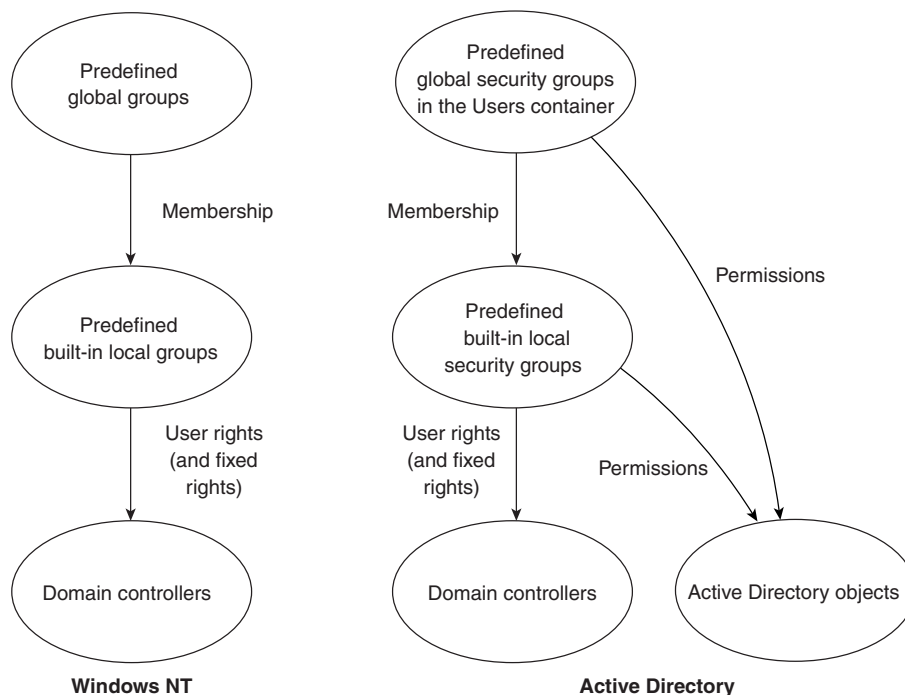
- User rights, such as permission to change the system time or log on locally. These rights are controlled with Group Policy settings and/or local policy settings. There are also some fixed rights. For example, only members of the Administrators group can format hard drives, and you cannot give this right to anyone else.
- Administrative permissions (i.e., the ability to create, delete, change, and so on) for Active Directory objects.
- Administrative permissions for registry keys.
- Administrative permissions for folders and files.
- Administrative permissions for other resources (printers, for example).

Most of the predefined groups have specific administrative rights or permissions associated with them, so you can give some users the appropriate rights and permissions by adding their names to the corresponding groups. Instead of worrying about all of the items in the list individually, it is far easier to just put Jack in the Account Operators group and Jill in the DNS Admins group, for example. They will get suitable permissions in one package.

Sticking just to “predefined” doesn’t get you through life, though—at least not with Active Directory. You often need to assign individual rights and permissions, probably not using the predefined groups. But that’s another story for another chapter (Chapter 4, to be exact).

Figure 3.2 shows the relationship among the groups in the Builtin and Users containers in an Active Directory domain. It also shows the corresponding relationships that existed in Windows NT.

## 148 Chapter 3 Managing OUs, Users, and Groups



**Figure 3.2** In Windows NT, the only meaning of predefined global groups (Domain Admins, Domain Users, and Domain Guests) was that they were members of some built-in local groups, which in turn had rights to administer the system. This is true also for Active Directory, but in addition both group categories have certain direct permissions to Active Directory objects.

---

**NOTE** In addition to the permissions and rights shown in Figure 3.2, built-in local (security) groups have permissions for system files and registry keys.

---

**NOTE** In Windows NT it was easy to make a user of another domain an administrator in a local domain. It only required making him a member of the Administrators group in the local domain. Because of the difference in how global groups get permissions, as illustrated in Figure 3.2, this is more difficult in Active Directory. If you make the foreign user a member of Administrators in Active Directory, he won't get the permissions of Domain Admins, so he will be only a partial administrator. You cannot make him a member of Domain Admins, because that group accepts members only from the same domain. Note that this note only applies to

the administrative privileges for the domain account database and domain controllers. It doesn't apply to the administrative privileges for any member servers or workstations.

### Predefined Built-in Local Security Groups

Table 3.3 describes the predefined groups in the Builtin container. You cannot delete, rename, or move any of them, because of their system-flags

**Table 3.3** The Predefined Built-in Local Security Groups

Name	Predefined Members	Abilities*
Administrators	Administrator, Domain Admins, Enterprise Admins	By default, members of this group have almost total control of the domain controllers of the domain, including formatting hard drives and all the rights that the following four "operators" have. For Active Directory, this group has by default "Full Control except Delete Subtree or Delete All Child Objects" permission for almost all objects in the domain.
Account Operators	None	By default, members of this group can create, delete, and manage user, inetOrgPerson, group, and computer objects in the Active Directory domain, except in the Domain Controllers OU. Account operators can log on locally to domain controllers of the domain and shut them down. In Windows 2000, Account Operators can modify their own account and the accounts of other account operators. However, they cannot modify administrator accounts. Starting with Windows 2000 SP4 and Windows Server 2003, Account Operators cannot modify account operators. For more information, see the "AdminSDHolder Object" section in Chapter 4. <i>(continued)</i>

\* See also Tables 4.44 through 4.47 in Chapter 4 for the lists of specific rights of these built-in groups.

**150 Chapter 3 Managing OUs, Users, and Groups****Table 3.3** The Predefined Built-in Local Security Groups (*cont.*)

Name	Predefined Members	Abilities*
Server Operators	None	In the domain controllers of the domain, members of this group can create, delete, and manage file shares and printers, and start, stop, and configure services. Also in the domain controllers, they can log on locally, back up and restore files, change computer time, and shut down the domain controller either locally or remotely. Note that the list here is descriptive, but not quite exhaustive.
Backup Operators	None	By default, members of this group can back up and restore files and folders in the domain controllers of the domain, even if the member user doesn't have permissions for those files and folders. They can also log on locally on the domain controllers of the domain and shut the domain controllers down.
Print Operators	None	Members of this group can create, delete, manage, and share printers in the domain controllers of the domain, and by default they can create, delete, and manage printer objects in the Active Directory domain. They can also log on locally on the domain controllers of the domain and shut the domain controllers down.
Users	Domain Users, Authenticated Users, Interactive	By default, this group has no user rights or permissions. You can just ignore this group.** If you want to give permissions to all forest users, you can use Authenticated Users. You can also create groups such as SanaoUsers or SanaoBostonUsers and use them instead of the predefined Users group.

\*\* Note that being able to ignore the Users group refers to the Users group in Active Directory, which is visible only on domain controllers. Each member server and workstation has a separate Users group, and each of them has some permissions for the corresponding local computer. Therefore, that latter Users group you probably need to use when managing permissions of the workstations and member servers in your organization.

**Table 3.3** The Predefined Built-in Local Security Groups (*cont.*)

Name	Predefined Members	Abilities*
Guests	Guest, Domain Guests, IUSR- _servername, IWAM_servername (in AD2000)	By default, this group has no rights or permissions. You can just ignore this group.
Pre-Windows 2000 Compatible Access	Everyone*** and Anonymous Logon, if you selected “Permissions compatible with pre-Windows 2000 server operating systems” when you installed the domain; otherwise, Authenticated Users (the latter is the default)	By default, this group has permission to see all the objects in a domain and all the properties of all users, inetOrgPersons, and groups. Everyone/Anonymous need these permissions if you have certain server services (e.g., Remote Access Service) running on Windows NT servers in your Active Directory domain.
Replicator	None	Windows NT servers and workstations use this group for the Directory Replicator service.
Remote Desktop Users	None	Members of this group are allowed to use the Remote Desktop connection.
Network Configuration Operators	None	Members of this group can modify the local TCP/IP settings and some other network settings of the domain controllers. For a complete list, see Microsoft Knowledge Base article 297938.
Performance Monitor Users	None	Members of this group can use the Performance console, or perform similar monitoring with another tool.
Performance Log Users	Network Service	Members of this group can use the Performance Logs and Alerts console, or perform similar monitoring with another tool. <i>(continued)</i>

\*\*\* This refers to the well-known security principal Everyone.

**152 Chapter 3 Managing OUs, Users, and Groups****Table 3.3** The Predefined Built-in Local Security Groups (*cont.*)

Name	Predefined Members	Abilities*
Incoming Forest Trust Builders	None	This group appears only in the forest root domain. The members can create incoming, one-way trusts to the forest.
Windows Authorization Access Group	Enterprise Domain Controllers	Members of this group can read the constructed <code>tokenGroupsGlobalAndUniversal</code> (TGGAU) attribute on user, <code>inetOrgPerson</code> , group, and computer objects. TGGAU contains a list of the object's global and universal group memberships, and an application can use this information, for example, to make decisions about users that are not logged on. The Pre-Windows 2000 Compatible Access group can also read TGGAU, but if the application is not in that group, you could use the Windows Authorization Access Group instead. For more information, see Microsoft Knowledge Base article 331951.
Terminal Server License Servers	None	This group is used for Terminal Server licensing.

protection. Note that each group in the table is always present in all domains. They have rights and/or permissions to their local domain only, and those rights/permissions apply only on the domain controllers. For example, the Remote Desktop Users group members can use the remote desktop of the domain controllers of the domain in question.

---

**NOTE** In the next chapter, we describe in more detail the default user rights and default Active Directory permissions of the groups in Table 3.3.

---

### ***Predefined Groups in the Users Container***

The remaining predefined groups are in the Users container. They are mostly global security groups, but there are also some domain local

security groups. Table 3.4 describes the predefined groups in the Users container of a domain.

**NOTE** When you install the first domain of the forest, Enterprise Admins and Schema Admins are global groups. When you later change this domain to the Windows 2000 native or Windows Server 2003 functional level (as discussed in Chapter 2), those groups will change to universal groups, which allows them to have members from other domains.

**Table 3.4** The Predefined Groups in the Users Container

Name	Predefined Members	Description
Enterprise Admins	Administrator of the first domain of the forest	Members of this group can administer all the domains in the enterprise. By default, this group is a member of Administrators in all domains of the forest. Enterprise Admins has Full Control to practically all objects in all domains of the forest. In addition, membership in this group is necessary to create child domains or sites. This group appears only in the first domain of the forest (that is, the forest root domain.)
Schema Admins	Administrator of the first domain of the forest	Members of this group can modify the schema of the forest. This group appears only in the first domain of the forest (that is, the forest root domain.)
Domain Admins	Administrator	Members of this group can administer this domain. By default, this group is a member of Administrators in this domain and all joined workstations/member servers. Domain Admins has Full Control to most objects of the domain.
Group Policy Creator Owners	Administrator	Members of this group can create Group Policy objects if they also have appropriate permissions for the OU for which they are creating the GPO. In addition, they can manage the GPOs they have created. <i>(continued)</i>

**154 Chapter 3 Managing OUs, Users, and Groups****Table 3.4** The Predefined Groups in the Users Container (*cont.*)

<b>Name</b>	<b>Predefined Members</b>	<b>Description</b>
Domain Users	Every user account of the same domain	By default, this group has no rights or permissions. You can use it if you need to give permissions to all users of the domain.
Domain Guests	Guest	By default, this group has no rights or permissions. You probably don't need this group.
Domain Controllers	Each domain controller of the same domain	By default, this group has no rights or permissions. You can use it if you need to give permissions to all domain controllers of the domain.
Domain Computers	Each workstation and member server of the same domain	By default, this group has no rights or permissions. You can use it if you need to give permissions to all workstations and member servers of the domain.
Cert Publishers	Each computer that is running an enterprise certificate authority	By default, this group has permission to read and write the <code>userCertificate</code> property of the users and computers in the domain. Therefore, members of this group can publish certificates for users and computers.
DnsUpdate Proxy	None	DHCP servers may dynamically register DNS resource records on behalf of DHCP clients. In this case, the DHCP servers become the owners of those records. This is a problem if the client or some other DHCP server later wants to start maintaining those records. By placing the computer objects of the DHCP servers as members in this group, the servers won't become record owners, so the problem described here is resolved. This group is missing if there is no DNS service in the domain.
IIS_WPG	None	The IIS_WPG group (worker process group) appears with Internet Information Services (IIS) 6.0, and it is used for the needs of IIS.

**Table 3.4** The Predefined Groups in the Users Container (*cont.*)

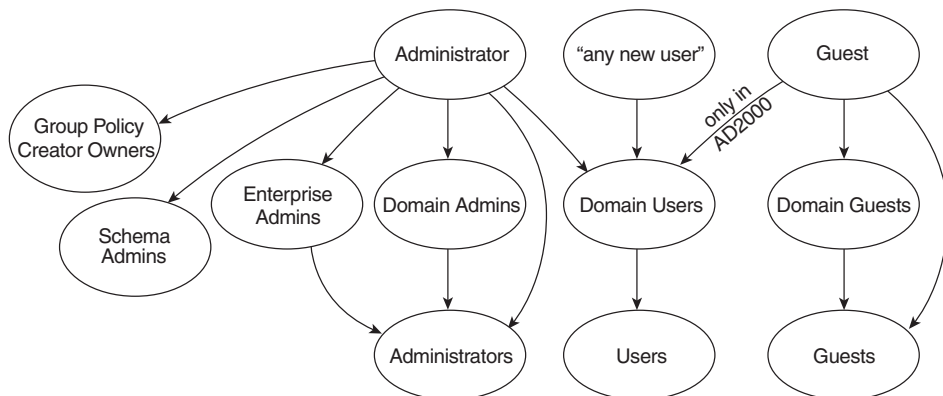
Name	Predefined Members	Description
DnsAdmins	None	Members of this group can administer the DNS service. This group is missing if there is no DNS service in the domain.
RAS and IAS Servers	Each computer that is running the Routing and Remote Access Services (RRAS)	By default, this group has permission to read Logon Information, Remote Access Information, Group Membership, and Account Restrictions of all users of the domain. RRAS servers need those permissions.

By default, Domain Admins is a member of the Administrators group of all workstations and member servers. Similarly, Domain Users is a member of the Users group of those computers.

Figure 3.3 illustrates the memberships of some predefined users, global groups, and built-in local groups that are listed in Tables 3.2 through 3.4.

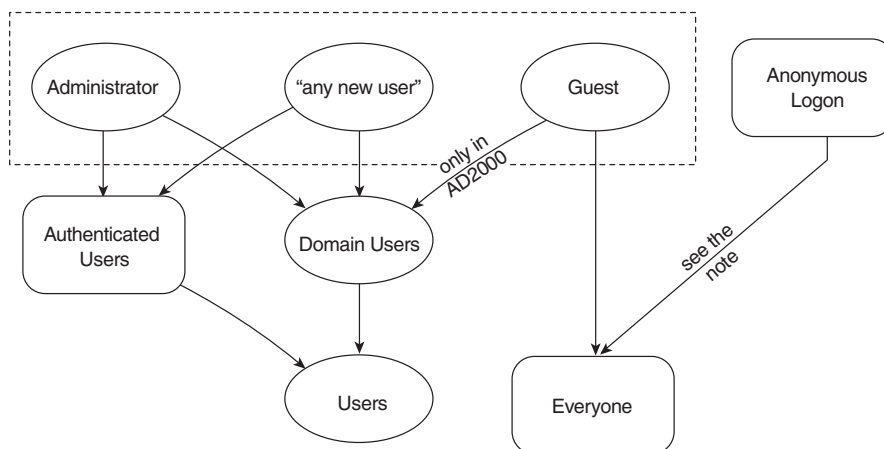
In Chapter 4, we discuss the well-known security principals. Many of them are like groups, and you can assign permissions to them. They are not real groups, however, because the operating system, not a network administrator, controls their “membership.”

Well-known security principals include Authenticated Users and Everyone, which you already saw as group members in Table 3.3 and



**Figure 3.3** The predefined users and groups have several predefined memberships. In addition, any new user is a member of Domain Users.

## 156 Chapter 3 Managing OUs, Users, and Groups



**Figure 3.4** The well-known security principals Authenticated Users, Everyone, and Anonymous Logon can be seen as part of the membership hierarchy. However, their “membership” is controlled by the operating system, not by a network administrator.

Table 3.4. Therefore, we have included Figure 3.4 to illustrate those memberships here, even though the remaining discussion is in the next chapter.

**NOTE** In AD2000, Anonymous Logon is a “member” of Everyone. In AD2003, the security was tightened, and the “membership” is true only if you enable the security policy “Network access: Let Everyone permissions apply to anonymous users.” This policy is located in Computer Configuration, Windows Settings, Security Settings, Local Policies, Security Options. See Chapter 7 for more information on policies.

Figure 3.4 reveals the memberships for the various end-user groups. Table 3.5 lists the end users (users of the group’s domain, users of the whole forest, and so on) that are members of each user group.

Typically, an administrator uses Authenticated Users to assign permissions to all users of a forest and Domain Users to assign permissions only to the users of one domain.

### Predefined Computer Objects

In the beginning, there is just one computer object. It is for your first (and at that point, only) domain controller in the Domain Controllers container.

**Table 3.5** End-User Memberships

Group	Type	All Users		Guest	Anonymous
		in Group's Domain	All Forest Users		
Everyone	Well-known	X	X	X	(see the Note)
Users	Built-in local	X	X	only in AD2000	
Authenticated Users	Well-known	X	X		
Domain Users	Global	X		only in AD2000	

## Administering OUs

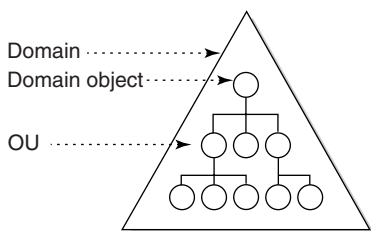
As you know, it is more efficient to organize your disk files in folders than to keep them in the root directory of a disk. Similarly, you are usually better off when you store Active Directory users, groups, and other objects in “folders” called OUs (organizational units). These OUs form an OU tree (also referred to as a domain structure) inside your domain. Figure 3.5 illustrates this.

**NOTE** In Figure 3.5, the uppermost circle (the root of the tree) is not an OU but rather the domain object that represents the domain (the triangle). We could drop the domain object out of the image, but it's more natural to have the tree as a whole. Also, in many ways the domain object behaves like an OU, so you can think of it as part of the tree.

### Features of OUs

Besides providing a logical structure through the OU tree, OUs offer the following benefits.

- An OU is a Group Policy target, so you can assign a different Group Policy to each OU.



**Figure 3.5** OUs inside a domain form an OU tree.

- If you want to delegate administration of some Active Directory objects, the most convenient way to do so is to put them in one OU and delegate administration of that OU. You could delegate administration of even single users and other objects, but the outcome would be difficult to manage. If you stick to only per-OU permissions, it is easier for you to track what you are doing.
- Using per-OU permissions, you can control object visibility—that is, which objects and object properties various users may see.

Unfortunately, even though you can assign permissions *for* OUs, you cannot assign permissions *to* OUs. In other words, you cannot define that all users in a certain OU get access to a certain folder or other resource. This will probably result in extra work for you, because you need to create a security group and put all the users in this group to give them access.

---

**IF YOU KNOW NDS** In NDS you can give permissions to OUs, so there is no need to create a group to correspond to each OU.

---

**NOTE** In Active Directory, OUs are not related to partitioning the directory database. They are purely logical units inside a domain. The domain in turn is the partition unit.

---

If there are several domains in your forest, each has a totally independent OU tree. The OU tree of an upper domain does not “continue” to a tree in a lower domain. However, if you have a Windows 2000 workstation and look at the tree by selecting My Network Places, Entire Network, Directory, you will see the child domains as siblings of the first-level OUs, as Figure 3.6 illustrates. Windows XP or Windows Server 2003 doesn’t show Directory in My Network Places.

---

**TIP** Although it is not supported by Microsoft, you can enable the OU browsing of Windows 2000 also in Windows XP. Just copy the file DSFolder.dll from Windows 2000 to the System32 folder of Windows XP and register it with the command `regsvr32 dsfolder.dll`.

---

---

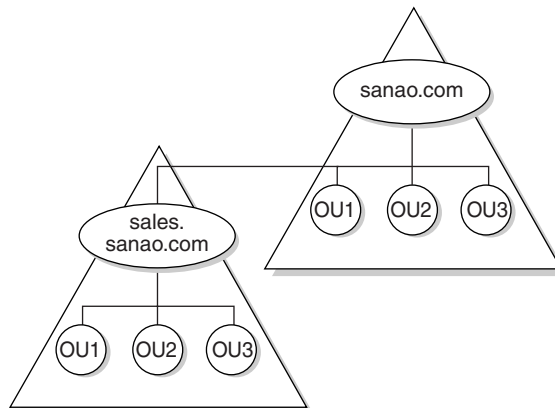
**IF YOU KNOW NDS** In NDS, all OUs form one big tree.

---

OUs are created primarily for administrators' use—end users don't usually see OUs. For example, when an end user performs a search operation for other people in Active Directory (by clicking the Start button and selecting Search), the user doesn't see the found users' OUs at all, and he couldn't even if he wanted to. For example, if there is a Jack Brown in OU Sales and another Jack Brown in OU Production, the person doing the search cannot tell the difference between them from the search dialog box. This is also true if a user is searching for a certain printer.

On the other hand, if the user has a Windows 2000 workstation and selects My Network Places, Entire Network, Directory, he will be able to browse the OU tree and see which user or printer is in which OU.

It is a matter of opinion whether hiding the OU tree from users is a good or bad thing.



---

**Figure 3.6** The Sales domain is a child of the Sanao domain. If you look at the tree via My Network Places of Windows 2000, you will see Sales as a sibling of the first-level OUs of Sanao.

## Managing OUs

Managing OUs includes the following tasks:

- Creating OUs
- Setting OU properties
- Moving, renaming, and deleting OUs
- Setting Group Policy, checking the Resultant Set of Policy (RSOP), assigning a COM+ partition set, assigning permissions, and delegating administrative tasks

In this chapter, we focus on the first three items in the list. The last item is discussed in later chapters as follows: Group Policy and RSOP, see Chapter 7; permissions and delegating, see Chapter 4; and COM+ partition sets, see Windows Server 2003 Help and Support Center.

As you read on, we encourage you to try these management tasks in your domain. You cannot do any irreversible harm to your domain.

### ***Creating OUs***

Creating an OU is as easy as creating a disk folder. Just follow these steps:

1. Launch the Users and Computers snap-in.
2. Right-click the parent OU you want (or the domain object) and choose New, Organizational Unit.
3. Type in the name you want and press Enter.

---

**IF YOU KNOW NDS** Unfortunately, the Insert key doesn't do the trick here as it does with the NwAdmin software for NDS.

---

The maximum number of characters in an OU's name is 64, which is usually more than enough. After all, it is best to use short (but descriptive) names. The OU name is a Unicode character string, so at least in theory you could have some Gurmukhi characters in an OU name. You could also put all the possible punctuation characters in an OU name, but this would make your life harder if every now and then you had to type the distinguished name of such an OU.

### Setting OU Properties

After you have created an OU, you can set its properties by right-clicking the OU and choosing Properties. The dialog box in Figure 3.7 will appear.

Table 3.6 lists the property choices. None of them affects the way Windows works. They just provide information for human beings.

Table 3.6 shows the property LDAP names, which you will need if you use certain Resource Kit utilities or scripting, or if you set per-property permissions. One of the properties in the table is indexed, and five are part of the global catalog. Indexing makes searches faster, and the global catalog makes reading properties faster if you have multiple domains and sites.



**Figure 3.7** Some of the properties that you can enter for an OU include address-related information.

**NOTE** When you set properties for an OU, if you add a user in the Managed By tab as the “manager” of an OU, that user doesn’t get any permissions for the OU. This setting is purely informational. The other fields on that tab are the manager’s properties, not the OU’s.

**162 Chapter 3 Managing OUs, Users, and Groups****Table 3.6** Properties of an OU Object

Property	LDAP Name*	Syntax	Indexed	In GC
Description	description	Text (1,024)**		X
Street	street (Street-Address)	Text (1,024) (Each new line takes two characters.)		X
City	l (Locality- Name)	Text (128)	X	X
State/province	st (State-Or- Province-Name)	Text (128)		X
Zip/Postal Code	postalCode	Text (40)		
Country/region***	co (Text-Country)	Text (128)		
	c (Country-Name)	Text (3)		X
	countryCode	Integer		
Managed By	managedBy	DN**** (You select a user or contact from a list.)		

\* In addition to the LDAP name, each property has a common name. It is included in parentheses if it is different from the LDAP name.

\*\* If the syntax is Text (i.e., a string of Unicode characters), we indicate also the maximum number of characters in the property (e.g., 1,024).

\*\*\* Country/region is stored in three properties: `co` contains the country's name (e.g., UNITED STATES), `c` contains the country's abbreviation (e.g., US), and `countryCode` contains the numeric ISO country code (e.g., 840).

\*\*\*\* DN = distinguished name.

---

**NOTE** Behind the scenes, the base schema lists 123 possible properties for an OU (104 in AD2000). Most of them are not used, so it doesn't matter that you can set only a few of them using the Users and Computers snap-in.

---

If you have Advanced Features turned on in the Users and Computers snap-in, you will see also the Security and Object tabs in the properties dialog box. The information in the former tab is discussed in Chapter 4, and the information in the latter tab is discussed in Chapter 5.

### ***Moving, Renaming, and Deleting OUs in a Tree***

You may find that your original OU tree is no longer optimal as a result of either insufficient planning or changed circumstances. If you need to rearrange your OU tree, you can easily move, rename, and delete OUs.

To move an OU inside a domain, either (a) drag it to a new location with the mouse, (b) use cut/paste with the keyboard or mouse, or (c) right-click the OU, select Move, and then choose the destination from the OU tree that opens up and click OK.

Note that not all of the OU's group policies and permissions move with it.

- Group policies and permissions that are assigned for the object being moved move with the object.
- Group policies and permissions that are inherited from above do not move with the object being moved. Instead, the OU will inherit new ones in its new location.

You can move several sibling OUs at once. Select them in the right-hand pane of the snap-in by using the Shift and/or Ctrl keys. Then proceed as previously described.

---

**NOTE** If you want to move an OU to another domain in your forest, you need to use another tool, such as the Support Tools command-line tool MoveTree. It is discussed further in Chapter 6.

---

You can rename an OU either by right-clicking the OU and selecting Rename or by selecting the OU and pressing F2. After you type the new name, press Enter.

Similarly, you delete an OU by right-clicking it and selecting Delete or by selecting the OU and pressing the Delete key. If the OU being deleted contains other objects, you are prompted to accept deleting them, too.

### **Planning OUs**

Even though "OU" stands for "organizational unit," you don't necessarily create OUs to match the organizational units of your company. You create OUs for administrative units, physical locations, and object types (e.g., an OU for users, an OU for printers, and so on), or you can create OUs based on corporate structure.

## 164 Chapter 3 Managing OUs, Users, and Groups

---

OU trees are like folder trees on disk: There isn't just one "right" way to create them. When planning your OUs, keep in mind the following aspects of OUs:

- OUs are purely logical entities: They are not related to physical partitions or replication.
- OUs are for delegation of administration.
- OUs are for Group Policy (including application publishing and assignment).
- OUs are for controlling object visibility.
- OUs are easy to reorganize. However, reorganizing them may confuse some users if they have learned a certain structure.
- Each OU should have a specific need and purpose to exist.
- There is no practical limit on how deep the OU tree can be. However, keep in mind the previous bullet about a specific purpose for each OU.
- OUs are mainly administrative units; typically users do not see them (although Windows 2000 users can if they want).

If you have more than one domain, you might want the OU trees in all domains to be planned according to similar principles.

The aforesaid suggests that you should create OUs based on how administration is organized in your organization. The three typical scenarios are the following:

- **Geographical:** If Boston has its own administrators and London has its own, you should probably create the Boston and London OUs.
- **Object type:** If some people administer users and others administer printers, you should probably create the Employees and Printers OUs, for example.
- **Organization:** If the Sales department has its own administrators and Production has its own, you should probably create the Sales and Production OUs.

It is also quite possible that more than one of these three divisions are used in your organization. In this case, you should create one level based on one division and another level based on another division. For example, your top-level OUs could be based on geography, and second-level OUs based on object type.

---

## Administering Users, InetOrgPersons, and Contacts

---

The traditional reason for creating user accounts is to give your users a means to log on to the network. The properties of a user's account control the user's access to the network, and the properties can define some network services for the user in question. Examples of these properties are the password, the account expiration date, a requirement for a smart card logon, and the network path of the user's home folder.

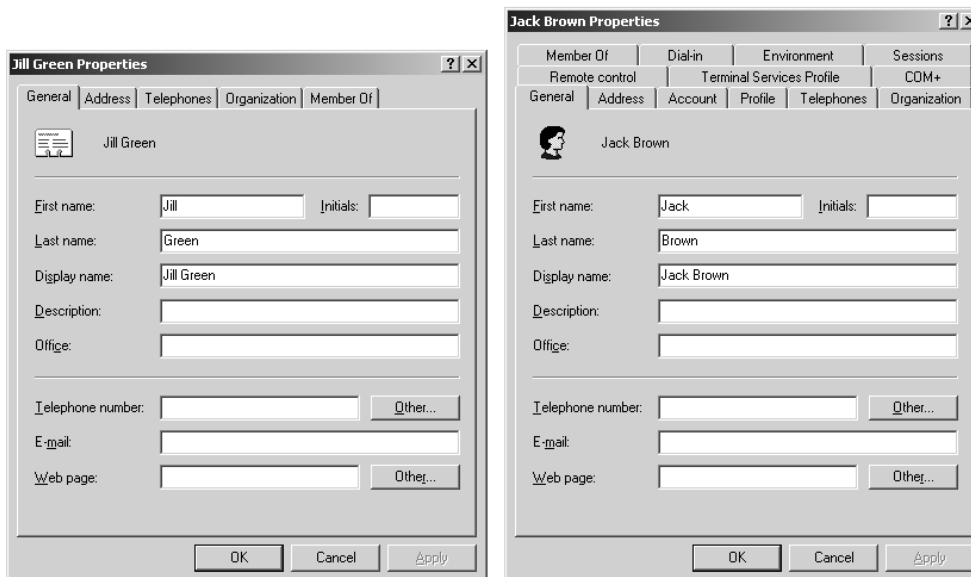
Directory services such as Active Directory have brought a second aspect to user accounts. At this point, we tend to refer to them as "user objects" instead of "accounts." In addition to being a means of access to the network and its services, a user object can store additional information about the user. Some of this information is meant for other human beings—for example, the user's fax number, title, or Web home page address. As a container of such "contact" properties, a user object can function much like an address book entry. A user object can also include properties for use by directory-enabled applications (e.g., Exchange e-mail, a faxing application, personnel-management software, and so on).

In addition to user objects, you can create *contact objects*. Typically you create a user object for each employee of your organization and a contact object for each person outside your organization whose contact information you want to store. A contact object can contain a subset of the properties that a user object can contain, as you can see in Figure 3.8 and Table 3.7.

The Users and Computers snap-in shows the properties of a contact and user object in a number of tabs in the properties dialog box, as shown in Figure 3.8.

Table 3.7 lists the tabs shown in Figure 3.8, except for the tabs Remote control, Terminal Services Profile, Environment, and Sessions, which are related to Terminal Services. (We don't cover them in this book about Active Directory.) Table 3.7 introduces the terms *significant properties* and *informational properties* and shows that a user object can contain both types of properties, but a contact object can contain only the latter.

## 166 Chapter 3 Managing OUs, Users, and Groups



**Figure 3.8** Contact object properties on the left are shown in five tabs. User object on the right has the same five tabs of a contact object and eight additional tabs. The five tabs that appear in both screen shots (General, Address, Telephones, Organization, and Member Of) contain the same properties except that the Member Of tab contains a Primary Group setting only for user objects.

The Users and Computers snap-in contains tabs for user and/or contact objects that are not shown in Figure 3.8.

- The Published Certificates tab is visible only when you turn on Advanced Features from the View menu.
- Turning on Advanced Features also makes the Object and Security tabs visible. Because they are common to all object types, we don't include them in this discussion of user and contact objects.
- Applications can add tabs. For example, if you install Exchange 2000, it will add some tabs, such as Exchange General and Exchange Features.

To summarize the functions for user objects (and to add a couple of functions):

**Table 3.7** The Nature of User and Contact Objects

Tab Name	User Object	Contact Object	Category*
Account	X		<i>Significant properties:</i> Properties that control user access to the network or define network services for the user
Profile	X		
Published Certificates	X		
COM+	X		
Member Of**	X		
Dial-in	X		
General	X	X	<i>Informational properties</i> Properties that contain information for human beings or are meant for some applications to use
Address	X	X	
Telephones	X	X	
Organization	X	X	
Member Of	X	X	

\* The terms “significant properties” and “informational properties” are not official. They are introduced in this book to distinguish these two types of properties.

\*\* The Member Of tab is shown twice because it has two natures: security and distribution list. The first nature applies only to user objects, but the second nature applies to both user and contact objects.

- A user object is an account that a user can log on with (using the corresponding significant properties).
- A user object is a placeholder for a collection of informational properties.
- A user object is a *security principal*. This means that you can give permissions to the user for resources and assign security group memberships to the user.
- The location of a user object in Active Directory dictates which group policies apply to the corresponding user.

A contact object (actually, the person who corresponds to the object) can never log on to the network. Also, a contact object is not a security principal, so it cannot have any permissions. Of course, even if a contact

## 168 Chapter 3 Managing OUs, Users, and Groups

object had permissions, no one would be able to use them, because a contact object cannot be used to log on.

The third type of people is the `inetOrgPerson` object, which is new to AD2003. An `inetOrgPerson` object is identical to a user object in practically every way. For more information, see the “Creating InetOrg Persons” section a little later in this chapter.

When you start to manage users and contacts, your tasks will include some or all of the following.

- Create users, `inetOrgPersons`, and contacts.
- Set user, `inetOrgPerson`, and contact properties.
- Copy users and `inetOrgPersons`, and move, rename, and delete users, `inetOrgPerson`, and contacts.
- Assign Group Policy and permissions, and delegate administration.

The next sections cover the first three items, but as mentioned earlier, the last item will be discussed in later chapters (Chapter 7 and Chapter 4).

If you want to try the management tasks discussed in this section, create a test OU where you can create test users.

### Creating Users

When you choose to create a user with the Users and Computers snap-in, you use a three-page wizard to do so. Figure 3.9 shows the first page

The screenshot shows a Windows XP-style dialog box titled "New Object - User". At the top left is a small icon of a person's head. To its right, it says "Create in: sanao.com/Boston". Below this are several input fields: "First name:" with "Jack", "Initials:" (empty), "Last name:" with "Brown", and "Full name:" with "Jack Brown". There are also two "User logon name:" fields. The first one has "jack.brown" and a dropdown menu showing "@sanao.com". The second one is labeled "User logon name (pre-Windows 2000):" and contains "SANA0\JackB". At the bottom of the dialog are three buttons: "< Back", "Next >", and "Cancel".

**Figure 3.9** On the first page of the user creation wizard, you enter the various names of the new user.

of the wizard, where you enter the various names of the new user. Figure 3.10 shows the second page of the wizard, where you can specify a password and some password settings. For example, you can require that a new user change her password at first logon so that only the user knows it and only she can legitimately log on with that account. Alternatively, you can specify that the user cannot change the password. This capability is useful, for example, when several users use the same account. With this setting, you can prevent any of the users from changing the common password. The third page of the wizard displays a summary of what you have selected.

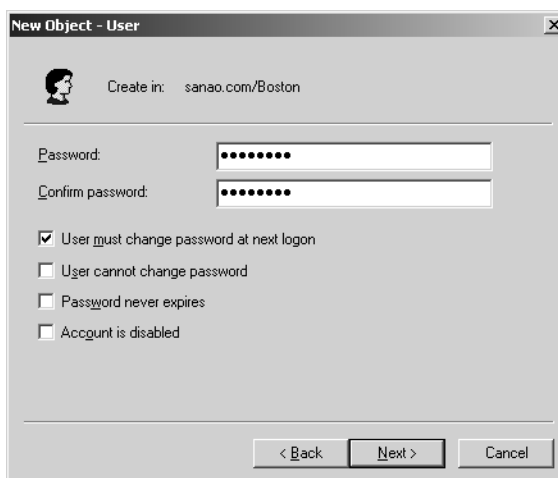
Table 3.8 describes the different name properties shown in the first page of the user creation wizard. All the name properties in the table are Unicode strings, and all, except Initials, are indexed and part of the global catalog.

---

**WARNING** Experience with Windows NT shows that using even common European characters, such as ä, in names may cause problems. Even though they are supported in principle, many command-line and graphical utilities can't handle them.

---

In addition to the name properties in Table 3.8, each object has a distinguished name and a canonical name (see Chapter 1). Furthermore,



**Figure 3.10** On the second page of the user creation wizard, you can specify a password and the way it will be used.

**170** Chapter 3 Managing OUs, Users, and Groups**Table 3.8** Name Properties of a User Object

Property	LDAP Name	Maximum Length (Characters)	Required	Unique	Description
First name	givenName	64		No	Purely informational.
Initials*	initials	6		No	Purely informational.
Last name	sn (Surname)	64		No	Purely informational.
Full name	name (RDN) and cn (Common-Name)	64	X	Within OU	This becomes the object's common name in the OU tree. The wizard suggests "firstname initials. last-name".**
Display name	display-Name	256		No	Purely informational, initially the same as Full name. You can change it later independent of Full name.

\* The user creation wizard treats Initials as the middle-name initial and not the first- and last-name initials (for example, "JB" for "Jack Brown").

\*\* You can modify the forest configuration so that the default full name is "lastname, firstname" instead of the normal "firstname lastname." We explain how to do this in Table 9.8 in Chapter 9.

**Table 3.8** Name Properties of a User Object (*cont.*)

Property	LDAP Name	Maximum Length (Characters)	Required	Unique	Description
User logon name	user-Principal-Name	1,024	X	Within forest	User can log on using this name on a Windows 2000 or later computer. This name is often the same as the user's e-mail address.
User logon name (pre-Windows 2000)	sAMAccount-Name	256 (schema rule), 20 (SAM rule)***	X	Within domain	User can log on using this name on any old or new Windows machine. Despite its label, this name can be used throughout Windows 2000 and later. This name also becomes the name of the user's profile folder when she logs on for the first time.

\*\*\* The maximum length of the sAMAccountName property is 256 characters in regard to schema rules. On properties that relate to Windows NT compatibility, however, Active Directory enforces SAM rules also. Consequently, the actual maximum length is 20 characters.

## 172 Chapter 3 Managing OUs, Users, and Groups

there are two name properties in the base schema that the snap-in doesn't display: the middle name and the generation qualifier (Jr., Sr., III, and so on).

In most cases, you create one user object for each network user. However, some situations call for a second user account.

- If a user is an administrator, he might have two user accounts: one with normal privileges for everyday use and another one with administrative privileges. It is safer if he uses the latter account only when performing administrative tasks.
- If a user needs to use several forests and there is no explicit trust between them, she needs a user account in each forest.
- If a user accesses the network with a mobile device through the Mobile Information Server, he may have a second account with fewer rights and permissions for this mobile access than his normal account has.
- If a user has a stand-alone server or workstation that is in a workgroup instead of a domain, he will need a *local* user account in that machine. Active Directory user accounts cannot be used when the computer hasn't joined a domain.

### UPN Suffixes

User logon names consist of two parts: the actual user name (e.g., jack.brown) and a *UPN suffix* (e.g., @sanao.com). For the first part you can enter any text, but for the second part you must choose the UPN suffix from a fixed list. By default, the list contains the name of the domain (e.g., sales.sanao.com) and the name of the root domain (e.g., sanao.com).

An enterprise administrator of a forest can add UPN suffixes to the list using the Domains and Trusts snap-in (click the Start button and then select Administrative Tools, Active Directory Domains and Trusts). Once the snap-in has started, the enterprise administrator right-clicks the uppermost line of the left pane (i.e., Active Directory Domains and Trusts) and selects Properties. The dialog box that appears enables the administrator to define additional UPN suffixes.

If the root domain is corp.sanao.com, for example, the administrator can add a UPN suffix sanao.com, so the users in the forest can have logon names such as jack.brown@sanao.com instead of jack.brown@corp.sanao.com.

## Creating InetOrgPersons

AD2003 includes a new object type (that is, object class), `inetOrgPerson`, which is identical to the user object type in practically every way. `InetOrgPerson` was defined in RFC 2798 to represent a standard network user, and many other directory services use it for this purpose. Therefore, `inetOrgPerson` was brought along to Active Directory so that it would be easier to interoperate with these other products or to migrate them to Active Directory.

Although `inetOrgPerson` should be identical to `user`, Microsoft recommends that you test it with your applications that would use Active Directory as an authentication method, and your other projected usage scenarios, before you actually start using `inetOrgPerson` objects.

If `inetOrgPerson` objects are not needed in your forest, you can modify the forest schema so that `InetOrgPerson` doesn't appear in the New context menu of the Users and Computers snap-in. You would need to change the `defaultHidingValue` property of the `inetOrgPerson` schema class definition to `TRUE`. This setting affects all administrators of the forest, unless they use some other tool to create objects. For more information, see Chapter 9 or Microsoft Knowledge Base article 311555 at <http://www.microsoft.com>.

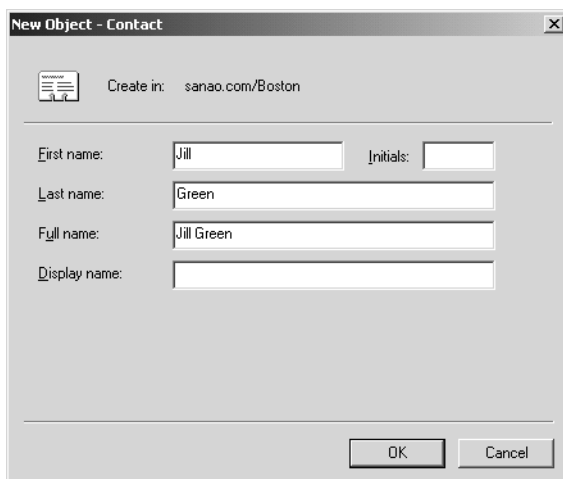
## Creating Contacts

To create a contact, you use the contact creation wizard in the Users and Computers snap-in. The wizard has only one page, which is shown in Figure 3.11. A contact object is like an address book entry for e-mail and other applications, and it contains only informational properties. It usually represents a person who is not working for your company, and a contact cannot log on to your network. Therefore, you don't specify a logon name for a contact object. The "Full name" entry becomes the common name of the object in the OU tree.

## Setting User, InetOrgPerson, and Contact Properties

You can define more than 50 settings for each user and more than 30 settings for each contact. Behind the scenes, a user object can have 257 properties (207 in AD2000) and a contact object can have 165 properties (138 in AD2000). Fortunately, the only *required* properties are a few names (which we mentioned in our discussion of creating users).

## 174 Chapter 3 Managing OUs, Users, and Groups



**Figure 3.11** When you create a contact, you don't specify logon names. Also, there is no second page, which would have the password settings (i.e., significant properties) that you saw when creating a user object.

---

**NOTE** Although we mention exact counts here and in many other places, you don't have to know the exact numbers. We use exact counts because it is simply easier to express "165 properties" than "well over 150 properties." It is not always possible to be precise, however. We say that you can define "more than 50" settings. In this case, there is more than one way to count the settings in the user interface.

---

Of the many possible settings, the major significant properties of a user object are set in the Account, Profile, and Dial-in tabs. The major informational properties of user and contact objects are set in the General, Address, Telephones, and Organization tabs. The Member Of tab is covered in the "Administering Groups" section of this chapter.

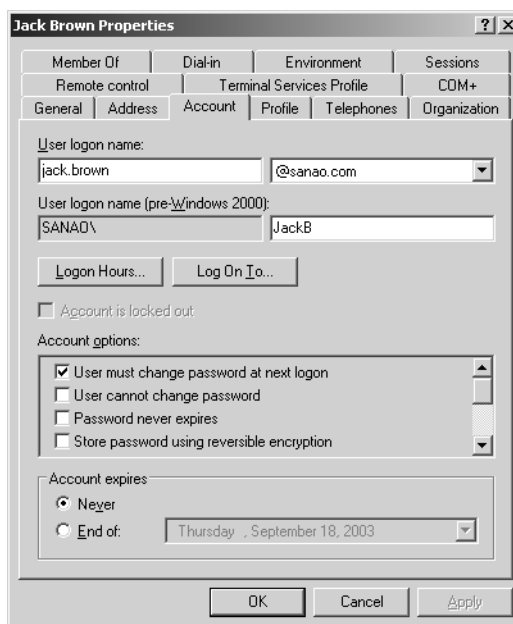
---

**NOTE** Windows provides context-sensitive help for each of the settings. In addition, many of the setting names are self-explanatory.

---

### ***Significant Properties of a User Object: The Account Tab***

Figure 3.12 shows the contents of the Account tab, which sets significant properties of a user. It includes settings that control how and when the



**Figure 3.12** The Account tab of the user Jack Brown

user can log on, as well as a few settings that control passwords. Table 3.9 lists other settings, except the 11 yes/no check boxes, which appear in Table 3.10.

**NOTE** Because Logon Hours is internally stored as GMT/UTC, an administrator who looks at a user's settings will see the hours as local to the administrator's time zone, regardless of where that is. For example, if a Boston administrator allows a user in Boston to log on between 8:00 AM and 3:00 PM, an administrator in Belgium (6 hours ahead of Boston) who checks that user's setting for logon hours would see times between 2:00 PM and 9:00 PM. There are no adjustments for daylight saving time, however. This is good because this way the allowed logon hours won't change twice a year, when daylight saving time and standard time start.

Table 3.10 lists the yes/no settings in the Account tab. You cannot set the first setting—you can only clear it. The other ten settings you can either set or clear. Eight of the 11 settings are stored in a property called `userAccountControl` so that one bit represents each setting.

**176 Chapter 3 Managing OUs, Users, and Groups****Table 3.9** Significant Properties of a User Object: The Account Tab

Property/ Setting	LDAP Name	Syntax	Description
User logon name	userPrincipal- Name	Text (1,024)	User can log on using this name on a Windows 2000 or later computer. This name is often the same as the user's e-mail address.
User logon name (pre-Windows 2000)	sAMAccount- Name	Text (256 [schema rule], 20 [SAM rule])*	User can log on using this name on any old or new Windows machine. Despite its label, this name can be used throughout Windows 2000 and later. Also, this name becomes the name of the user's profile folder when she logs on to each Windows NT/2000/XP/Server 2003 computer for the first time.
Logon Hours**	logonHours	(Binary)	Weekdays and hours in one-hour increments during which the user is allowed to log on.
Log On To/Logon Workstations	user- Workstations	Text (1,024)	A list of computer NetBIOS names that the user is allowed to log on to.
Account options	userAccount- Control	Yes/No	These 11 settings are described in Table 3.10.
Account expires	account- Expires	Date	The date after which the user account is no longer usable (although it doesn't vanish then). You can use this for temporary users.

\* If the syntax is Text (i.e., a string of Unicode characters), we indicate also the maximum number of characters in the property (e.g., 256).

\*\* The Logon Hours property is set and shown in local time but internally stored as GMT/UTC. The amount of time zone correction is taken from the local computer configuration.

The setting “Account is locked out” is stored in the `lockoutTime` property, the setting “User must change password at next logon” is stored in the `pwdLastSet` property, and the setting “User cannot change password” is determined by permissions. You can learn more about the way settings are stored in Chapter 11.

**Table 3.10** Significant Properties of a User Object: The Account Options

Setting	Description
Account is locked out	If someone tries to log on and enters a wrong password too many times, the account is locked either for a specified time or until the administrator unlocks it. You define the acceptable number of wrong attempts and associated time periods using Group Policy.
User must change password at next logon	After you assign a password to a user, it is a good practice to require the user to change it as soon as he logs on. Then you won't know it anymore.
User cannot change password	This is useful, for example, if several users use one account. You can use this setting to prevent them from changing the password.
Password never expires	You can force users to change their passwords periodically (e.g., every 30 days), but then use this setting to exempt some users from this policy. This is useful, for example, when defining passwords for service accounts. In that case, there is no human being to change the password every month.
Store password using reversible encryption	Normally Active Directory stores passwords using irreversible encryption, meaning that the user's clear-text password cannot be calculated (except through a special “dictionary attack”). You must enable this setting if the corresponding user is using a Macintosh workstation or if she wants to use IIS digest authentication to be able to pass a firewall.
Account is disabled	If a user is away a long time, you can “freeze” the user's account for that time but still not delete it.
Smart card is required for interactive logon	Self-explanatory.

(continued)

**178 Chapter 3 Managing OUs, Users, and Groups****Table 3.10** Significant Properties of a User Object: The Account Options (*cont.*)

Setting	Description
Account is trusted for delegation	This setting is described in Chapter 4 in the “Impersonation and Delegation” section. Note that when the domain is on the Windows Server 2003 functional level, this setting appears on the Delegation tab, and that tab is only visible for accounts that have been assigned service principal names.)
Account is sensitive and cannot be delegated	This setting is described in Chapter 4 in the “Impersonation and Delegation” section.
Use DES encryption types for this account	This setting causes Windows 2000 and later to use Kerberos DES-CBC-MD5 instead of the default RSADSI RC4-HMAC for this user account. The setting affects how Kerberos ticket-granting tickets (TGTs) are encrypted. Data Encryption Standard (DES) is used to encrypt both the ticket and the key of the initial TGT, and DES is also used to encrypt the key of the forwarded TGT. However, RSA is used to encrypt the ticket of the forwarded TGT.
Do not require Kerberos preauthentication	Normally Windows 2000 and later use preauthentication with Kerberos authentication, but it is not compatible with all implementations of Kerberos. Consequently, you must not require preauthentication if the corresponding user account is going to use such an implementation. Selecting this option may expose the user account to denial-of-service attacks.

There is one password setting that is not visible in the Users and Computers snap-in. You could type the following command on the command line when sitting at a domain controller:

```
NET USER JackB /PasswordReq:No
```

This command relieves JackB from having a password. For example, even though other users of the domain would be required to use at least a six-character password, he would not. Note that you must use the pre-Windows 2000 name of the user in this command.

Even though this command relieves Jack from having a password, he cannot clear his password—an administrator must do this. If Jack later changes his password to “abcdef,” he cannot change it back to empty.

You can see the current setting for Jack using the following command:

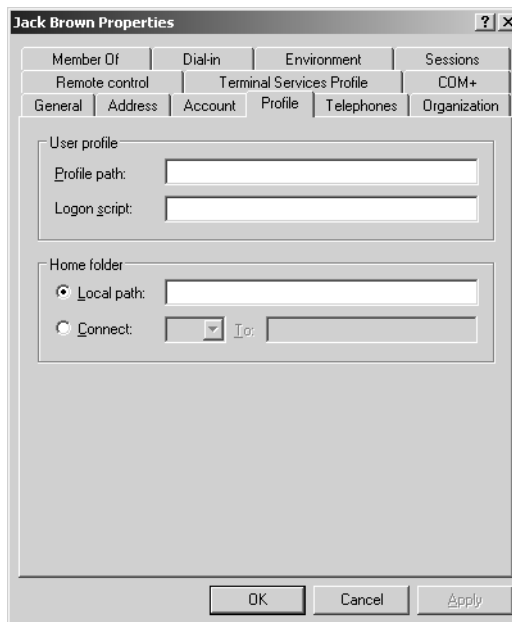
```
NET USER JackB
```

**NOTE** The minimum length of a password for domain users is set using Group Policy, which is discussed in Chapter 7.

Microsoft has prepared a long and thorough online document called Account Passwords and Policies, which you can access at the address <http://www.microsoft.com/technet/prodtechnol/windows/server2003/technologies/security/bpactlck.mspx>.

### **Significant Properties of a User Object: The Profile Tab**

Figure 3.13 shows the contents of the Profile tab. The Profile tab is not about control as the Account tab is—it's about providing services to users. Table 3.11 lists the Profile tab's four significant properties. They all may contain an “unlimited” number of Unicode characters.



**Figure 3.13** The Profile tab of the user Jack Brown

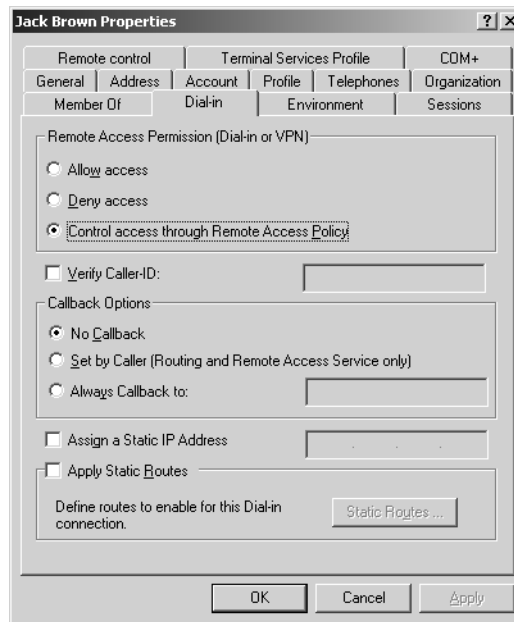
**180 Chapter 3 Managing OUs, Users, and Groups****Table 3.11** Significant Properties of a User Object: The Profile Tab

Property	LDAP Name	Description
Profile path	profilePath	This specifies a Uniform Naming Convention (UNC) name, such as \\Server\Prof\$\JackB, to be the network folder where the user's roaming profile is stored. This way, Jack's roaming profile is downloaded to whichever Windows NT/2000/XP workstation he logs on to, and it is uploaded back to the server when he logs off. The dollar sign (\$) in the Prof\$ sharename makes it invisible so that users don't browse it.
Logon script	scriptPath	This field is the old (i.e., Windows NT) way to define a logon script for a user. The new way (i.e., Active Directory) is to use Group Policy. An example of this path is Logon.Bat. The name is relative to the UNC path \\anydomaincontroller\Netlogon.
Home folder: Local path/To	homeDirectory	You can assign each user a private or shared folder on some server. The To field defines the path—for example, \\Server\Users\JackB. If possible, the snap-in creates the folder for you. It also gives Administrators and the user Full Control. The snap-in doesn't, however, remove the inherited permissions, so it is quite possible that the Users group will have Read permission for the new home folder. A home folder is an alternative to the My Documents folder, which you can also store on a server using Group Policy. When saving documents, newer applications usually default to <i>My Documents</i> , whereas some may use the %homedrive% and %homepath% environment variables. The "Local path" field defines a path such as D:\JackB, but that path exists on only one local machine.
Home folder: Connect	homeDrive	A drive letter that connects (or maps) to the user's home folder.

You may use the %username% environment variable in the “Profile path” and “Home folder: To” fields. Its value will be the user’s logon name (pre–Windows 2000)—that is, his “downlevel logon name.” For example, the path \\Server\Prof\$\%username% actually means \\Server\Prof\$\JackB. This variable is handy when you edit several users at once. At that time, for example, you will be able to set the home folder for several users at once.

### ***Significant Properties of a User Object: The Dial-in Tab***

The settings in the Dial-in tab define whether the user may use dial-in or virtual private network (VPN) connections, and if so, in what way. These significant properties apply more to managing communication settings than to managing user settings. Therefore, this tab is outside the scope of this book. The screen shot in Figure 3.14 is provided here for reference.



**Figure 3.14** The Dial-in tab defines whether the user may use dial-in or VPN connections.

### ***Informational Properties of Users and Contacts***

As previously stated, the informational properties don't affect the network user (unless you create a query-based group with the Authorization Manager snap-in and base that group on some otherwise informational property). They provide information for other people and for applications that use them. Consequently, these two criteria dictate how you use each of the informational properties. We cannot tell you here the rules to use each informational property, but we can offer a few general guidelines.

If you or any of your users are not interested in these properties, and if you don't have applications to take advantage of them, you can simply leave all the informational properties blank.

Except for Country/region and Manager, both of which you select from a list, you edit all the informational properties in text fields that have very little format checking. These fields have no stringent requirements for acceptable entries. This means that you could fill in the property fields with just about anything, such as your favorite recipes or the hair color of each user, even though the property label indicates a phone number.

Although you have free rein in determining informational properties, the following are some guidelines to keep in mind.

- Use each property consistently. Ideally, you have a written document that describes which properties are in use in your company and in what format the information should be entered.
- Some of the properties can be used in search operations. Here, consistency is especially important.
- Some of the properties can be used in query-based groups. Here, consistency is even more important.
- By default, each user can see all of his or her properties. Each user can also change those properties that are categorized as Personal Information and Web Information (together consisting of 43 properties, and the same number in AD2000).
- By default, every logged-on user can see certain properties of all other users. These properties are categorized as General Information, Public Information, Personal Information, and Web Information, and they consist of a total of 93 properties (89 in AD2000).

---

**NOTE** The information categories mentioned here (Personal Information, General Information, and so on) are used in the management of

permissions. Therefore, they are covered in detail in the next chapter, which deals with securing Active Directory. Unfortunately, the categories are quite different from the tabs in user properties. For example, General Information doesn't have anything to do with the General tab.

Table 3.12 lists the properties in the four tabs containing informational properties. We don't include screen shots, because they would show just a number of text boxes.

**Table 3.12** Informational Properties of User and Contact Objects

Property	LDAP Name	Syntax (Characters)	Index	GC	Comments
<b>General Tab</b>					
First name	givenName	Text (64)	X	X	
Initials	initials	Text (6)			Even though the creation wizard treats this as a middle-name initial, you can enter "JB" for an existing Jack Brown.
Last name	sn (Surname)	Text (64)	X	X	
Display name	displayName	Text (256)	X	X	This is not the common name (cn) you see in the OU tree. The user's display name is shown in the Computer Locked dialog box, for example.
Description	description	Text (1,024)		X	
Office	physical-Delivery-OfficeName	Text (128)	X		
Telephone number	telephone-Number	Text (64)		X	This is the primary office phone number. <i>(continued)</i>

**184 Chapter 3 Managing OUs, Users, and Groups****Table 3.12** Informational Properties of User and Contact Objects (*cont.*)

Property	LDAP Name	Syntax (Characters)	Index	GC	Comments
Phone Number (Others)	other-Telephone	Text (64)			These are the other office phone numbers.
E-mail	mail	Text (256)	X	X	
Web page	wwwHomePage	Text (2048)			http:// something, ftp:// something, file:// something.
Web Page Address (Others)	url	Text			A list of multiple values.
<b>Address Tab</b>					
Street	street-Address	Text (1,024)			
P.O. Box	post-OfficeBox	Text (40)			
City	l (Locality-Name)	Text (128)	X	X	
State/province	st (State-Or-Province-Name)	Text (128)		X	
Zip/Postal Code	postalCode	Text (40)			
Country/region	co (Text-Country)	Text (128)			For example, "UNITED STATES."
	c (Country-Name)	Text (3)		X	For example, "US."
	countryCode	Integer			For example, "840."
<b>Telephones Tab</b>					
Home	homePhone	Text (64)		X	

**Table 3.12** Informational Properties of User and Contact Objects (*cont.*)

Property	LDAP Name	Syntax (Characters)	Index	GC	Comments
Home Phone Number (Others)	otherHome-Phone	Text (64)			A list of multiple values.
Pager	pager	Text (64)			
Pager Number (Others)	otherPager	Text (64)			A list of multiple values.
Mobile	mobile	Text (64)			
Mobile Number (Others)	otherMobile	Text (64)			A list of multiple values.
Fax	facsimile-Telephone-Number	Text (64)			
Fax Number (Others)	other Facsimile-Telephone-Number	Text (64)			A list of multiple values.
IP phone	ipPhone	Text (64)		X	
IP Phone Number (Others)	other-IpPhone	Text		X	A list of multiple values.
Notes	info	Text (1,024)			
<b>Organization Tab</b>					
Title	title	Text (64)			
Department	department	Text (64)			
Company	company	Text (64)			
Manager	manager	DN; you select a user or contact from a list		X	Setting this doesn't give the manager any permissions. (continued)

**186 Chapter 3 Managing OUs, Users, and Groups****Table 3.12** Informational Properties of User and Contact Objects (*cont.*)

Property	LDAP Name	Syntax (Characters)	Index	GC	Comments
Direct reports	direct-Reports	DN			This property is read-only in the snap-in. If you set Jill to be Jack's manager, Jack will appear in the direct reports of Jill.

The “Country/region” field has a fixed set of options from which you choose. The result is stored in three properties, as described in the table.

**Editing Multiple Users**

The Windows Server 2003 version of the Users and Computers snap-in enables you to edit multiple objects at the same time. Typically, you would use this feature for user objects (or possibly for inetOrgPerson objects). For other object types, you can edit only the description text.

As you see in Figure 3.15, you can edit quite a few properties for multiple users simultaneously.

**Other Operations to Manage Users, InetOrgPersons, and Contacts**

After you have created a number of users and contacts (and possibly inetOrgPersons) and packed them full of properties, you are ready to perform other operations. Open the context menu by right-clicking with your mouse or press a shortcut key to manipulate existing users and contacts in the following ways:

- Copy (only users and inetOrgPersons, not contacts)
- Move
- Rename
- Delete
- Disable an account (only users and inetOrgPersons, not contacts)



**Figure 3.15** You can edit quite a few properties for multiple users simultaneously.

- Reset a password (only users and inetOrgPersons, not contacts)
- Open a home page
- Send e-mail

### ***Copying Users and inetOrgPersons***

You can copy an existing user or inetOrgPerson to create a new one. You do this by right-clicking the object and then selecting Copy. This launches a wizard similar to the one that enables you to create users from scratch. For brevity, we only talk here about users, because inetOrgPersons behave identically.

Copying a user saves time if the new user will have many of the same properties as an existing one. When you copy the user, by default 33 properties of the existing user are copied to the new one. However, only 20 of these properties are visible in the Users and Computers snap-in. Table 3.13 lists these properties, as well as some other categories.

The remaining 13 properties may have values to copy if you have set them programmatically with ADSI Edit or with some other means. However, it's not likely that you have done so.

**188 Chapter 3 Managing OUs, Users, and Groups**

Obviously, several properties (e.g., names and phone numbers) are personal and therefore not meaningful to copy. On the other hand, there are properties that would be nice to copy, but which are by default not included in the 33 copied properties. Table 3.13 lists five such properties.

If you anticipate needing to create several similar user objects, you can create user templates. A user template is a normal user object that represents a typical user of some department. When you need a new

**Table 3.13** Properties That Are Copied When Users Are Copied

Category	Properties
Copied and visible in the snap-in (20 properties)	accountExpires, c (Country/region)*, co (Country/region), company, countryCode (Country/region), department, homeDirectory, homeDrive, l (City), logonHours, manager, memberOf, postalCode (Zip/Postal Code), postOfficeBox, primaryGroupID, profilePath, scriptPath (Logon script), st (State/province), userAccountControl (Account options), and userWorkstations (Logon Workstations)
Copied but not visible in the snap-in (13 properties)	Assistant, codePage, division, employeeType, localeID, logonWorkstation, maxStorage, otherLoginWorkstations, postalAddress, preferredOU, showInAddressBook, showInAdvancedViewOnly, and street
Not copied but visible in the snap-in and would be nice to be copied (5 properties)	description, facsimileTelephoneNumber (Fax), otherFacsimileTelephoneNumber (Fax Number (Others)), physicalDeliveryOfficeName (Office), and streetAddress (Street)

\* We have included in parentheses the property names that you see in the Users and Computers snap-in if those names are quite different from the LDAP names in the table.

user for that department, you can copy the user template to be the new user and modify it as necessary.

The copied properties are defined in the schema. You can add attributes (e.g., `streetAddress`) to the list, as Chapter 9 will explain.

### ***Moving Users, InetOrgPersons, and Contacts***

Every now and then you may want to move some users, `inetOrgPersons`, or contacts from one OU to another. You move them within a domain either (a) by dragging the object to a new location with the mouse, (b) by using cut/paste with the keyboard or mouse, or (c) by right-clicking the object, selecting Move, and then choosing the destination from the OU tree that opens up and clicking OK. Note that

- Permissions that are assigned for the object being moved move with the object.
- Group policies (regarding users and `inetOrgPersons`) and permissions that are inherited by the object from above do not move with the object being moved. Instead, the moved object inherits the new group policies and permissions in its new location.

You can move several sibling objects at once. Select them in the right-hand pane of the snap-in by using the Shift and/or Ctrl keys. Then proceed as usual.

It is possible to move objects to another domain in your forest. To do so, you need to use another tool, such as the Support Tools command-line tool `MoveTree`, which is discussed in Chapter 6.

### ***Renaming Users, InetOrgPersons, and Contacts***

You can rename a user, `inetOrgPerson`, or contact by right-clicking the object and selecting Rename or by selecting the object and pressing F2. A third way is to click an already selected object. After you type the new name, press Enter. Because these objects have many names, you have a chance to change one or all of the names in a dialog box, as Figures 3.16 and 3.17 show.

After you rename a user, the old name still appears in the following properties: E-mail, Web page, Profile path, Logon script (if using personal), and Home folder. Also, the corresponding physical folders, as well as the local copy of the user's profile (i.e., `C:\Documents and`

## 190 Chapter 3 Managing OUs, Users, and Groups

**Figure 3.16** When you rename a user or an inetOrgPerson, you are prompted with a dialog box that enables you to change a number of names at once. The first field, Full name, refers to the common name of the object.

Settings\*username*), will keep the old name. If you want all of these to reflect the new name, you must change each of them manually.

### **Deleting Users, InetOrgPersons, and Contacts**

You delete an object by right-clicking it and selecting Delete or by selecting the object and pressing the Delete key. As a safety mechanism, you need to confirm the delete but you cannot undo it.

**Figure 3.17** When you rename a contact, you are prompted with a dialog box that enables you to change a number of names at once. The first field, Full name, refers to the common name of the object.

A user object or an inetOrgPerson object is a security principal: It may have security group memberships and permissions for resources. Each security principal has a security ID (SID), which is the identifier to be used in these assignments. A SID is a long number and a SID is never reused. If you delete a user object and then re-create it, it will have a new SID, so the new user has none of the memberships or permissions of the old user. You must assign memberships and permissions specifically to the new user.

### ***Disabling User or InetOrgPerson Accounts***

The context menu for a user object or an inetOrgPerson object contains an operation called “Disable Account.” It has the same effect as the “Account is disabled” check box in the Account tab of the properties dialog box. This operation is usually used for a limited time. For example, if someone is out of the company for six months, you could freeze his user account but still not delete it.

When you see a red X icon on the account, it is already disabled. In this case the context menu has an operation called “Enable Account.”

### ***Resetting User or InetOrgPerson Passwords***

You will never see your users’ or inetOrgPersons’ passwords, but you can change them using the “Reset Password” operation in the context menu. The most obvious reason to do this is because a user has forgotten his password.

### ***Opening Home Pages of Users, InetOrgPersons, and Contacts***

If someone has a home page, and the corresponding property is defined in his object, you can open the home page in a browser using the “Open home page” operation in the context menu.

### ***Sending E-mail to Users, InetOrgPersons, and Contacts***

If someone has an e-mail address, and the corresponding property is defined in her object, you can send her e-mail with the “Send mail” operation in the context menu.

---

## Administering Computer Objects

---

Just as Active Directory has a user object for each network user, it has a computer object for each computer in the domain. However, this applies “only” to Windows Server 2003, Windows XP, Windows 2000, and Windows NT computers. Other workstations (e.g., Windows 95 and 98 and non-Microsoft operating systems) that are not using the NT-based integrated security cannot have a computer object.

---

**IF YOU KNOW NDS** NDS allows a broader range of workstation types than does Active Directory, which means that you can manage more types of workstations with the help of the directory service.

---

Also, computer objects are used only for computers that join a domain. If a stand-alone server or workstation will be in a workgroup instead of a domain, it will not be assigned a computer object in Active Directory.

You could categorize computer object properties as either significant or informational, just as we did with user objects. However, the distinction among computer objects is not as clear as it is among user objects, so we don't use these terms with computer objects in this book (short of a couple of exceptions).

The purposes of computer objects are as follows:

- As inherited from the very first version of Windows NT back in 1993, a computer account ties the workstation or server to the Windows NT/2000/XP/Server 2003 security model.
- A computer object is a placeholder for properties that help you when you are remotely installing and managing workstations.
- A computer object is a placeholder for properties that are purely informational.
- A computer object is a security principal. This means that just as with a user, you can give permissions for resources and assign security group memberships to the computer.
- The location of a computer object in Active Directory dictates which group policies apply to the corresponding computer.

Computer objects are treated slightly differently, depending on whether they are for domain controllers or for workstations and member servers. Table 3.14 compares the two.

**Table 3.14** Comparing Domain Controllers and Other Computer Objects

Feature	Domain Controller	Workstation and Member Server
Creation of the object	Automatically while installing Active Directory on the server (using DCPromo).	<ul style="list-style-type: none"> <li>• Semiautomatically while joining the computer to the domain.</li> <li>• Manually with (a) the Users and Computers snap-in, (b) the DSAdd Computer command, (c) the NetDom tool (part of the Support Tools), or (d) using a script.</li> </ul>
Default container of the object	Domain Controllers.	Computers.
Use of the default location	Probably yes.	Probably not (place the computer objects in OUs instead).
Computer GUID	You cannot set this property.	You may set this property, which helps when using Remote Installation Services and signifies a managed computer.

When you start to manage computer objects, your tasks will include the following:

- Create computer objects.
- Set computer object properties.
- Move, rename, disable, reset, and delete computer objects.
- Assign Group Policy and permissions, and delegate administrative tasks.

In this chapter, we focus on the first three items in the list. The last item is discussed in later chapters. If you want to try the management tasks discussed in this section, you can create some test computer objects in your test OU. To test all the features, however, you will need some test workstations.

## Creating Computer Objects

As Table 3.14 in the previous section implies, computer objects are created in three ways.

- A computer object for a domain controller is created automatically in the Domain Controllers OU when you install Active Directory on that server by running the Active Directory Installation Wizard (i.e., DCPromo).
- When you join a stand-alone server or workstation to a domain, either during computer installation or afterward, you have the option to create the computer object. An object created in this way goes to the Computers container.
- You precreate the computer object manually using one of the four ways listed in Table 3.14. The Users and Computers snap-in way—the graphical choice—is explained next. The DSAdd Computer command is introduced at the end of this chapter.

---

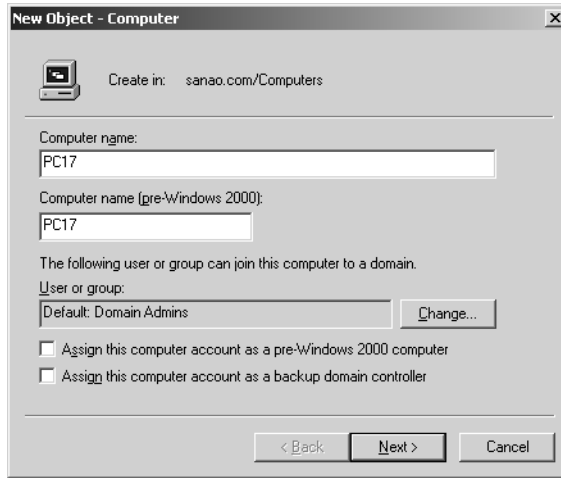
**NOTE** The second and third items in the list require appropriate permissions or user rights, which are explained in Chapter 4. In short, any forest user can by default join ten workstations to a domain.

---

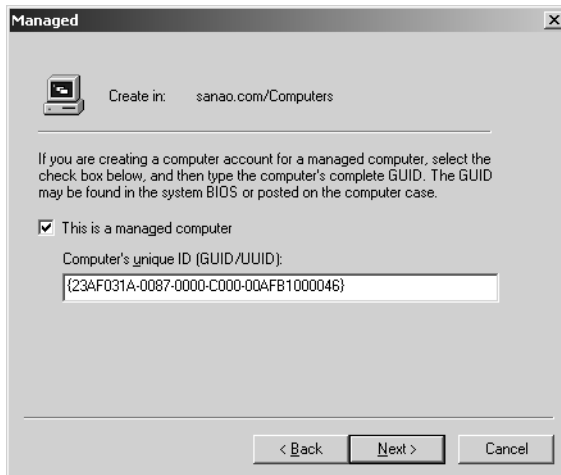
You can store the computer objects either in the Computers container or in various OUs in the domain. The latter option allows different OU-based group policies for different computers.

When you right-click the appropriate target OU and select New, Computer, you will launch a three-page or four-page creation wizard, the first page of which you see in Figure 3.18. Here you specify the name for the object, the downlevel name for the computer, and the user or group who can later join the computer to the domain. If the joining computer is running Windows NT, you must select the “pre-Windows 2000” check box. If the joining computer will be a Windows NT backup domain controller, you must select the “backup domain controller” check box.

Figure 3.19 shows the second page of the creation wizard. If you use Windows 2000, the pages beyond the first one will appear only if you have installed Remote Installation Services (RIS) to install Windows 2000 Professional computers.



**Figure 3.18** When you create a computer object, on the first page of the creation wizard you are prompted to specify the name for the object, the downlevel name for the computer, and the user or group who can later join the computer to the domain.



**Figure 3.19** On the second page of the creation wizard you can specify that this is a “managed computer” (to indicate that you will use Remote Installation Services, or RIS, “prestaging” for this computer) and enter the computer’s GUID.

## 196 Chapter 3 Managing OUs, Users, and Groups

---

**NOTE** Whether you get the additional wizard pages in Windows 2000 or not depends on which computer you are sitting at. For example, if there are two domain controllers in your domain (DC1 and DC2) and you have installed RIS on DC2, you will see the two additional pages if you are sitting at DC2 or any workstation. However, if you are sitting at DC1, you won't see the pages.

---

Computer manufacturers assign a unique GUID to each computer they sell. If you enter this GUID into Active Directory, it will help RIS to match a certain computer system to a certain computer object.

After you have bought a computer and turned it on for the first time to install Windows 2000 or Windows XP onto it, the RIS service sends the computer's GUID to a RIS server. This way, RIS can locate the correct computer object in Active Directory.

If you selected the "This is a managed computer" option on the wizard's second page, you will see a third page, which is shown in Figure 3.20. The last page displays the summary of your selections, and we don't show this screen.

---

**NOTE** The computer GUID shown in Figure 3.19 is not the same as the GUID that each Active Directory object has. Chapter 8 offers more in-depth treatment of object GUIDs.

---

**NOTE** You cannot specify the computer GUID or RIS server name for an existing computer object using the Users and Computers snap-in if you didn't specify "managed computer" when you first created the object. To edit properties directly, you need to use ADSI Edit or some other means. The aforementioned information is stored in the properties `netboot-GUID` and `netbootMachineFilePath`.

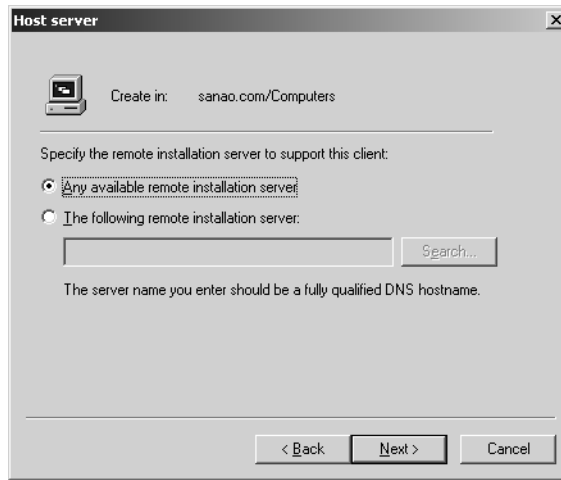
---

A computer object has several names, which are listed in Table 3.15.

### Setting Computer Object Properties

The Users and Computers snap-in shows you about 15 computer object properties, and you can set about 8 of them. Behind the scenes, a computer object may have 280 properties (228 in AD2000.)

Table 3.16 lists the properties in five tabs. We discuss a sixth tab, Member Of, later in this chapter in the "Administering Groups"



**Figure 3.20** If you selected the “This is a managed computer” option in the creation wizard’s second page (Figure 3.19), you will see a third page that enables you to specify a certain remote installation server. You can use this for load balancing, so that certain client computers (identified by the GUID) install Windows 2000 or Windows XP from a certain server.

section, and a seventh tab, Delegation, in Chapter 4. An eighth tab, Dial-in, relates to managing communication settings, so we don’t cover it in this book about Active Directory. We don’t include screen shots, because they would show just a number of text boxes. Many of the setting names are self-explanatory. Note that Windows Server 2003 also provides context-sensitive help for each of the settings.

### Other Operations to Manage Computer Objects

Other operations you can do to manipulate computer objects are move, delete, disable, and reset. You can also rename computers or start computer management to manage the computer corresponding to the object.

#### *Moving Computer Objects*

If you need to move a computer object from one OU to another, you do it in the same way you move users. When you are moving a computer within a domain, either (a) drag it to a new location with the mouse, (b) use cut/paste with the keyboard or mouse, or (c) right-click the

**198 Chapter 3 Managing OUs, Users, and Groups****Table 3.15** Name Properties of a Computer Object

Property	LDAP Name	Maximum Length	Required	Unique	Comments
Computer name	name (RDN) and cn (Common-Name)	64	X	Within OU	This becomes the object common name in the tree.
DNS name	dNSHostName	2048		In the world	The target computer updates this property automatically.
Computer name (pre-Windows 2000)	sAMAccountName	256 (schema rule), 20 (SAM rule)	X	Within the enterprise	This is the downlevel name of the computer, which is also the same as the computer NetBIOS name. Internally, Active Directory stores a dollar sign (\$) at the end of the name.

computer, select Move, and then choose the destination from the OU tree that opens up and click OK. Between domains in a forest you use another tool, such as the Support Tools command-line tool MoveTree, which is discussed in Chapter 6.

You can move several sibling objects at once by selecting them in the right-hand pane of the snap-in by using the Shift and/or the Ctrl key.

When you move computer objects

- Permissions that are assigned for the object being moved move with the object.
- Group policies and permissions that are inherited from above do not move with the object being moved. Instead, the moved object inherits the policies and permissions from its new location.

**Table 3.16** Properties of a Computer Object

Property	LDAP Name	Syntax *	Index	GC	Comments
<b>General Tab</b>					
Computer name (pre-Windows 2000)	sAMAccount-Name	Text (256 [schema rule], 20 [SAM rule])	X	X	This is the downlevel name of the computer, which is also the same as the computer NetBIOS name. Internally, Active Directory stores a dollar sign (\$) at the end of the name.
DNS name	dNSHostName	Text (2048)		X	
Role	userAccount-Control	Two choices	X	X	Bit 0x2000 indicates a “Domain controller”; bit 0x1000 indicates a “Workstation or server”.
Description	description	Text (1024)		X	
Trust computer for delegation	userAccount-Control	Yes/no	X	X	This setting is described in Chapter 4 in the “Impersonation and Delegation” section. Note that when the domain is on the Windows Server 2003 functional level, this setting appears on the Delegation tab.

*(continued)*

\* If the syntax is Text (i.e., a string of Unicode characters), we indicate also the maximum number of characters in the property (e.g., 1,024).

**200 Chapter 3 Managing OUs, Users, and Groups****Table 3.16** Properties of a Computer Object (*cont.*)

Property	LDAP Name	Syntax *	Index	GC	Comments
<b>Operating System Tab</b>					
Name	operating-System	Text			A read-only text such as "Windows Server 2003."
Version	operating-System-Version	Text			A read-only text to indicate the normal version, such as "5.2" (Windows 2000 is "5.0", Windows XP is "5.1", and Windows Server 2003 is "5.2"), and the more precise version (i.e., build number), such as "3790."
Service Pack	operating-System-ServicePack	Text			A read-only text to indicate whether or not you have installed any service packs on the machine, such as "Service Pack 1."
<b>Location Tab</b>					
Location	location	Text (1,024)	X	X	

**Table 3.16** Properties of a Computer Object (*cont.*)

Property	LDAP Name	Syntax*	Index	GC	Comments
<b>Managed By Tab</b>					
Managed By	managedBy	DN; you select a user or contact from a list			The user or contact you select gets no permissions for the computer. This setting is purely informational. The other fields on the tab are the manager's properties. Note that this setting is not related to the "This is a managed computer" check box that you saw in the creation wizard.
<b>Remote Install Tab**</b>					
Computer's unique ID	netbootGUID	Binary (text in the user interface)	X	X	Same as the computer's GUID. It helps when using RIS, and it signifies a managed computer.
Remote Installation server	netboot-Machine-FilePath	Text		X	This property specifies the DNS name of the selected installation server.
Server Settings	N/A	N/A	N/A	N/A	This button takes you to the properties of the server object.

\*\* The Remote Install tab is present only if you created the object for a "managed computer" by checking the box on the second page of the creation wizard. If you are using Windows 2000, the tab is present only when you are sitting at the correct computer, as explained in the preceding section, "Creating Computer Objects."

### ***Deleting Computer Objects***

You delete an object by right-clicking it and selecting Delete or by selecting the object and pressing the Delete key. Because there is no Undo option, a safety mechanism asks you to confirm the deletion.

A computer object is a security principal like a user object. Therefore, if you delete a computer object and then re-create it, the new object doesn't have the memberships or permissions of the old one.

If you delete a computer object, the corresponding computer is no longer part of the domain. Therefore, no one can log on to the computer using a domain user account.

### ***Disabling Computer Accounts***

You can disable the computer account by right-clicking the computer object and selecting Disable Account. Doing so will prevent users sitting at that computer from logging on using a domain user account.

You cannot disable a domain controller.

### ***Resetting Computer Accounts***

When a Windows NT/2000/XP/Server 2003 computer that is a member of a domain starts, the computer logs on to the domain using the computer account and some password known to the machine. After this, a user sitting at the computer can enter his username and password to log on to the domain.

The aforementioned machine logon sets up a *secure channel*, which enables the member computer to communicate with a domain controller to exchange user and password information. For example, if the computer account password stored in the local computer (called *LSA secret*) doesn't match the one stored in Active Directory, authentication to the domain is not possible, and the user will receive an error such as the one shown in Figure 3.21.

An administrator can solve the problem by using the Reset Account context menu item on the corresponding computer object. Resetting a computer account resets its password to the initial value, which is "computername\$" (without quotes). In addition, the member computer must be joined to a workgroup and then joined to the domain again.



**Figure 3.21** If the member computer cannot establish a secure channel with a domain controller, the user receives an error message such as the one shown here and is not able to log on using a domain user account.

---

**NOTE** You can reset a computer account also with the DSMod Computer command and `-reset` option. In addition, Support Tools includes two command-line utilities, NetDom and NLTest, which you can use to reset computer accounts, among other things.

---

### **Managing Computers**

When you right-click the computer object and select Manage, the Computer Management snap-in starts and sets the focus to the corresponding computer. This way you can manage its system tools, storage, server applications, and services.

### **Renaming Computers**

You rename a Windows 2000/XP workstation or a Windows 2000/Server 2003 member server using the Control Panel of that computer. Select System, then the Computer Name tab, and finally the Change button. Once you enter a new name and click OK, you are prompted for the name of a domain user who has permission to change the name of the workstation or member server, as well as that user's password.

This operation renames the computer (i.e., the NetBIOS name and DNS name) and changes the common name and the pre-Windows 2000 name of the computer object.

Renaming domain controllers was discussed in Chapter 2.

## Administering Groups

---

Managing users, inetOrgPersons, contacts, and computer objects is usually much more effective when you treat them in groups than when you treat them individually. Whether you need to send e-mail or assign permissions for a printer, you most often want the target to be several users instead of just one. When you need the same group again, the fact that you already have it created will save you work. Of course, there is no laborsaving benefit if you create a group and then use it only once.

Groups are extremely handy and you really cannot manage a network without them. However, you use them mainly for assigning permissions and group policies. Specifically, you cannot use groups for the following purposes:

- Setting properties of several users, inetOrgPersons, contacts, or computer objects, or applying properties for them
- Moving or deleting several users, inetOrgPersons, contacts, or computer objects

In addition to administrators, end users can use Active Directory groups, usually as distribution lists. Table 3.17 describes in more detail the purposes for which you can use groups.

---

**NOTE** Because inetOrgPerson objects behave identically to user objects, each time we discuss users in this “Administering Groups” section, that discussion applies also to inetOrgPersons. For brevity, we don’t show inetOrgPersons separately.

---

### Group Types

You can create two types of groups in Active Directory: *security groups* and *distribution groups*. Both types can have users, inetOrgPersons, contacts, and computer objects as members. In addition, AD2003 introduces new basic and query-based application groups. The Windows operating system, however, doesn’t currently use them. We explained these new application groups briefly in Chapter 1.

**NOTE** AD2003 allows any object type to be a group member. You don't need such a feature in normal user administration, but a directory-enabled application could use it.

Table 3.17 illustrates that security groups have two natures, but distribution groups have only one. Thus, distribution groups have a subset of security group features.

**Table 3.17** The Nature of Security and Distribution Groups

Nature	Security	Distribution	Purpose
<p><b>Security nature</b></p> <p>Group is a security principal, which Windows uses to determine permissions.</p>	X		<p>Assign permissions, and possibly audit settings, for folders, files, and Active Directory objects.</p> <p>Assign group policies (not directly, but by assigning permissions for a certain Group Policy only to some group).</p> <p>Other miscellaneous use, such as check the group membership in a logon script and then apply some commands, in case the user was in that group.</p>
<p><b>Application nature</b></p> <p>Group is available to directory-enabled applications. Windows doesn't use it, but any application may use it.</p>	X	X	<p>Send e-mail (i.e., the group operates as a distribution list).</p> <p>When using a directory-enabled application, use the group for whatever purpose the application needs.</p>

## 206 Chapter 3 Managing OUs, Users, and Groups

---

**NOTE** Even though labeled “application nature,” an application could use the distribution feature also for some security use. For example, you could have an application that controls the doors of your company. The application could open a certain door for a user if he is a member of a certain distribution group. This Note is related to Table 3.17.

---

**IF YOU KNOW NT** Security groups are the traditional groups that existed in Windows NT. Distribution groups were introduced with Active Directory.

---

As Table 3.17 indicates, a security group has all the features of a distribution group, and it can also be used for assigning permissions. This leads to the following question: Why do we use distribution groups at all, if they are less capable? The reason is that they are a little “cheaper” than security groups in terms of the logon process.

When a user logs on to the network or accesses the resources of a server for the first time, Windows builds an *access token* for that user. An access token is a list in RAM that contains the user’s identity and the groups that the user belongs to. But it doesn’t contain any distribution groups. Because distribution groups are not needed when determining access, they are not needed in the access tokens. This, in turn, leads to a somewhat faster logon process and a smaller access token in memory. Of course, you probably won’t notice the difference in a small network.

**NOTE** Access tokens are discussed in more detail in Chapter 4.

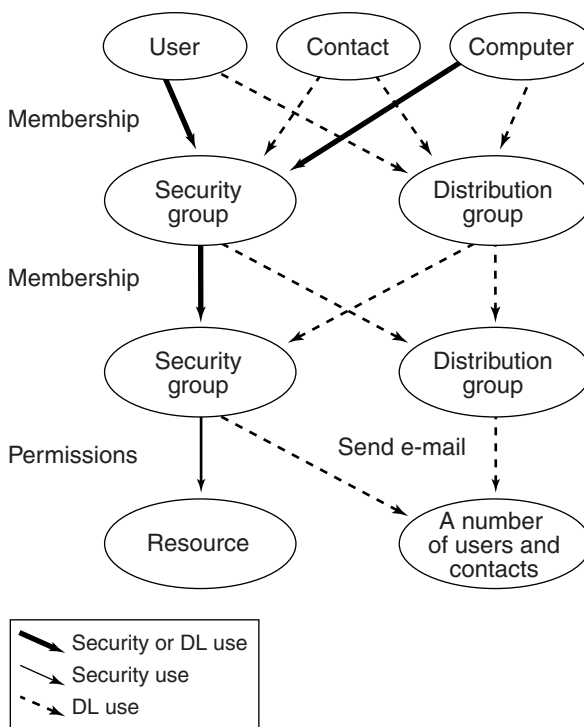
---

On the other hand, as long as you don’t have any directory-enabled applications, you cannot use distribution groups, even though you can create them. Remember that Windows doesn’t use them. This means that it’s quite possible that you need to create only security groups, even though they are a little more “expensive.”

Figure 3.22 summarizes the features of the two group types.

**NOTE** A contact is never able to log on, so it never gets an access token, and it cannot access resources. Therefore, it is not part of the security nature of groups, even though it can be a member of a security group.

---



**Figure 3.22** The solid lines in this figure represent the security nature of groups. All lines except the thin solid line in the lower-left corner represent the application nature of groups.

**NOTE** Figure 3.22 shows groups as members of other groups. Depending on the domain functional level and group scopes (discussed in the next section), not all groups can be members of all other groups.

### Group Scopes

In addition to the two group types (security and distribution), groups are divided into three scopes: *global groups*, *universal groups*, and *domain local groups*. The group scope indicates if the group can accept members from other domains and if it can be used in other domains.

Group scopes are not very important if you have only one domain. In that case, you can do just fine with only universal groups, unless you anticipate having several domains at some later time.

Group nesting means that groups can be members of other groups; that is, a group is inside another group. Group scopes and nesting behave differently depending on the domain functional level. Regarding group functionality, the four levels fall in two categories. The first category could be called “Windows NT compatible” (including the levels Windows 2000 mixed and Windows Server 2003 interim), and the other category could be called “pure Active Directory” (including the levels Windows 2000 native and Windows Server 2003).

We will first explain the Windows NT-compatible case. Even if you have already raised your domain functional level and/or plan to use only “pure Active Directory,” you should read the section about mixed mode. In this section we explain many principles of how to use groups in administration, regardless of the domain functional level.

### ***Group Scopes in Windows NT-Compatible Functional Levels***

Distribution groups in Windows NT-compatible functional levels work just like distribution groups in pure Active Directory levels. Consequently, we’ll discuss distribution groups in the next section, which is about pure Active Directory levels.

---

**NOTE** Contact objects don’t quite follow the containment rules that we will present from this point on. A global group can have user and computer object members only from its own domain, but it can have contact object members also from other domains.

---

Security groups in Windows NT-compatible functional levels work like the groups in Windows NT. You can have global and domain local security groups, but you can’t have universal security groups.

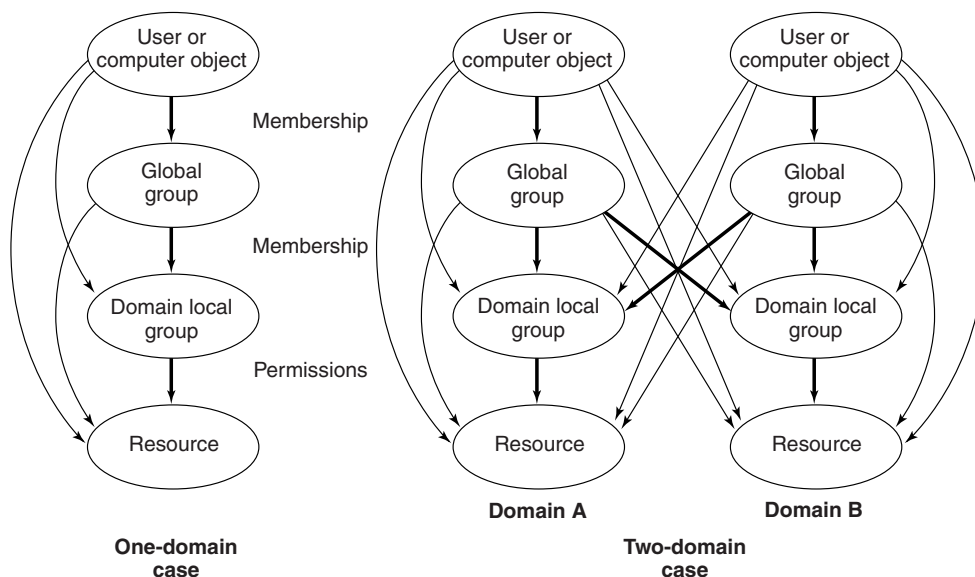
You cannot freely nest security groups in Windows NT-compatible functional levels, but you can put global groups as members into domain local groups, as Figure 3.23 illustrates.

---

**NOTE** We don’t include contact objects in the figures from now on because we are concentrating on the security nature of the groups.

---

In the one-domain case in Figure 3.23, you can draw an arrow from any circle to any other, as long as you move downward. In the two-domain case, only the two upper circles are visible in the other domain, and only the two lower circles accept arrows from the other domain.



**Figure 3.23** In Windows NT-compatible domain functional levels you can put users and computer objects in global groups, put global groups in domain local groups, and then give permissions to domain local groups. This preferred arrangement is indicated in the image by thick lines. You can also use the shortcuts indicated by the thin lines.

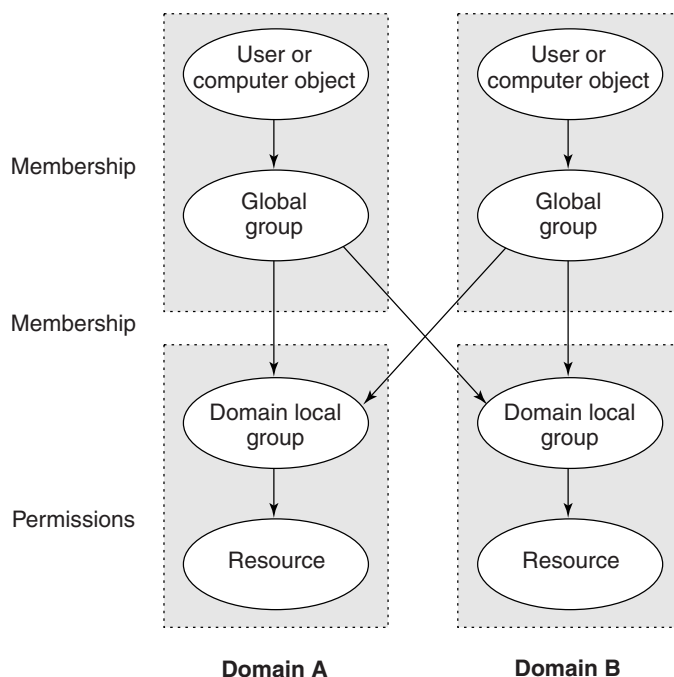
Remember that normal trust relationships in Active Directory are bidirectional. In the case of two domains, domain A trusts domain B and vice versa. Consequently, a domain A global group can be a member of a domain B domain local group, and a domain B global group can be a member of a domain A domain local group. In other words, if you looked at Figure 3.23 in a mirror, you would see a similar figure.

Because Figure 3.23 has quite a few arrows, to simplify the two-domain case, Figure 3.24 shows only the preferred (thick) arrows.

The thin lines (shortcuts) are less desirable for the following reasons:

- By giving permissions to one group instead of 200 users, you get dramatically shorter permission lists. This saves disk space, speeds up permission evaluation, and is easier to manage. For example, when your organization hires a new employee, she will get all the needed permissions when you add her to a few groups. The worse alternative would be to go through all server folders and add permissions to this new user.

## 210 Chapter 3 Managing OUs, Users, and Groups



**Figure 3.24** Global groups are usually associated with people (and computer objects). Domain local groups are resource oriented. The dotted boxes symbolize this division.

- You should use global groups to group users (and computer objects). These groups are also a level of isolation. If the user list changes, the groups stay the same, and therefore hide the changes from the lower levels. This is especially valuable in a multidomain situation. If one domain gets a new user, the administrators in other domains don't have to do anything, because they already have the appropriate global groups as members in the appropriate domain local groups.
- You should use domain local groups for resource-oriented purposes. You often create a domain local group either for one resource or for a certain type of resource, such as all color printers. This way, if a new group of users needs access to all the color printers, you can just make this group a member of the rColorPrintersPrint domain local group, instead of giving permissions for 17 color printers individually.

---

**NOTE** The first *r* in the rColorPrintersPrint domain local group name indicates that it is a resource-oriented group. *Print* indicates that this group has the print permission for the corresponding printers. You can use these kinds of naming conventions at will.

---

**NOTE** A domain local group is valid only in the local domain. You can use such a group to assign permissions for an Active Directory object, but when that object is replicated to global catalog servers that are members of other domains, your Allow or Deny permissions are not valid there. If a user subsequently queried this object using one of those “foreign” global catalog servers, she might get access you did not intend (because a Deny permission is not in effect). Therefore, Microsoft recommends that you shouldn’t use domain local groups to assign permissions for Active Directory objects in a multidomain forest. Typically, the term *resource* you see in the figures on these pages refers to a shared folder or printer on a server, instead of an Active Directory object.

---

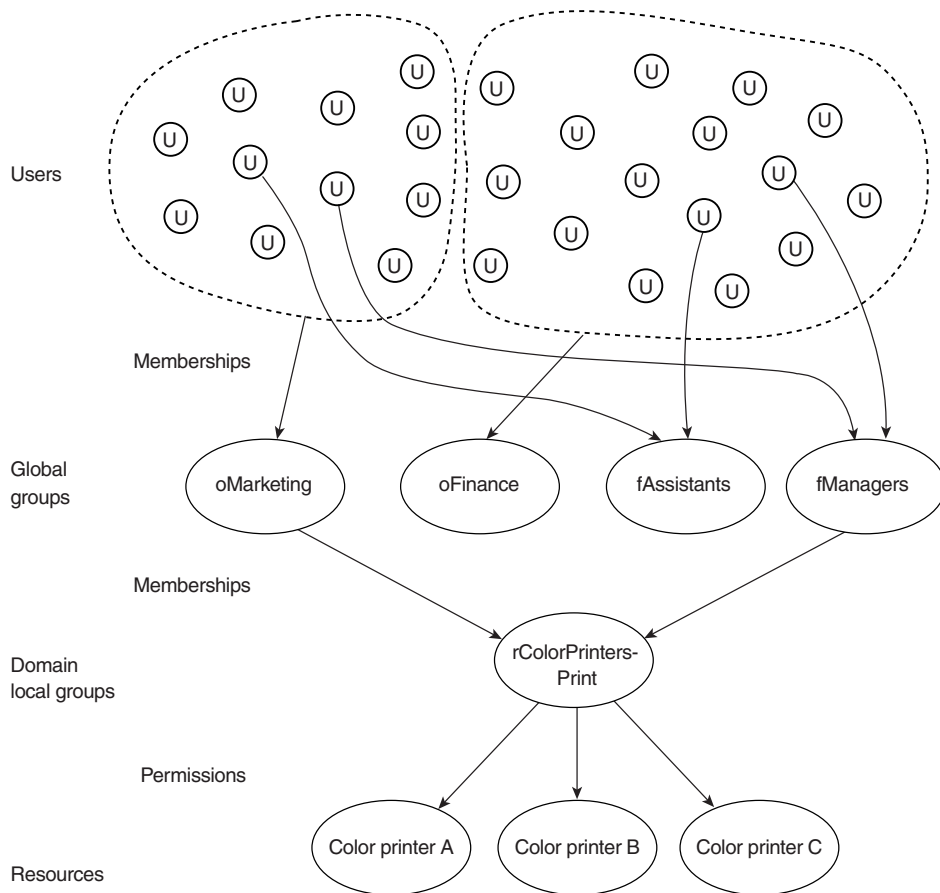
### **Example of Group Usage**

We present in this section a basic example of group usage. We want to limit the number of people who can print on the color printers in our domain, so we perform the following steps.

1. When we deployed Active Directory, we established certain global groups to group the users in our domain. We put the users in groups based on the organizational structure (oMarketing and oFinance), as well as functional categories (fAssistants and fManagers). Note that we cannot use OUs for anything here.
2. Now we create a new local group, rColorPrintersPrint, and give that group permission to print to each color printer. We have three color printers and we have to assign the Print permission for each printer individually.
3. As the final step, we assign appropriate global groups as members of the rColorPrintersPrint group. We want everyone in the marketing department and all managers to be able to print in color.

Figure 3.25 illustrates the result.

## 212 Chapter 3 Managing OUs, Users, and Groups



**Figure 3.25** We have grouped our users into global groups, so we don't need to handle individual users. We give the actual Print permission to a domain local group and then assign appropriate global groups as members of this domain local group.

**NOTE** If a workstation or member server, instead of a domain controller, handles one of the color printers, the domain local group cannot be used while we are still in one of the Windows NT-compatible functional levels. Once we raise the level to one of the pure Active Directory levels, we can start using domain local groups in workstations and member servers.

If you have only one domain and you feel that you don't need two levels of groups, you may skip either level. Either you can make users (and computer objects) members of domain local groups, or you can give permissions directly to global groups.

The domain local group in Figure 3.25 may seem unnecessary. However, imagine that you have 17 color printers and, along with marketing personnel and managers, you want to allow assistants to print to them. With the domain local group you can do this quickly: You only need to put fAssistants as a member in rColorPrintersPrint. Without the domain local group, you would need to open the properties dialog box of 17 printers in quite a few servers and assign permissions to fAssistants individually in each dialog box.

See also the "Efficient Group Nesting" section later in this chapter and the associated Figure 3.38.

### **Group Scopes in Pure Active Directory Functional Levels**

In the pure Active Directory functional levels, you can have any of the three group scopes in either of the two group types—that is, there are six possible combinations. Unlike in the Windows NT-compatible functional levels, now you can have universal security groups.

Security groups and distribution groups now work the same way with each other. Therefore, we don't need to make a distinction between them, and we don't discuss them separately here.

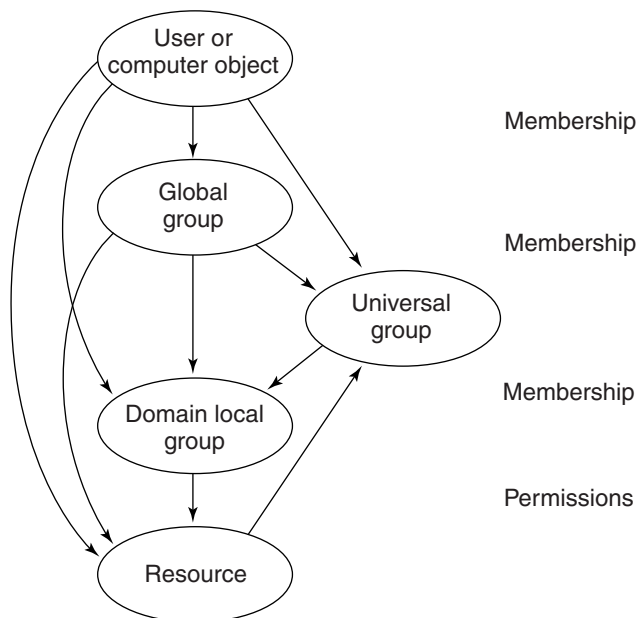
In the Windows NT-compatible functional levels, global groups are the upper level and domain local groups are the lower level. In the pure Active Directory functional levels, universal groups are a third level between the two earlier levels. A group from a higher level can be a member in a group from a lower level, as Figure 3.26 illustrates.

---

**NOTE** Figure 3.27, Figure 3.28, and Figure 3.29 are more complex than earlier images. To be as clear as possible, we don't show users and computer objects on the top or resources on the bottom in those three figures. You can still imagine them to be there.

---

Deciding how to use all these groups in the pure Active Directory functional levels is more complicated than in the Windows NT-compatible functional levels. We delve into this discussion in the



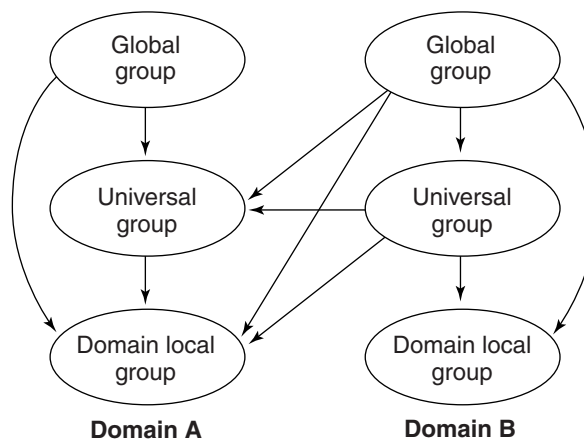
**Figure 3.26** In the pure Active Directory functional levels, you have three levels of groups. Any upper-level object can be a member of any lower-level object. This figure illustrates the situation in one domain.

“Planning Groups” section later in the chapter. We’ll just mention the three basic strategies here:

- Forget universal groups and use only global and domain local groups, as described in the preceding section about the Windows NT-compatible functional levels.
- Use only universal groups.
- Use all three levels (and pray that you know what is going on in Active Directory). If you have several domains and sites, you will probably need all three levels.

**NOTE** Because there are now three possible strategies for using groups, Figure 3.26 does not indicate (with thick lines) a preferred path.

Figure 3.27 introduces a second domain. It illustrates that global groups cannot accept members from other domains (except contacts),



**Figure 3.27** When crossing domain boundaries, global groups cannot accept members from other domains, and domain local groups cannot be used in other domains. Universal groups, however, have no such restrictions.

and domain local groups cannot be used in other domains, but universal groups don't have either of these restrictions.

**NOTE** As you may remember from the Windows NT-compatible section, the arrows between domains should be symmetrical. To keep Figure 3.27 uncluttered, we do not show the arrows from domain A to domain B.

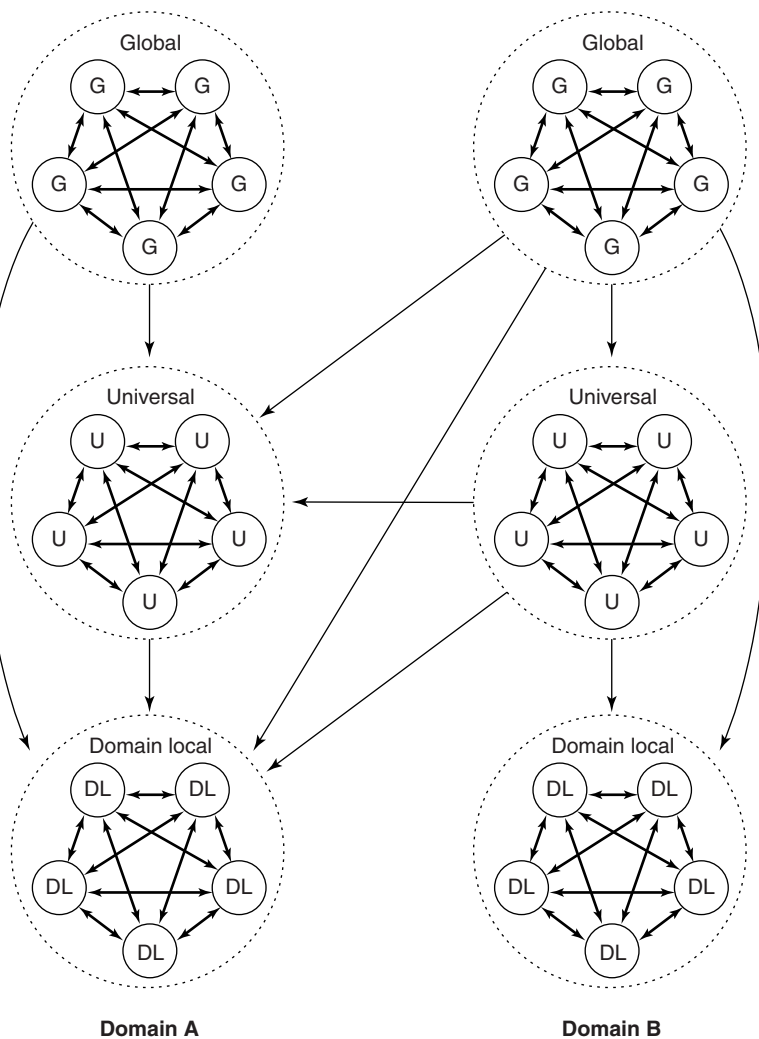
If one of the domains in Figure 3.27 were in one of the Windows NT-compatible functional levels and the other were in one of the pure Active Directory functional levels, the image would still be accurate. Obviously, one of the domains couldn't have universal security groups, but after having removed that, all the remaining arrows would be valid. For example, if domain A were in the Windows 2000 mixed functional level and domain B were in the Windows Server 2003 functional level, the domain local groups in domain A would accept both global and universal groups as members from domain B.

The figures so far have shown only one group of each scope in each domain. In reality, you will have many groups of each scope. In the pure Active Directory functional levels, you can freely nest groups of the same scope. Global group A can be a member of global group B, which is a member of global group C, which is a member of global group D, and

## 216 Chapter 3 Managing OUs, Users, and Groups

so on. In other words, any group can be a member of any other group of the same scope in the same domain.

Building on Figure 3.27, Figure 3.28 shows five groups of each scope in each domain.



**Figure 3.28** Within each scope in each domain, any group can be a member of any other group. From one scope or domain to another, groups that can be a member of other groups are indicated with an arrow.

---

**NOTE** In Figure 3.28, the arrow from the global groups in domain A to the universal groups in domain A symbolizes that any of the upper five groups can be a member of any of the lower five groups. An actual representation would have 25 arrows, but we use just one. These 25 arrows would be needed 14 times, between each scope of groups in each domain. To give you a clear image, we use only 10 arrows instead of 350.

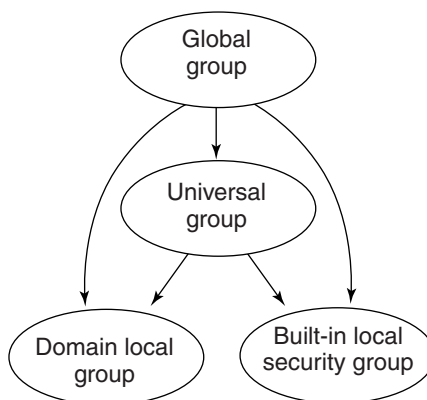
---

**NOTE** Again, in Figure 3.28, arrows from domain A to domain B were left out to make the image clear.

---

### **Built-in Local Groups**

The last aspect of group scopes concerns the built-in local security groups (Administrators, Account Operators, and so on) that reside in the Builtin container. Technically, they belong to a different “domain”—the Builtin domain—therefore, you cannot nest domain local groups with built-in local security groups or vice versa. Figure 3.29 illustrates this concept.



---

**Figure 3.29** Built-in local security groups belong technically to a different “domain.” Therefore, you cannot nest them with domain local groups.

## Managing Groups

Now you are ready to create and manage groups. Before you implement the groups in your production environment, you should first read the “Planning Groups” section of this chapter.

Managing groups includes the following tasks:

- Creating groups of different types and scopes
- Changing the type or scope of a group
- Managing group memberships
- Setting the primary group of a user
- Setting group properties
- Moving, renaming, and deleting groups
- Sending e-mail to groups

When you create and manage groups, we suggest that you visualize your groups in the way that we have presented groups in the figures in this book. Having a clear visual image of them in your head, or even on paper, will help. The user interface of the Users and Computers snap-in doesn't indicate graphically that you should put users in global groups, global groups (perhaps) in universal groups, and so on.

### Creating Groups

You create groups with the Users and Computers snap-in just as you create any other object. Right-click the target OU and select New, Group. Figure 3.30 is a screen shot of the dialog box that appears. Because you cannot assign permissions to an OU, the first group you create is probably a global security group with the same name as the OU. When you add each user of the OU to this group, you can give him or her permissions with the help of this group.

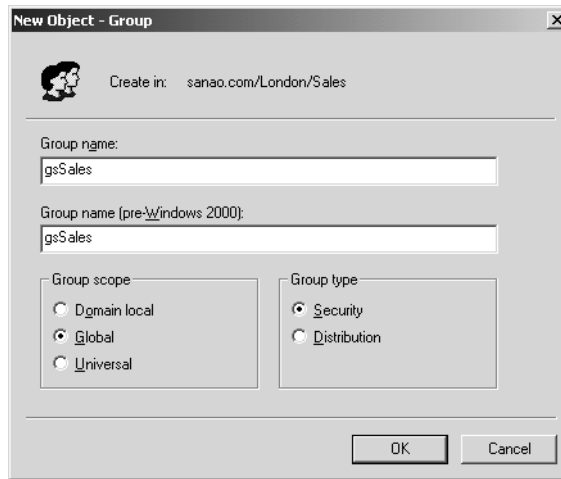
Table 3.18 describes the two names shown in Figure 3.30.

---

**NOTE** Distribution groups are created for directory-enabled applications. It is unlikely that those applications use the pre-Windows 2000 name. However, you must define it for every distribution group.

---

Many of the dialog boxes in the Users and Computers snap-in give no hint of the scope or type of existing groups. Therefore, you might consider adding your own hint—for example, add “gs” to the name, with “g” standing for “global” and “s” standing for “security” group. With



**Figure 3.30** When you create a group, the first dialog box that appears calls for naming the group and assigning its scope and type. The first group is likely to have the same name as the OU.

**Table 3.18** Name Properties of a Group Object

Property	LDAP Name	Maximum Length	Required	Unique	Description
Group name	name (RDN) and cn (Common-Name)	64	X	Within OU	This becomes the object common name in the tree.
Group name (pre-Windows 2000)	sAMAccountName	256	X	Within domain	This name appears on non-Active Directory computers and software, such as the old User Manager. Despite its label, this name can be used throughout Windows 2000 and later.

## 220 Chapter 3 Managing OUs, Users, and Groups

domain local groups, you could use “l” instead of “d” to not confuse them with distribution groups.

You could also use letters to indicate whether the group was created by organization, by functionality, by resource, or by some other criteria. Table 3.19 gives some suggestions on how to use these symbol letters.

In addition to making names more descriptive, these symbols sort similar groups sequentially when the user interface is using an alphabetical list.

Table 3.19 presents examples of letters that indicate scope and type, such as “gs,” and letters that indicate logical grouping, such as “o.” Of course, you can use both types, but be aware that confusion can arise if you use too many identifiers like these.

### **Changing Group Type or Scope**

The Windows NT-compatible functional levels don't enable you to change group type or scope. Raising the functional level to either pure Active Directory alternative (Windows 2000 native or Windows Server 2003) enables these changes, with two restrictions (see Figure 3.31).

**Table 3.19** Symbol Letters for Group Names

Letter(s)	Examples	Meaning
gs	gsSales	Global security groups.
us	usSAPUsers	Universal security groups.
ls	lsSAPUse lsColorPrint	Domain local security groups. The first group has permissions to use SAP software, and the second group has permissions to print in color.
o	oDirectSales oChannelSales	Groups created according to the organizational structure (which don't match OUs).
ou	ouSales	Groups created to match OUs.
f	fSalesmen fAssistants	Groups created according to function (for example, salesmen from all OUs).
r	rSAPUse rColorPrint	Groups created for resources. Because these are usually domain local groups, this example has the same group names as the “ls” example.



**Figure 3.31** You can change a group scope to and from a universal group, but you can't change scope directly from a domain local group to a global group or vice versa.

- If the new type or scope would lead to an illegal situation in terms of memberships, the change is obviously forbidden. For example, if your domain local group has other domain local groups as members, you cannot change it to a universal group. Universal groups cannot have domain local groups as members.
- You cannot change a domain local group to a global group or vice versa, except via a universal group.
- In a multidomain forest, you cannot change a universal group to a domain local group, unless you perform the change with a domain controller that is a global catalog server. Consequently, if none of the domain's domain controllers is a global catalog server, you cannot perform the change at all.

### ***Managing Group Memberships***

Each user, contact, computer, and group is a “member” of only one OU. At the same time, each can be a member of several groups, because a group membership is just a group property; it is not part of the tree structure.

A user's or computer's new group membership becomes effective in each server or workstation when the user or computer authenticates to that server or workstation the next time. This typically takes place when the user logs on or the computer is restarted, but it also takes place if the user creates a connection to a server where there was no existing connection.

The Users and Computers snap-in allows you to manage group membership in three ways:

- The Members tab of the group
- The Member Of tab of the (incoming) member
- The “Add to a group” function

## 222 Chapter 3 Managing OUs, Users, and Groups

---

You cannot drag objects over groups to become members of them.

---

**WARNING** If you have an AD2000 forest or an AD2003 forest running on the Windows 2000 functional level, you should take into account the following warning: If you delegate group management to assistant administrators, you should advise them to modify group memberships only on one domain controller (perhaps the PDC emulator). All members of a group are stored in one multivalued property. If that member list is modified on two domain controllers simultaneously (within replication latency), one of the two changes will be lost.

---

---

**WARNING** You could give users the permission to “Add/Remove self as member” of some group. For the reason and scenarios given in the previous warning, some changes could be lost, and the risk would be quite great if all users could modify membership themselves.

---

---

**NOTE** Because all members of a group are stored in one multivalued property, there is a limit of 5,000 members in one group. The limit doesn’t apply in the Windows Server 2003 (or Windows Server 2003 interim) forest functional level.

---

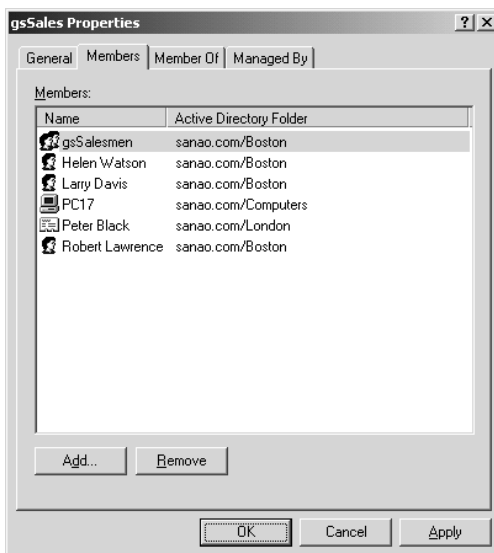
### ***The Members Tab of the Group***

The first way to manage groups is through the Members tab. When you right-click a group, select Properties, and then click the Members tab, you’ll see a list of the members of the group, as Figure 3.32 shows.

As you would guess, you remove members by selecting them and clicking the Remove button.

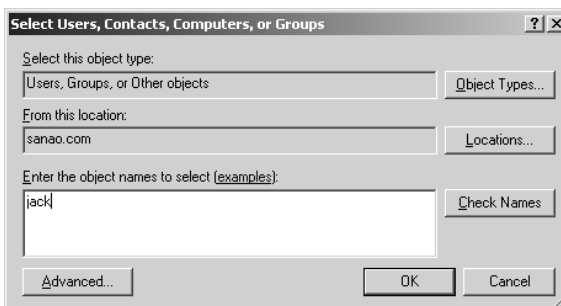
To add members to a list, click Add. Another dialog box opens, where you can enter the objects to be added as members (see Figure 3.33). In the “From this location” field choose the domain or folder (or Entire Directory).

After you have typed the new member names in the text box, you can check whether they are valid with the Check Names button. If you want to type several names, you must separate them with semicolons. If more than one object matches the name you typed, clicking OK or Check Names brings up the dialog box in Figure 3.34.



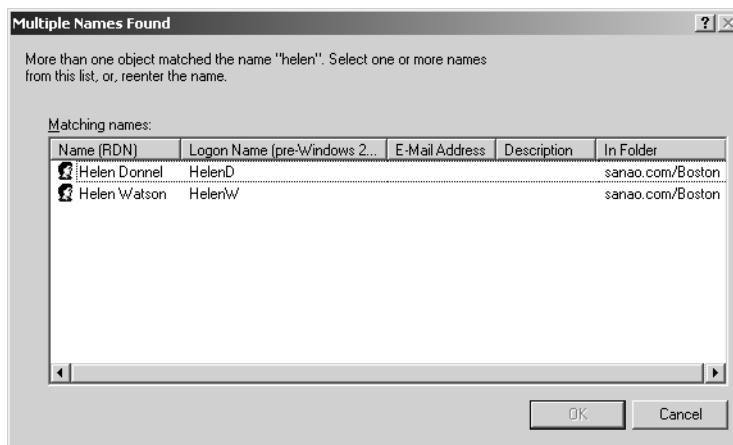
**Figure 3.32** The gsSales group currently has users, contacts, computers, and other groups as members.

**NOTE** When you added group members, Windows 2000 showed by default a list of possible members. Windows Server 2003 doesn't show this, because in a large domain it is time consuming. You can display the list, however, by clicking Advanced in the dialog box shown in Figure 3.33 and then clicking Find Now in the dialog box that opens.



**Figure 3.33** To enter new members for a group, first choose the domain or folder (or Entire Directory) in the “From this location” field and then enter the names, using semicolons as separators. If you want, you can click Check Names before clicking OK.

## 224 Chapter 3 Managing OUs, Users, and Groups



**Figure 3.34** If you type a name that matches several objects, you are prompted to select the name you intended.

### ***The Member Of Tab of the Incoming Member***

User, contact, computer, and group objects have a Member Of tab, which shows the groups that the object belongs to. If you have several domains in the forest, however, from other domains the tab shows just universal groups.

The Members Of tab and consequent dialog boxes work in the same way as the Members tab and consequent dialog boxes.

### ***Add to a Group Function***

The context menu of each user and contact (accessed with a right-click) has an “Add to a group” option. When you select this menu item, you can choose the group in which to place the selected object or objects.

In Windows 2000, the “Add to a group” option is a menu item also for each OU. In this case, you can make all users and contacts in the OU members of the group. If the OU has child OUs, you can choose for each one whether to include users and contacts in them as well.

### ***Setting a User’s Primary Group***

Each user and computer object’s Member Of tab includes a setting for a *primary group*. You probably won’t need this setting, because it is used only by the POSIX subsystem (i.e., when running a kind of UNIX application in Windows) or by Apple Macintosh workstations.

The default primary group of a user is Domain Users, and the default primary group of a computer is Domain Computers. If needed, you can change this setting to some other global or universal security group.

You cannot remove an object from its primary group. Therefore, if you want to move a user out of Domain Users, you first must change that user's primary group to something else.

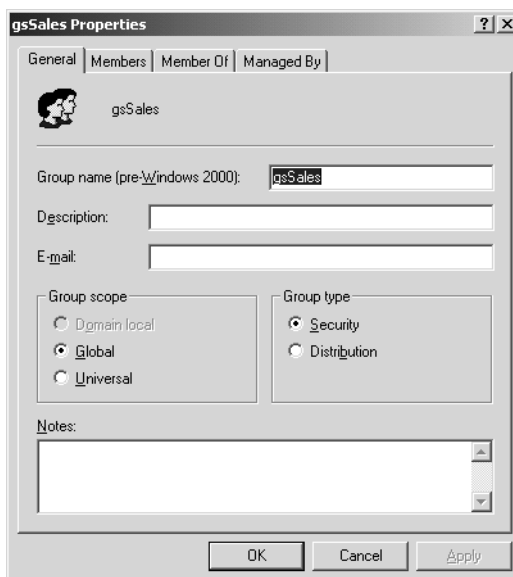
---

**NOTE** The primary group of a user is not stored in the `members` property of the group, but rather in the `primaryGroupID` property of the user. Consequently, the 5,000-member maximum of the Windows 2000 forest functional level doesn't apply to primary groups, which means that you could have 100,000 users (or more) in your domain and they could all be members of Domain Users.

---

### Setting Group Properties

Behind the scenes a group object may, by default, have 132 properties (107 in AD2000). The user interface displays only a few of them, as shown in Figure 3.35.



**Figure 3.35** There are not many properties that you can set for a group object.

**226 Chapter 3 Managing OUs, Users, and Groups**

Table 3.20 lists the properties of group objects that are visible in the Users and Computers snap-in other than group type, scope, and members, which we discussed in the previous sections. The settings are mostly self-explanatory. Note that Windows Server 2003 also provides context-sensitive help for each of the settings.

**Table 3.20** Properties of a Group Object

Property	LDAP Name	Syntax*	Index	GC	Comments
<b>General Tab</b>					
Description	description	Text (1,024)		X	
Group name (pre-Windows 2000)	sAMAccountName	Text (256)	X	X	This name appears on non-Active Directory computers and software, such as the old User Manager. Despite its label, this name can be used throughout Windows 2000 and later.
E-Mail	mail	Text (256)	X	X	
Comments	info	Text (1,024)			
<b>Managed By Tab</b>					
Managed By	managedBy	DN** you select a user or contact from a list			The user or contact you select doesn't get permission for the group. This setting is purely informational. The other fields on the tab are the manager's properties.

\* In the Syntax column, Text (256) means a text field with a maximum of 256 Unicode characters.

\*\* DN = distinguished name.

**Table 3.20** Properties of a Group Object (*cont.*)

Property	LDAP Name	Syntax *	Index	GC	Comments
Manager can update membership list	nTSecurity-Descriptor	N/A		X	The Windows Server 2003 version of the Users and Computers snap-in contains this new check box. If you check this setting, the manager gets permission to modify the member property of the group.

### ***Moving Groups***

You move groups between OUs just as you move other objects. When moving a group within a domain, either (a) drag it to a new location with the mouse, (b) use cut/paste with the keyboard or mouse, or (c) right-click the group, select Move, and then choose the destination from the OU tree that opens up and click OK. To move groups between domains in your forest, you use another tool, such as the Support Tools command-line tool MoveTree. It is discussed in Chapter 6.

You move several sibling objects at once by selecting them in the right-hand pane of the snap-in and using the Shift and/or Ctrl key.

When you move groups

- Permissions that are assigned for the object being moved move with the object.
- Permissions that are inherited from above do not move with the object being moved. Instead, the object inherits new permissions in its new location.

### ***Renaming Groups***

You rename a group either by right-clicking it and selecting Rename or by selecting the group and pressing F2. After you type the new name, press

## 228 Chapter 3 Managing OUs, Users, and Groups

---

Enter. Because groups also have a pre-Windows 2000 name, a dialog box appears that gives you a chance to change that name, too.

### ***Deleting Groups***

You delete a group by right-clicking it and selecting Delete or by selecting the group and pressing the Delete key. Because there is no Undo, as a safety mechanism, you must confirm that you want to delete the group.

Like a user, a group is a security principal. Therefore, if you delete and then re-create it, the new object doesn't have the memberships or permissions of the old one.

### ***Sending E-mail to Groups***

If the group has an e-mail address defined, you can send it e-mail with the "Send mail" operation in the context menu. Naturally, you need an e-mail application for this feature to work.

### **Planning Groups**

Now you know group mechanics and properties, so you can use this knowledge to decide what the best way is to use groups effectively for a specific network in terms of manageability, administrative burden, and cost to network efficiency.

Planning groups involves deciding on group names, types, and scopes.

- It often pays to use letters in group names that indicate the kind of group it is, as explained earlier in this chapter.
- As explained earlier, because of access tokens, you should use distribution groups when you don't need the security feature but intend just to use the group with some directory-enabled application.
- This section concentrates on group scopes and describes three strategies for using them.

Before we discuss the three strategies, we need to study universal groups a little more.

### **Universal Groups Revisited**

Recall that universal groups don't have the limitations of global or domain local groups. This prompts the following question: Why not use only the most feasible (i.e., universal) groups? Actually, Microsoft originally planned Active Directory to have only universal groups, not global or domain local groups. But the universal groups introduce extra cost, so Microsoft brought along the other two scopes.

Universal groups are more expensive in two ways. The first is related to the global catalog and the second is related to access tokens. Table 3.21 explains both.

The outcome of the rightmost column in Table 3.21 is that if you have only one domain, neither cost in the table is an issue, so you can use universal groups with confidence.

The reason to have universal group members in the global catalog is to provide an efficient means to effectively implement groups in a WAN environment with multiple sites. The global catalog takes care that the

**Table 3.21** The Extra Costs Related to Universal Groups

<b>Cost</b>	<b>Explanation</b>	<b>Is an Issue</b>
Global catalog	All membership information of universal groups is replicated to the global catalog. This means that every time the members change, this information has to be replicated to all sites of the enterprise throughout the world (provided that they all have a global catalog server). To minimize changes, have only groups as members of universal groups. This way, changes in membership don't occur as often as when users are members.	If you have both multiple domains and multiple sites
Access tokens	Global and domain local groups come only from the applicable domain into the user's access tokens. Universal groups, however, come to the user's access tokens from all domains of the enterprise forest. Thus, using universal groups leads to larger access tokens (consuming some memory) and to slower logon times.	If you have multiple domains and a fairly large number of groups

## 230 Chapter 3 Managing OUs, Users, and Groups

---

membership information is present on all sites (provided each site has a global catalog server). Therefore, checking a user's membership (needed to determine his access to resources) doesn't require crossing WAN links to other sites.

---

**NOTE** If you test universal groups, you may run into the following "problem" (only in AD2000): You create a test universal group on a domain controller that is not a global catalog server. Then you test the new group and find out that it doesn't work (yet). The reason is that because the universal group membership is read from a global catalog server, it doesn't work until the group and the membership information have been replicated to the global catalog server, where your domain controller reads this information. You may even be sitting at a domain controller that contains this information, but still it must be read from elsewhere.

---

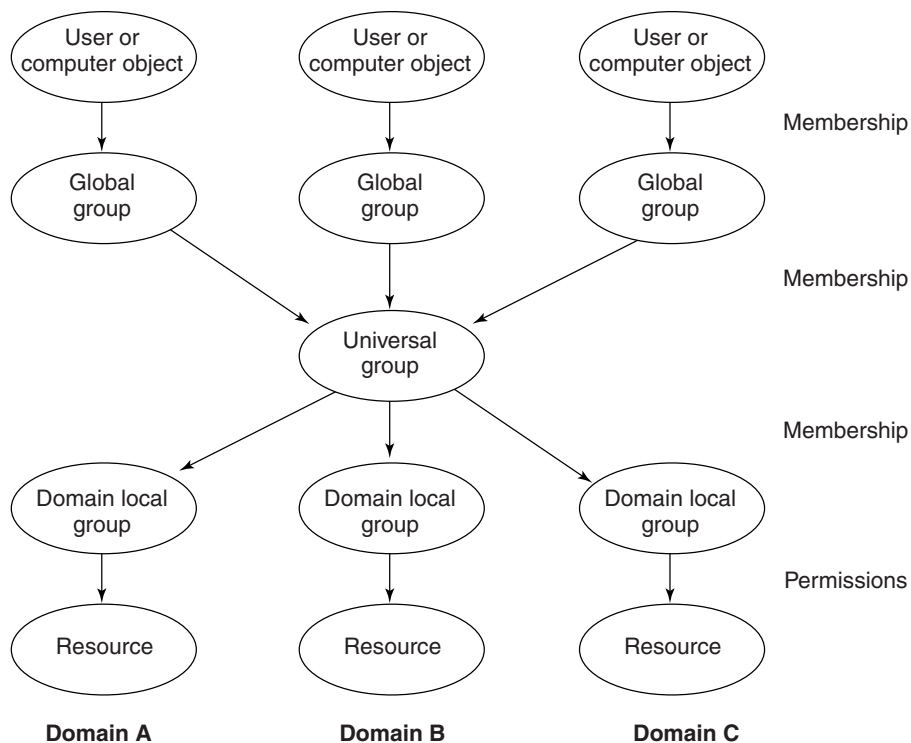
To summarize, we can make two claims that may sound contradictory at first:

- Universal groups are suitable for small networks.
- Universal groups are suitable for large networks.

The rationale behind the two claims comes from the three benefits of universal groups.

- *For small networks:* Cost is not an issue, and universal groups are easy to learn because there is only one scope with free nesting.
- *For large networks:* Universal groups provide an effective way to create groups with members from multiple sites.
- *For large networks:* Only universal groups have the scope to take members from different domains and to be assigned permissions for resources in different domains (see Figure 3.36).

The first benefit (for small networks) means that you would use only universal groups. The other two benefits (for large networks) mean that you would use universal groups occasionally in addition to global and domain local groups.



**Figure 3.36** If you need a group that can have members from different domains and that can be given permissions for resources in different domains, your only choice is a universal group.

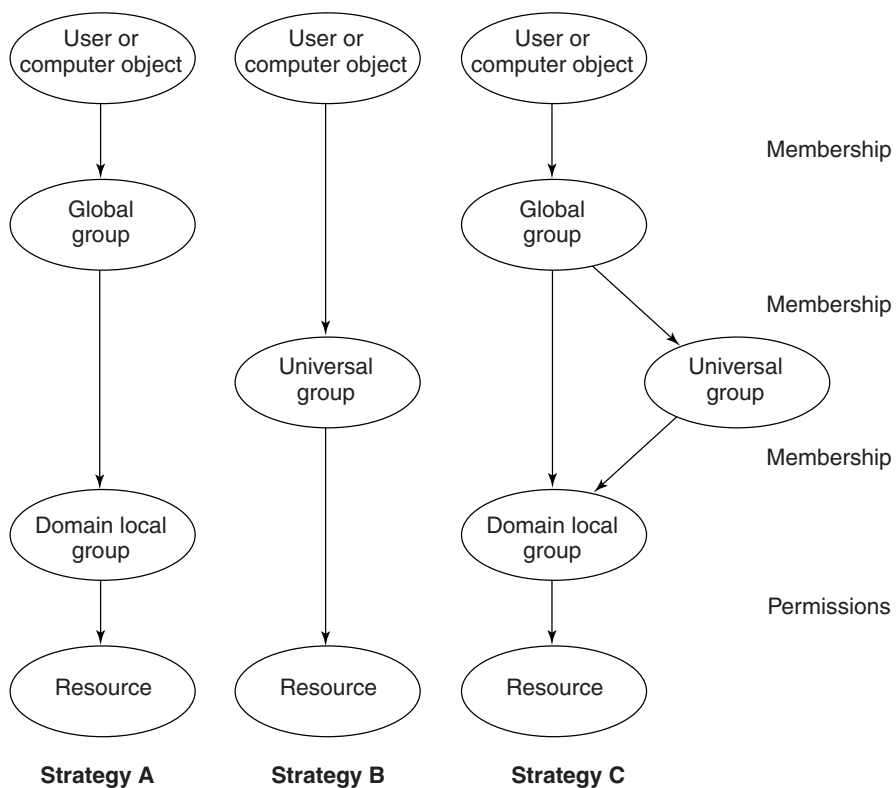
**NOTE** If you removed the universal group from Figure 3.36, you could achieve the same networking result. It would be very cumbersome to do so, however. You would need 9 (3 x 3) direct memberships from the global groups to the domain local groups. Or, with 17 domains, you would need 289 (17 x 17) direct memberships.

### Three Group Strategies

There are three basic approaches to organizing groups according to scope, as Figure 3.37 illustrates.

- *Use only global and domain local groups (strategy A).* If you feel comfortable with the two levels of groups that global and domain

## 232 Chapter 3 Managing OUs, Users, and Groups



**Figure 3.37** Depending on your network's size and needs, you can choose one of three group scope use strategies.

local groups provide (most likely from earlier Windows NT experience), you could use this as your group strategy. You have either just one domain or maybe a few of them.

- *Use only universal groups (strategy B).* If you have only one domain and you don't want to learn and think about different group scopes or levels, you will do fine with using just universal groups. You can use one level of groups between users and resources, without group nesting. Or you can put some groups in other groups to have a little nesting. Whether or not you develop logical levels for your groups is your choice. Of course, there are always some predefined global and local groups.
- *Use all three scopes (strategy C).* If you have multiple domains (and perhaps sites), you probably need all three group scopes. You'll mostly use global and domain local groups because they

don't have the extra "cost." In this strategy, you use universal groups only when you need a group with members from different domains (perhaps in different sites) and when you want to assign permissions for resources in different domains.

We have a few final comments about group usage before we move on to the next section.

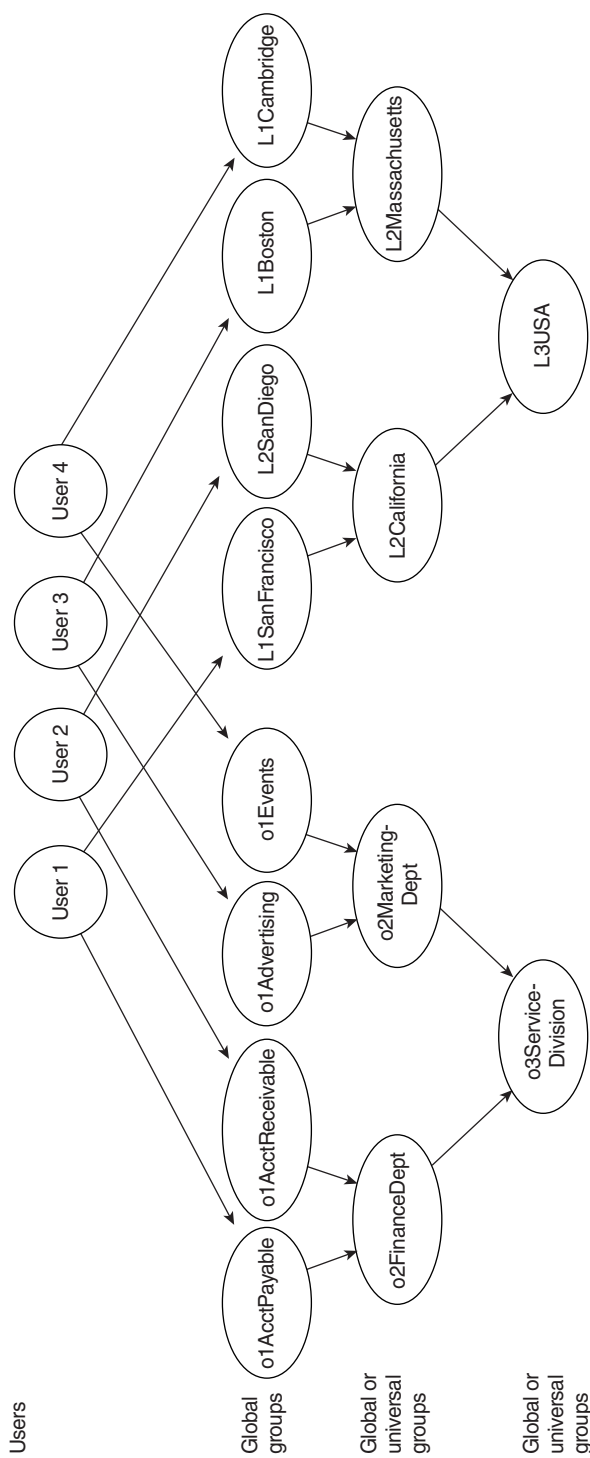
- Don't get carried away with group nesting. If you have more than three levels, you might lose track of your group hierarchy. Too much nesting could easily confuse, rather than simplify, network administration.
- You might want to create a group containing only one person. Active Directory doesn't have a role object class as Novell NDS has, but you could use groups in this sense. Usually one person at a time holds a role, so the group has only one member. For example, if some user is taking care of backups this month, you could put her in a group and give that group the appropriate permissions. When someone else takes over the role, you change the group membership by removing the first user and adding the new user.

### ***Efficient Group Nesting***

Group nesting in pure Active Directory functional levels is relatively free. The only restrictions are—as explained earlier in this chapter—that (a) you cannot put "lower" groups as members in "upper" groups, such as putting a universal group in a global group, and (b) based on group scopes, only some memberships are allowed across domain boundaries. Consequently, you can do almost whatever you want with group nesting.

We cannot present all the various scenarios here, but we present one efficient and systematic use of nesting. It uses the following steps (illustrated in Figure 3.38).

1. We put each user in one global group that is based on the smallest (necessary) organizational unit (not to be confused with the OUs of Active Directory). We denote these groups with "o1".
2. We put each user in one global group that is based on the smallest (necessary) location unit. We denote these groups with "L1" (where uppercase is more distinguishable than lowercase).



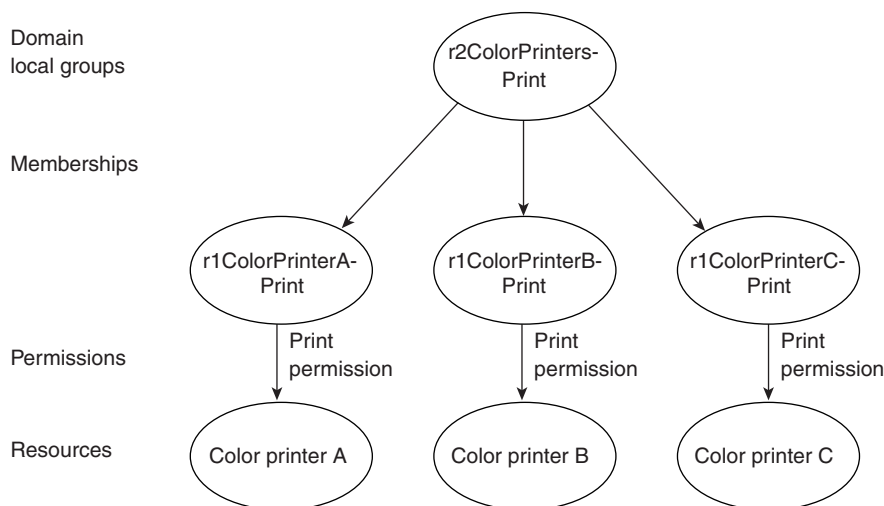
**Figure 3.38** You can develop a systematic group nesting approach by assigning each user to one small organizational group and to one small geographical group. Then you assign these groups to larger groups, and those to even larger groups.

3. We nest the smallest organization groups into larger ones, denoted with “o2”, and similarly the smallest location groups into larger ones, denoted with “L2”.
4. As the fourth step, we nest the o2 groups into o3 groups, and L2 groups into L3 groups.

This approach deserves the following comments:

- You need to put each user in only a minimal number of groups (two in this case).
- In addition to this systematic model, you may need to put users in some other groups, such as in the functional groups illustrated in Figure 3.25.
- If you implement the scenario in a single domain, you can do with just global groups.
- If you have several domains, either your organizational or geographical group division does not match your domain division. Consequently, some of your group memberships must cross domain boundaries, and consequently, some of the o2, o3, L2, or L3 groups must be universal instead of global. It is most likely that all the members of each o1 and L1 group are in a single domain, so all o1 and L1 groups can be local.
- In the unlikely event that, for example, the members of o1AcctPayable are in two domains, you must make that group universal. You must also make intermediate global groups, one in each domain, because due to excessive replication, users should not be directly in universal groups. You could call these intermediate global groups o0AcctPayableDomA and o0AcctPayableDomB. Note that these intermediate groups are not shown in Figure 3.38.
- It is quite possible that either the organizational or geographical group structure more or less matches your OU structure. The group structure, however, has nothing to do per se with the OU structure. Also, the groups' locations in the OU tree don't affect how they work.
- If your OU structure matches the geographical structure, for example, you can place each geographical group in the corresponding OU. This way the person who was delegated the administration of the Boston OU, for example, can manage both the Boston users and the L1Boston group. And the person who is responsible for the Massachusetts OU can put L1Boston and L1Cambridge into L2Massachusetts.

## 236 Chapter 3 Managing OUs, Users, and Groups



**Figure 3.39** To simplify permission management, you can create a domain local group (or a local group in a member server) for each color printer and then assign a larger color printer group to each per-printer group.

- If you have no use for one of the largest groups in Figure 3.38, such as L3USA, you can obviously choose not to create it at all.

If you feel you have the need, you can also nest the resource-oriented (domain local) groups. For example, you can create a group for each color printer and then a larger group to cover several color printers. See Figure 3.39 for an illustration.

### Tips on Tools

We use the Users and Computers snap-in often, as a main tool, and there are some helpful tips that we haven't yet covered. In addition, the snap-in is not the only tool available to manage Active Directory objects. Before we conclude this chapter, we'll say a few words about the Users and Computers snap-in, as well as about other means to manage objects.

#### The Users and Computers Snap-In

We have been using the Users and Computers snap-in throughout the chapter. Here we'll briefly fill in some last few holes.

### **Choosing a Domain**

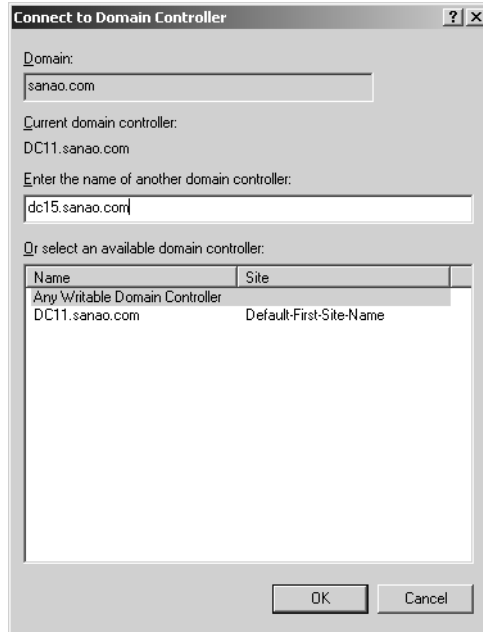
You can connect to another domain by right-clicking the uppermost line of the left pane (Active Directory Users and Computers . . .), selecting Connect to Domain, and then specifying a new domain either by typing its name or selecting it from a list.

### **Choosing a Domain Controller**

Sometimes you want to communicate with a certain domain controller. You can choose one by right-clicking the uppermost line of the left pane (Active Directory Users and Computers . . .), selecting Connect to Domain Controller, and then selecting a new domain controller from a list or typing in a new domain controller's name in the dialog box shown in Figure 3.40.

### **Finding Objects and Information**

The context menu of the domain object and each OU has a Find item. You can use it to find objects that match certain criteria.



**Figure 3.40** You can specify a domain controller to communicate with in the Connect to Domain Controller dialog box.

## 238 Chapter 3 Managing OUs, Users, and Groups

Windows Server 2003 includes a new Find feature called Common Queries. It enables you to find things such as all disabled user accounts or user accounts that haven't logged on for two months.

See also the "Search Tools" section in Chapter 6.

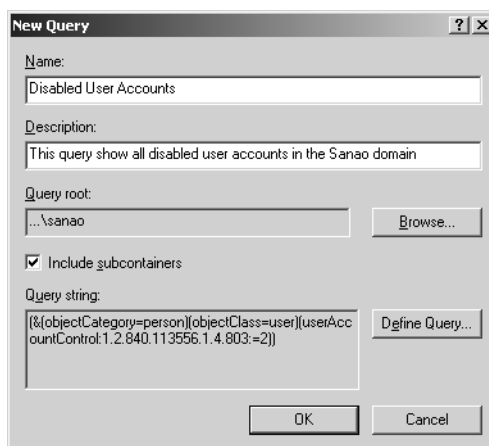
### Filter Options

The View menu of the snap-in includes Filter Options. This feature enables you to specify the objects you want to see when you browse various container objects. (For more information about finding and filtering objects, see Chapter 6.)

### Saved Queries

The Windows Server 2003 version of the Users and Computers snap-in includes a new feature called saved queries. You see the Saved Queries folder in the left-hand pane of the snap-in and—as the name implies—you can store there queries for later use. For example, you can define a query that displays all users of the domain that don't have a home folder property set.

As you can see in Figure 3.41, when you create a new query, you can specify a name for it, the starting point ("Query root"), whether to include subcontainers, and finally, the actual query. The query string is shown as an LDAP query string, but when you define it (by clicking Define Query),



**Figure 3.41** You can create and save often-used queries in the Saved Queries folder of the Users and Computers snap-in.

you have a myriad of user-friendly alternatives to specify the query that suits your needs. Or you can also directly type an LDAP query string, if you first learn how to form them in Chapter 6.

### **Viewing Advanced Features**

The View menu of the snap-in includes Advanced Features. If you turn on those features, the user interface will make the following adaptations:

- Each object will show additional tabs in the property pages. We discuss the Security tab in Chapter 4 and the information in the Object tab in Chapter 5.
- You will see additional containers and objects. The System container includes miscellaneous domain-specific objects, such as the Group Policy containers. The LostAndFound container includes objects that lost their parent container due to a replication conflict. This is explained in Chapter 5. The Program Data container includes things such as Authorization Manager data that defines query-based groups. Finally, the NTDS Quotas container includes quota specifications if certain security principals have a certain maximum of how many directory objects they may own (see Chapter 4.)

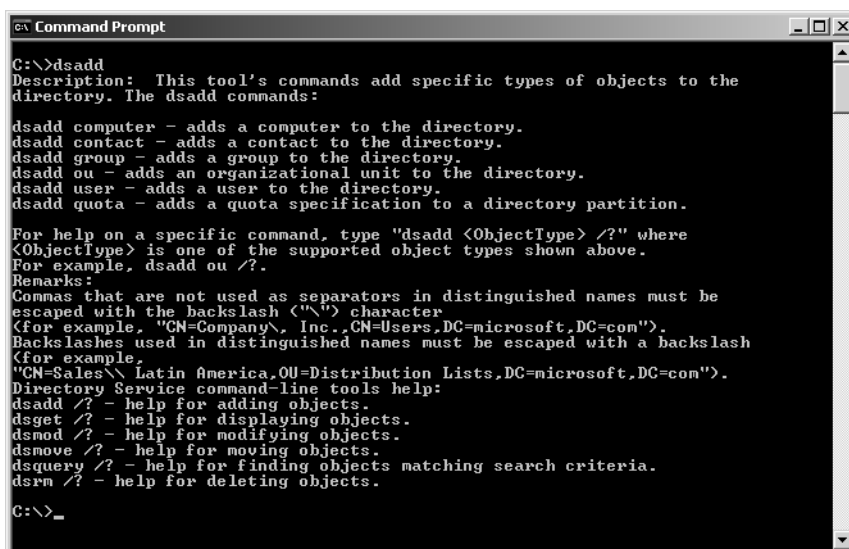
### **DSAdd and Other Command-Line Tools**

Windows Server 2003 includes a number of handy new command-line tools to search, display, add, modify, move, and delete objects. You can add computers, contacts, groups, OUs, users, and quota specifications. For other operations, the object type selection is wider, and you can search, move, or delete any object types. Figure 3.42 shows the help for the DSAdd command. In addition to the help provided from the command line, the Help and Support Center includes the help for the commands.

Figure 3.43 shows an example of how to create a user account with the DSAdd command. Note that the switches to specify first name, last name, and so on are not LDAP names or display names of the properties.

You can use the output of one command as input for another command. This is called *piping*. For example, you can list all users in a certain OU and add them as members in a certain group, as shown in Figure 3.44.

## 240 Chapter 3 Managing OUs, Users, and Groups



```

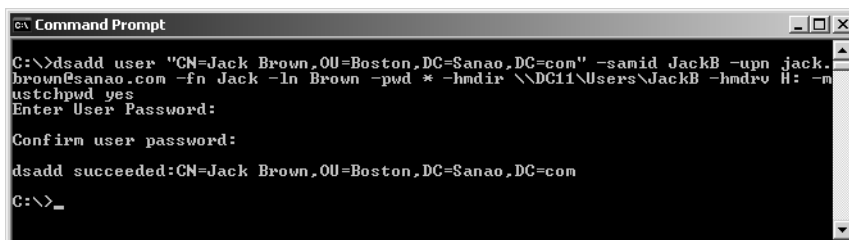
C:\>dsadd
Description: This tool's commands add specific types of objects to the
directory. The dsadd commands:

dsadd computer - adds a computer to the directory.
dsadd contact - adds a contact to the directory.
dsadd group - adds a group to the directory.
dsadd ou - adds an organizational unit to the directory.
dsadd user - adds a user to the directory.
dsadd quota - adds a quota specification to a directory partition.

For help on a specific command, type "dsadd <ObjectType> /?" where
<ObjectType> is one of the supported object types shown above.
For example, dsadd ou /?.
Remarks:
Commas that are not used as separators in distinguished names must be
escaped with the backslash ("\") character
(For example, "CN=Company\, Inc.,CN=Users,DC=microsoft,DC=com").
Backslashes used in distinguished names must be escaped with a backslash
(For example,
"CN=Sales\ Latin America,OU=Distribution Lists,DC=microsoft,DC=com").
Directory Service command-line tools help:
dsadd /? - help for adding objects.
dsget /? - help for displaying objects.
dsmod /? - help for modifying objects.
dsmove /? - help for moving objects.
dsquery /? - help for finding objects matching search criteria.
dsrm /? - help for deleting objects.

C:\>_
  
```

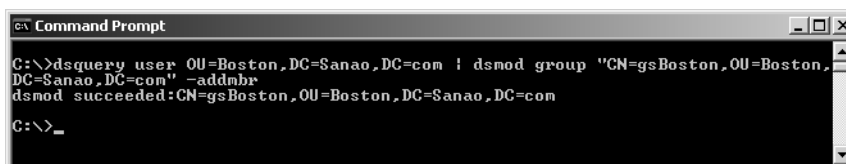
**Figure 3.42** The DSAdd command enables you to add objects of six different types. There are also other commands to search, display, modify, move, and delete objects.



```

C:\>dsadd user "CN=Jack Brown,OU=Boston,DC=Sanao,DC=com" -samid JackB -upn jack.
brown@sanao.com -fn Jack -ln Brown -pwd * -hmdir \\DC11\Users\JackB -hmdrv H: -n
ustchpwd yes
Enter User Password:
Confirm user password:
dsadd succeeded:CN=Jack Brown,OU=Boston,DC=Sanao,DC=com
C:\>_
  
```

**Figure 3.43** With the DSAdd command, you can create users from the command line and set many properties for them.



```

C:\>dsquery user OU=Boston,DC=Sanao,DC=com | dsmod group "CN=gsBoston,OU=Boston,
DC=Sanao,DC=com" -addmbr
dsmod succeeded:CN=gsBoston,OU=Boston,DC=Sanao,DC=com
C:\>_
  
```

**Figure 3.44** You can use one command to list all users of an OU (DSQuery) and then use this list as input for another command (DSMod).

## Alternative Means to Manage Users and Other Objects

In addition to the Users and Computers snap-in, you have the following means available to you to manage users and other objects:

- *ADSI Edit*: This tool is part of Windows Support Tools. While the Users and Computers snap-in shows only some objects and some of their properties, ADSI Edit shows everything. It is not practical for everyday administration, but occasionally you might need it. We use ADSI Edit in quite a few places in later chapters.
- *LDIFDE and CSVDE*: These two tools are part of the operating system. They enable you to import and export objects between Active Directory and a text file. We explain how to use them in Chapter 6.
- *Net commands*: The operating system includes about 20 Net commands that were inherited from Windows NT, which inherited them from LAN Manager. You can create batch files with them to automate administration, but they don't understand the directory structure of Active Directory. You can get a list of these commands by typing "NET HELP" (without quotes), and you can get help with an individual command by typing "NET HELP *command*."
- *WSH scripts*: You can download scripts from the Internet or write scripts that will do "anything," including managing Active Directory objects. Chapter 10 and Chapter 11 provide further information.

## Conclusion

At this point you should have a pretty good understanding of users, computers, and groups in Active Directory and how to manage them. Later chapters address designing Active Directory and give practical examples of how to use the objects discussed in this chapter.

This chapter focused mainly on one tool: the Users and Computers snap-in. Later chapters introduce some Windows Support Tools and Resource Kit tools, and explain how to use scripting in user and group management.

This chapter assumed that you have full control over all objects in Active Directory. The next chapter explains how to control access and administrative rights to Active Directory by assigning permissions and user rights.

