FIGURE 5.12

The maximum number of subordinate containers is 50 at each level in the tree. Beyond this number you should consider adding another level in your tree to distribute your containers.

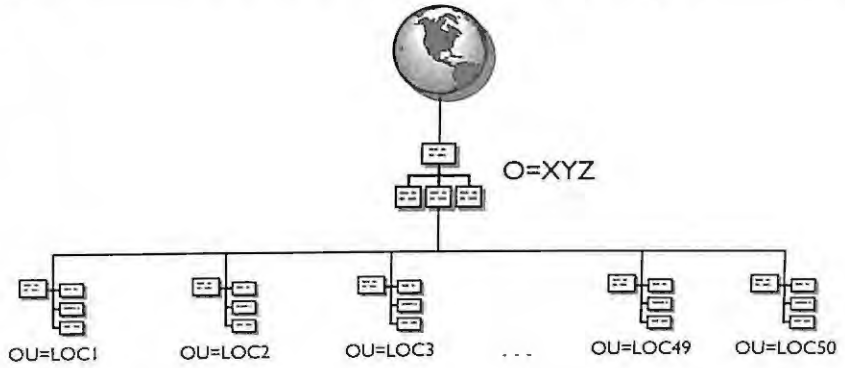
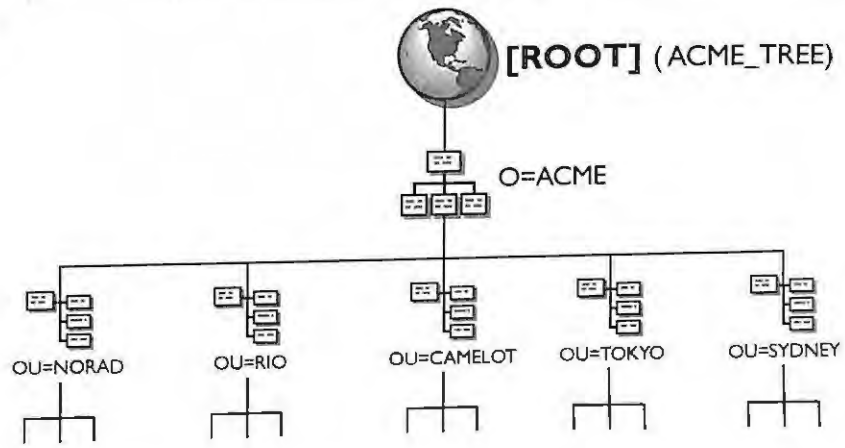


FIGURE 5.13

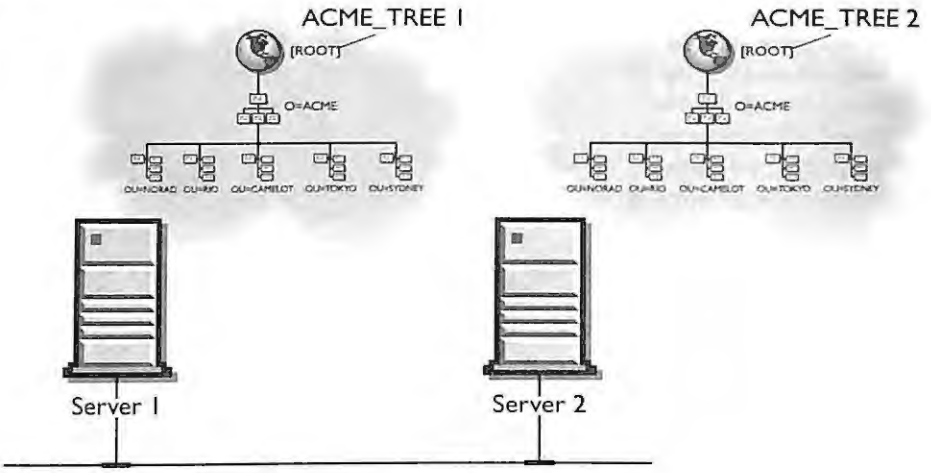
Another name for the [ROOT] object is ACME_TREE.



The name of the tree should be a unique value on the network wire because the tree uses SAP to broadcast to the client or workstations where the tree can be found. SAP bootstraps clients and all applications requiring NDS to find the NDS database very efficiently. If you need to install more than one physical NDS tree make sure that the trees have different names. An example of this important point is illustrated in Figure 5.14.

FIGURE 5.14

You may have multiple trees on your network. Each NDS tree must have a unique name such as ACME_TREE1 on Server1 and ACME_TREE2 on Server2.



The company name plus _TREE is recommended because it clearly identifies the tree SAP as an NDS tree when you display servers at a console. Be careful not to make this name too long. Also, the SAP does not support spaces in the name, which is advertised via SAP, and the NetWare 4.1 INSTALL utility will not let you place spaces in the tree name.

The NDS tree always starts with the [ROOT] container object. In most discussions, however, the [ROOT] object is not counted as a layer in the tree.

The object class definition in the schema that defines the [ROOT] is the object class TOP. The [ROOT] object is the only instance of the object class TOP and for this reason TOP is known as an effective class.

The [ROOT] object is parent to either the C=Country and/or O=Organization. Novell Consulting Services recommends that you use the O=Organization below [ROOT] rather than the C=Country object.

To Use or Not to Use the C=Country Object

The C=Country designator is used to specify a particular country code based on the X.500 standard. Public network providers, such as NetWare Connect Services (NCS) being offered with the cooperation of Novell, will make use of the Country object in their tree. The question often asked is, "If a company wants to connect to a public service provider, is it required to use the C=Country code in the NDS tree?" Most companies are not required to use the country object for their corporate tree. Instead, they can create a separate tree used for connecting to the public data network or through a client that can connect to multiple trees.

If you choose to use the Country object, keep in mind that it will add an additional layer to your NDS tree and it will also create some rather odd distinguished names for your objects. Consider the example below in Figure 5.15. If we were to add the Country object to the ACME tree, which country do we choose? Do we use multiple country codes? For our example, ACME is headquartered primarily in CAMELOT; therefore, our Country object will be C=UK for England in this example.

Let's look at some of the user's contexts that would be created in other locations if we used only the C-UK COUNTRY designator. Abe Lincoln resides in the RIO location and so his context would be:

```
CN=ALINCOLN.OU=AUDIT.OU=ADMIN.OU=RIO.O=ACME.C=UK
```

User Sherlock Holmes in the TOKYO location would have the following context:

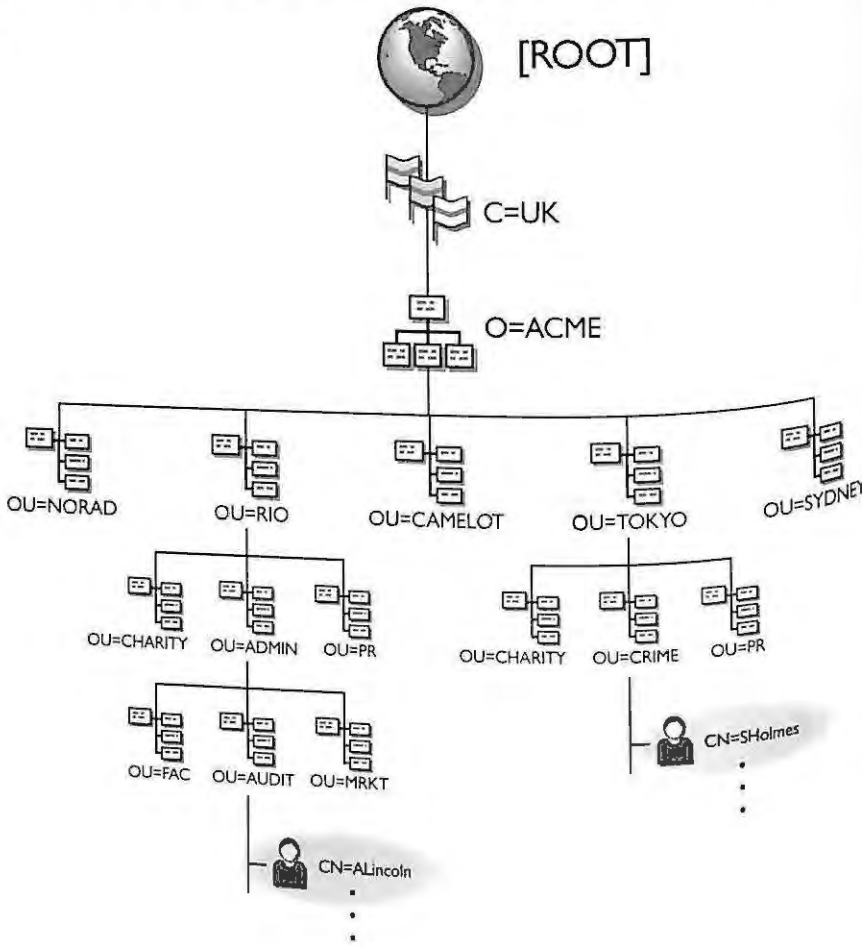
```
CN=SHOLMES.OU=CRIME.OU=TOKYO.O=ACME.C=UK
```

Now, if your names are supposed to adequately describe and identify a user's location in the tree, these examples are a little confusing and add more length to the context. Also, for some users who work in both the UK and the United States, it is difficult, if not impossible, to determine where in the tree they belong.

If you have already implemented the Country object in your tree, not to worry. It does not cause any serious consequences, but keep in mind the previous considerations.

FIGURE 5.15

The use of the Country object can create some odd contexts in your tree.



Name the O=Organization for Your Company

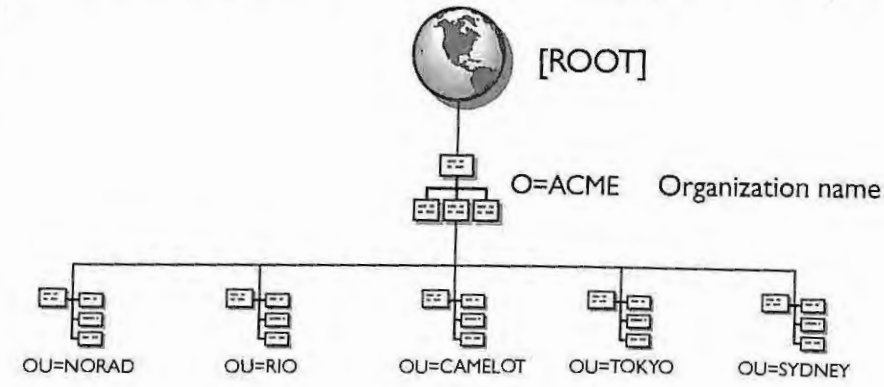
After the [ROOT] object at the top, you will provide the NDS tree with at least one O=Organization. At least one O=Organization object is required for all NDS trees. The subsequent layers in the tree (the OUs) will be placed directly below the O=Organization.

We recommend that you name the O=Organization the same name as your company or use an abbreviation. Most companies use an abbreviation for the company name because it is easier when you are typing an object's context. For example, our company is named A Cure for Mother Earth, which is abbreviated to ACME. In almost every case,

the Organization layer in the tree contains only one O=Organization, which gives you a single object to represent the entire company. Figure 5.16 shows how we have named our organization to represent our company name ACME.

FIGURE 5.16

ACME is representative of our entire company and is used as our organization name in our NDS tree.



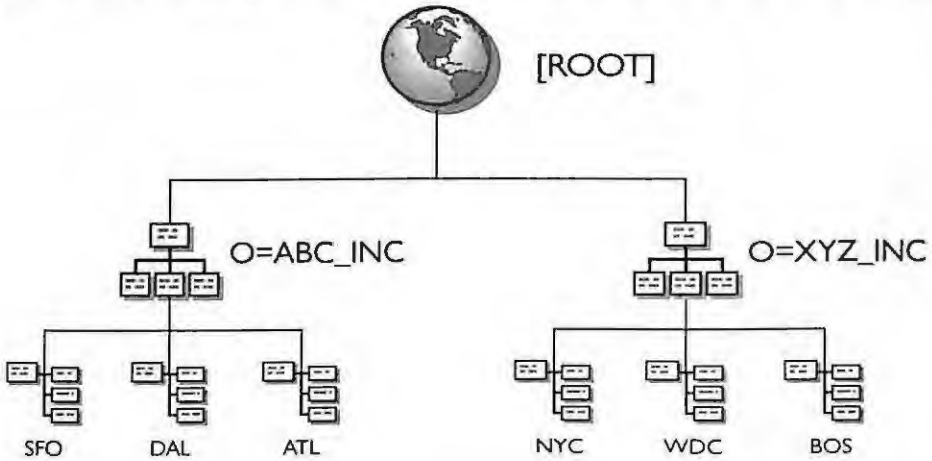
CONSULTING EXPERIENCE

We recommend that you not name the O=Organization the same name that you used for the NDS tree. For troubleshooting purposes, the NDS tree should be named with the company name plus _TREE, and the O=Organization should be named with just the company name or an abbreviated company name. The ACME corporation would therefore be named O=ACME, with a tree name ACME_TREE.

Your company may want to use more than one O=Organization if your corporation has multiple companies that do not share the same network infrastructure. For example, the large conglomerate shown in Figure 5.17 uses multiple O=Organization objects because there are two separate companies (separate network infrastructures) included in a single NDS tree.

FIGURE 5.17

A large conglomerate company with multiple O=Organization objects



CONSULTING EXPERIENCE

A single NDS tree with two or more O=Organization objects is rarely used and is not usually recommended. This configuration is not often used because one of the design goals is to represent the entire corporation in the single tree with the same organization name.

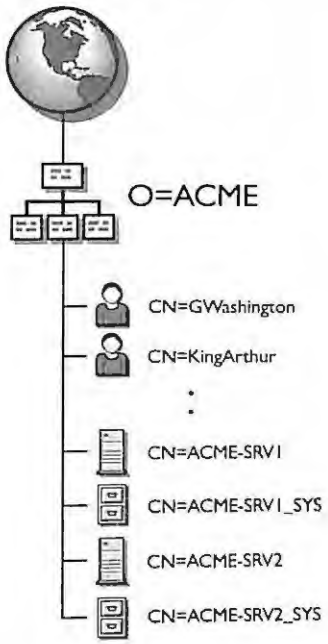
Small Companies

In some cases the tree design can be finished very easily for small companies at this point because most servers, users, and other resources can be placed in the Organization container without creating any more containers. If you are the network manager responsible for all users, printers, and servers, you can simply group everyone in the same container, which can be the Organization container.

Figure 5.18 shows how a tree design can be very simple for a small company. If ACME had only a few servers in a single location, its tree could appear as shown in the figure. You may still want to subdivide the tree a little if you have separate groups.

FIGURE 5.18

Small companies can group all their resources in the organization container if they have only a single network administrator managing all resources.



The Geographic Design: Top Layers of the Tree are the OU=Organizational Units

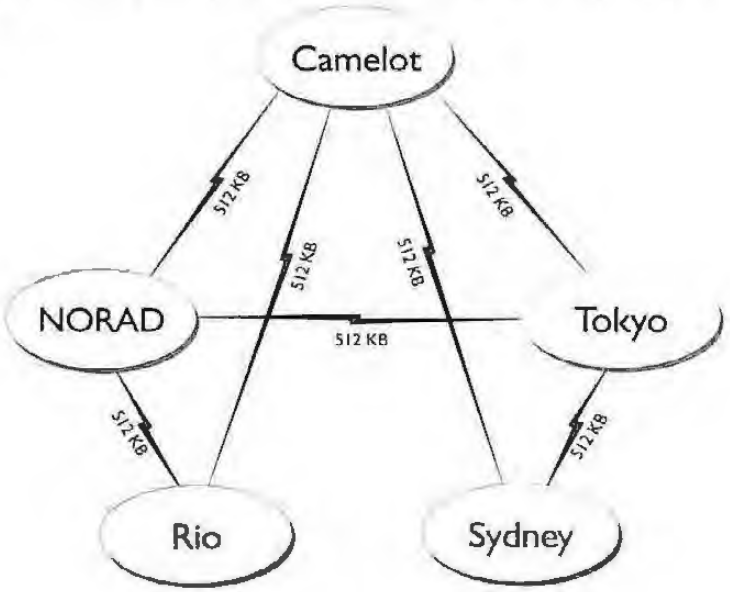
As mentioned previously, the layer below the O=Organization is the first layer of OU=Organizational Units in the NDS tree. This layer of OUs is the most important layer of the NDS tree because it represents the geographical locations of your company. Using your company WAN maps or WAN documentation you can carefully design the contents of this layer, which becomes the foundation for the entire tree. This method is often referred to as the geographical design approach because you use your company's geographic or location information for the design at the top of the tree.

The key to designing the top of the tree is to match the WAN infrastructure or locations of your company with the first OU layers or containers. Based on our experience at many sites, the design of the top of the tree should be completely based on the WAN infrastructure. You will have a successful NDS tree design if you follow the guideline of representing the sites of your company with the top-most OUs. Figure 5.19 and Figure 5.20 illustrate how the top layer of the ACME tree is designed based on the physical WAN layout of the company.

If you are a small company (no WAN, 5 servers or fewer), you can simply use the first Organization container you created to name your company. This container will hold all your objects including printers, servers, and users. Administration of a single container is very easy and requires very little maintenance.

FIGURE 5.19

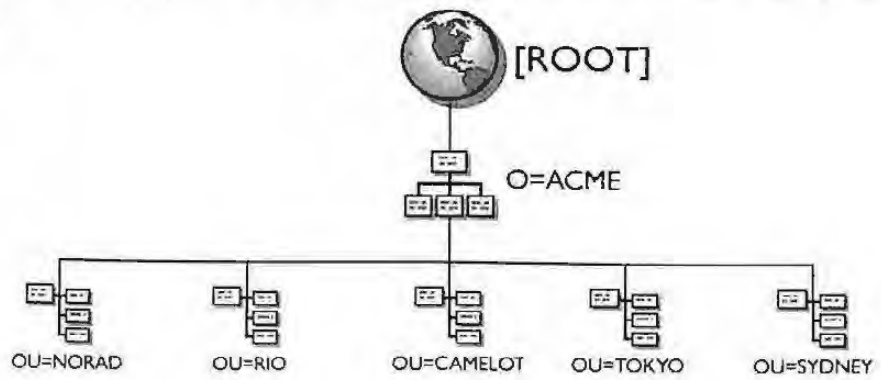
Physical WAN layout for ACME



Physical WAN Layout for ACME

FIGURE 5.20

Top layer of the tree design for ACME, which is based on the physical or geographical WAN sites



In general, if your company has multiple geographic sites or locations, you should represent the locations in the NDS tree at the top of your tree. The organizational structure of departments, divisions, and workgroups will be placed under each of these locations. Keep in mind that one of our design goals is to design a flexible tree in which changes are easily made. As you might expect, there are a few exceptions to the practice of designing geographically:

- Companies with a single site or campus-connected network are not dependent upon the geographic design approach. Since this configuration does not have physical locations that can be placed under or created as OUs, you will skip the geographical design approach at the top of the tree and proceed directly to the departmental design approach. Some companies with few servers and users may not need to create additional containers. Rather, they can place all the NDS objects under the single O=Organization.

- For companies with WAN sites or locations connected with very high speed links, such as T-3 or greater, the location OUs are less important because the limitation of the WAN has been removed. This is because WAN speeds are approaching LAN speeds. For the purpose of NDS tree design, the high-speed WAN connections really represent LAN bandwidths. However, we still recommend that you use the geographic design approach. See the section "Design the Bottom Level of the Tree" later in this chapter.

CONSULTING EXPERIENCE

Many companies still choose to use geographic containers even though they have very high speed WAN links. One company, for example, has a metropolitan area network (MAN) running FDDI to connect 12 buildings together across a city. The basis for the company's decision to use geographic sites at the top of the tree was twofold. First, for administrative purposes, the company wanted a single administrator to support each site. The sites gave the tree a good place to break out security administration. Second, the company was installing an e-mail application on its servers at each geographic location. So, even though a company has high-speed links, it may still choose to design geographically.

When considering a campus network layout, such as a research park or university, consider first the speed of the links between the buildings or floors of the campus network. The locations in the campus network, such as buildings, could be used to represent minor sites in the network infrastructure and in the NDS tree. The buildings in the campus network can be useful container objects if they help organize your network resources and the NDS tree. The ability to effectively organize network resources is one of your design goals. The ACME tree NORAD location as shown in Figure 5.21 has used buildings named by function as its organizational units. Either design approach is acceptable. You must determine which one provides the best description of your environment.

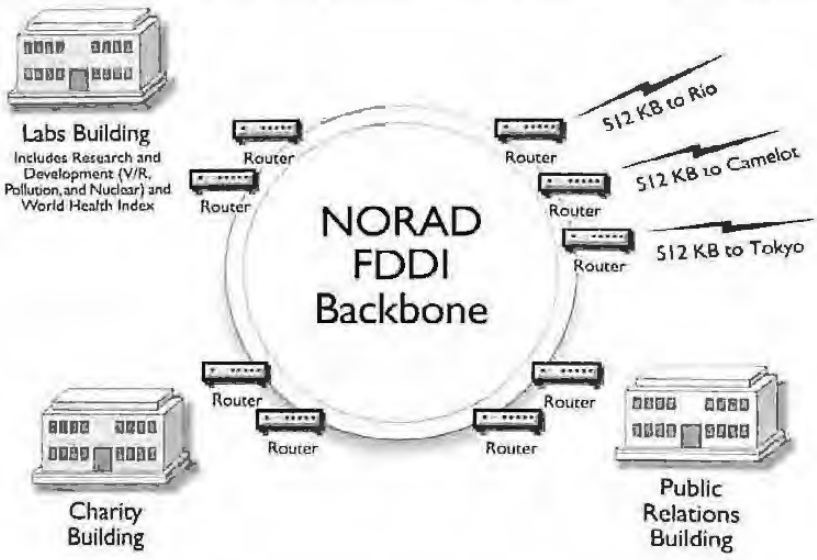
If your company does not have a WAN infrastructure but only a LAN network, then you can skip the geographical design approach and go directly to the departmental design discussed later in this section.

Regional Layer of Organizational Units Helps Distribute the Location OUs

In some cases, it will be necessary to place regional containers directly below the O=Organization in the NDS tree to more fully distribute the total number of locations or geographical sites. Placing regional OUs under the O=Organization, but before the actual location OUs, will increase NDS operating efficiency and give the tree a closer pyramid shape.

FIGURE 5.21

The ACME NORAD organization units are named based on their functions, which are also the building names such as OU=CHARITY, OU=LABS, and OU=PR.



As an example, consider the company ACME as we change the WAN layout to include more offices or cities around the world. We are changing ACME's WAN infrastructure only for this example. Figure 5.22 illustrates the offices or cities that are connected together via 56K links. Each of the cities added to the WAN layout is connected to its appropriate regional hub. Using the WAN infrastructure, we have designed a new tree, which includes regional OUs named North America (NA), South America (SA), Europe (EUR), Asia (ASIA), and Australia (AUST). These regional OUs group the appropriate cities and help keep the NDS tree design closer to a pyramid shape. See Figure 5.23 for the new ACME tree based on regional containers. Notice how the physical WAN layout in Figure 5.22 is driving the tree design in Figure 5.23.

If your network utilizes a WAN infrastructure with a number of physical sites or offices, you may want to create regional containers based on those WAN sites at the top layer, which will help distribute the individual offices. Having the regional OUs helps the NDS tree operate more efficiently during all phases of operation.

FIGURE 5.22

Example of ACME with regions and cities. This is typically called a "hub and spoke" WAN infrastructure.

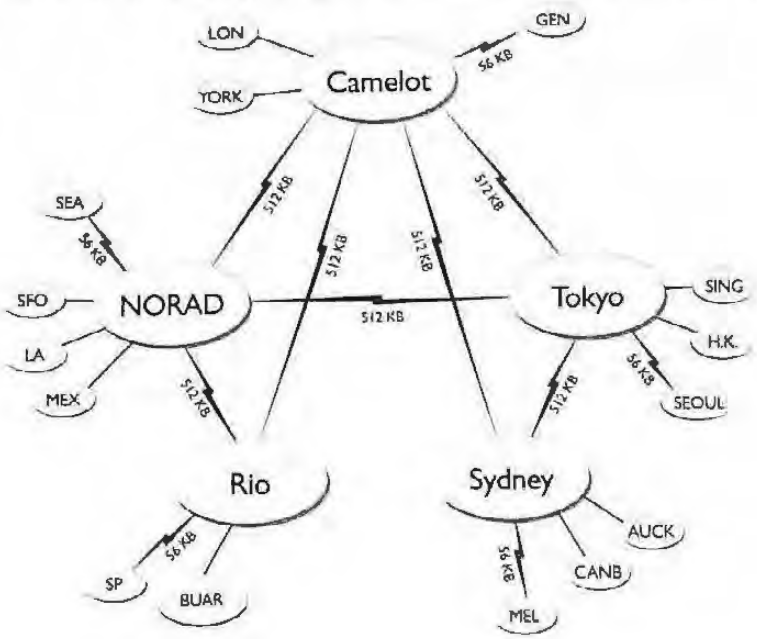
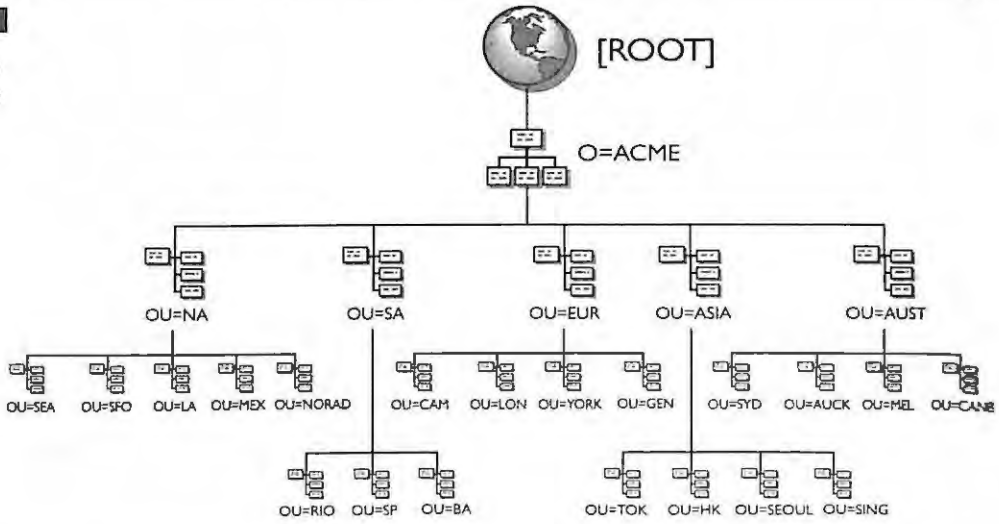


FIGURE 5.23

Example of ACME tree with the regions and cities as the top layers



Departmental Design: Top Layers Not Based on Your WAN

The departmental design approach can be used most efficiently at the top of the tree only if your company does not have a WAN infrastructure or other locations to consider. If your company has only a LAN-based network, then you can skip the design of the top layers and go directly to the bottom layer design, which is based solely on the organization of the company.

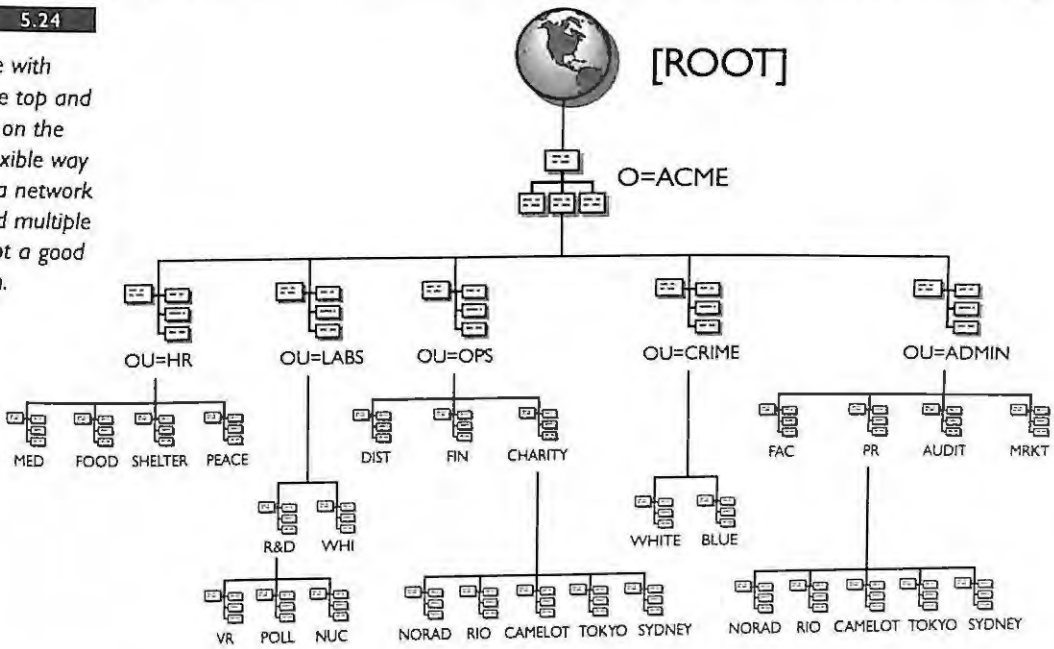
If you have WAN links you may consider designing your NDS tree by placing the departments, divisions, and workgroups at the top of the tree and placing the physical locations at the bottom. This method is often called the departmental design approach and is not recommended for a company with a WAN infrastructure. Having the organizations placed at the top of the tree is a less efficient tree design because any change to the top organizations will ripple down the entire structure, including the sites locations below.

Consider the example in Figure 5.24 in which we have designed the top of the tree organizationally with locations at the bottom. The first question you need to ask is where do most network changes occur? Most changes will occur in your organization. That's not to say that changes don't occur in geographic sites as well, but they are less

frequent. Therefore, when you make changes to the tree you want to impact as few people as possible. This is the third design goal of building flexibility into the tree design. In terms of other design elements, such as administration, partitions, replicas, network resource placement, login scripts, and bindery services, it is apparent that the organization layers at the bottom of the NDS tree more adequately address these features.

FIGURE 5.24

The ACME tree with organizations at the top and geographic sites on the bottom is a less flexible way to design a tree in a network with WAN links and multiple locations. This is not a good tree design.

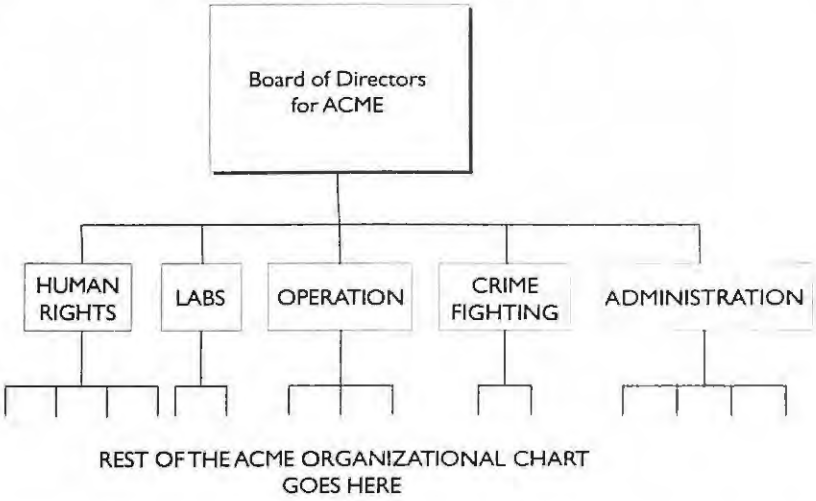


DESIGN THE BOTTOM LEVEL OF THE TREE

You should design the bottom level of the NDS tree along the organizational lines of your company by using your company's organizational charts or similar documents. The bottom layers of the tree are made up of OU containers, which are based on the divisions, departments, workgroups, and teams under each of the various locations defined at the top of the tree. Figure 5.25 shows the ACME organization chart that we will use in our tree.

FIGURE 5.25

The ACME organization chart used in our NDS tree



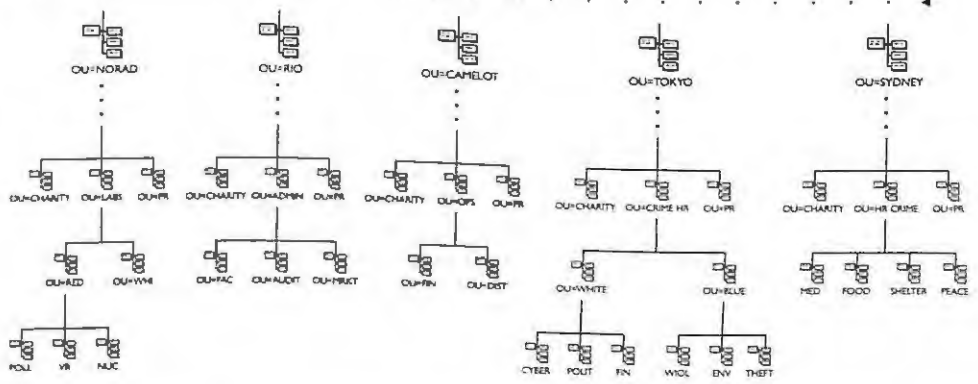
The bottom layers of the tree should represent the network resources located in the LAN network of the location or site. Since the LAN supports a greater bandwidth or throughput of information than the WAN, the design of the bottom layers is extremely flexible. You, as the designer and administrator, can shape the bottom of the tree to meet your specific needs.

We recommend that you design the bottom of the tree based on the organizational chart documents because the users and administrators are already familiar with that type of layout. Remember that the bottom section is flexible if it is designed around organizations. You will discover through experience that a tree designed with the organizations at the bottom of the tree can more easily adapt to the changing requirements of the corporation. Figure 5.26 shows the bottom layers of the ACME tree based on the organizational charts for each ACME site.

During the design of the bottom of the NDS tree, ensure that there is a place for every user and network resource currently in your company. Remember that the primary goal in designing the NDS tree is to organize the network resources, including the users. If you do not have a place for all the users or network resources then you need to adjust your tree design. The bottom layers are typically the only ones affected. Refer back to Table 5.1 for the ACME resource list. This list has information on servers and printers and provides you with helpful information for placing resources in your tree.

FIGURE 5.26

The ACME tree with the bottom layers of OUs based on the organizational charts of the company



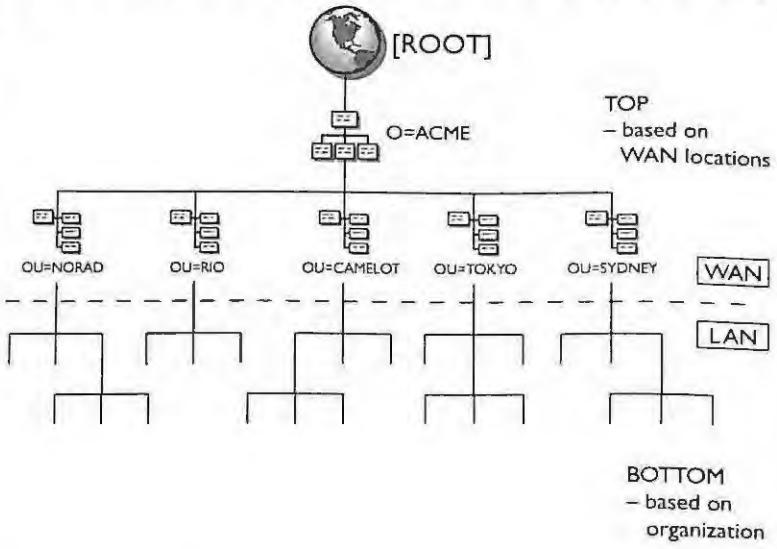
As mentioned earlier, the bottom containers or OUs in the tree are typically the divisions, departments, workgroups, and teams of your company. Do not include as containers any individuals that appear as division or department heads in your company's organizational charts. You simply want to identify the functional groups or departments; the individuals become the users in each container.

The ACME Tree Design

Notice in Figure 5.27 that the top layers of the ACME tree are based solely on the WAN infrastructure and will remain fairly stable or constant. Once the WAN infrastructure for ACME is considered in the design, the design effort shifts to the bottom of the tree. The bottom of the tree is based on the organizational chart for ACME. Most of the network resources will be placed in the bottom of the tree. Figure 5.27 illustrates a clear division between the top and bottom of the tree design phases in which the top is based on locations in the WAN and the bottom is based on the company's organizational information after crossing into the LAN infrastructure.

FIGURE 5.27

The ACME tree has been designed and consists of two phases: the top and bottom of the tree.



Placement of Network Resources

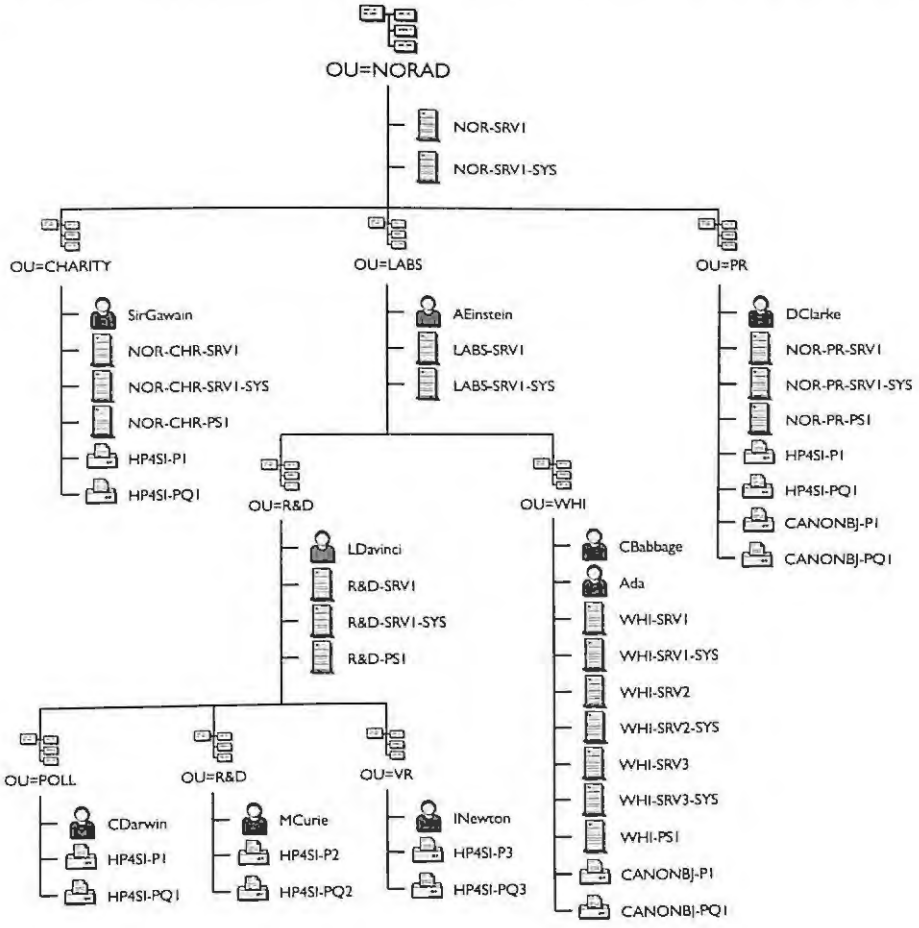
The placement of the network resources, such as servers or printers, in the tree could affect how you design the bottom layers. As you decide where to place the physical network resources in the tree, you should consider the needs of the users who will share these resources. If the network resources are organized according to divisions, departments, and workgroups, they should be placed in the same container with the users. However, if the network resources offer services to multiple departments in one site or location, you should place the resources in the location OU.

The placement of the network resources is an important design consideration for the bottom of the tree because the appropriate OUs or containers need to exist to place resources. If the OUs or containers do not exist then they will need to be created. Remember that one of the primary goals for designing the NDS tree is to organize your network resources.

With your resource list in hand, you can place your resources in their appropriate locations in the NDS tree. Below, we display illustrations of the ACME tree's five main sites. Included with each of these illustrations are some examples of how objects can be used in the ACME tree for the greatest impact and efficiency. Figure 5.28 shows the NORAD subtree with its resources.

FIGURE 5.28

ACME NORAD Site



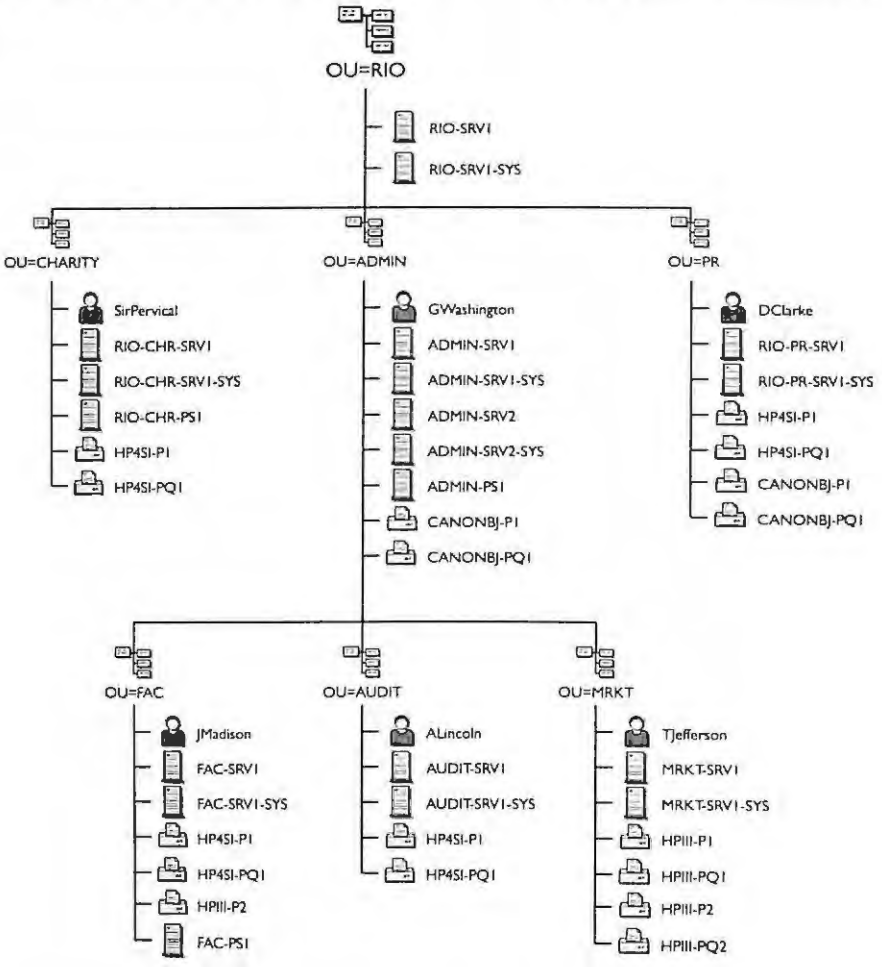
At the NORAD site, as well as all other sites, we have placed a central server at the top OU=NORAD. This server will hold the master replica of the NORAD partition and can also function as an e-mail server for this location. The same process is repeated at all five sites.

Notice that the naming standards follow a very simple pattern based on our naming standards document. Servers are always defined by unique names across the entire tree because of the SAP requirement. Printers and print queues, however, can have the same name as long as they reside in different containers, such as HP4SI-P1, found in both OU=CHARITY and OU=POLL containers.

In Figure 5.29, the RIO location shows the placement of resources in each of the departments. It is not necessary to place all users in your tree using drawings such as these. We have included a user in each location as an example. The primary purpose in placing objects in this fashion is to determine their general placement in the tree. This will give you a better understanding of organizations and their resources.

FIGURE 5.29

ACME RIO Site



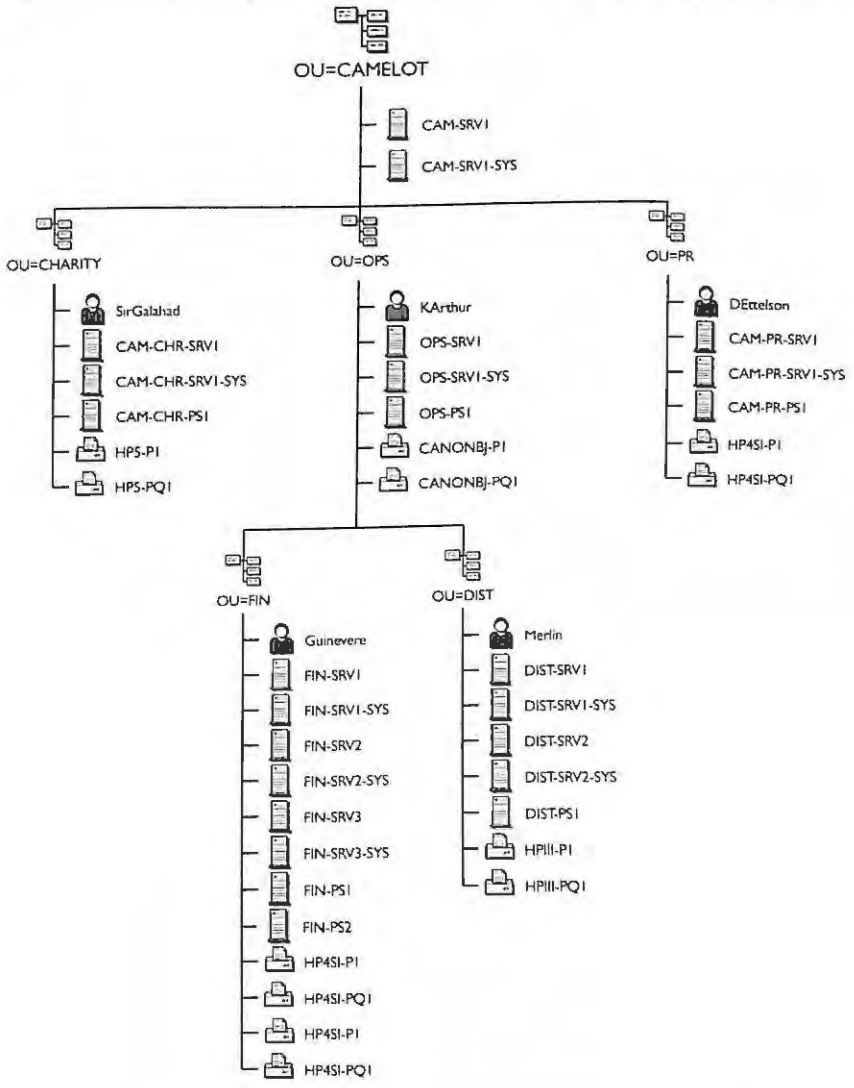
In addition to creating user and server objects, you will want to create some other objects as well. For the RIO location, as well as all major locations, you should consider creating an organizational role object as the site administrator. Grant supervisor rights at the site location, such as RIO, to the organizational role object. For example, create a

role called ADMIN_RIO. You can then move a user or two in as occupants of the role. If you have multiple administrators managing organizations at the same site, you may want to create separate roles for each department.

Since CAMELOT is basically the center of activity for the ACME tree, you may want to maintain control over the ADMIN user object from this location. Change the password frequently and limit the number of users who know the password. An example of the CAMELOT site is shown in Figure 5.30.

FIGURE 5.30

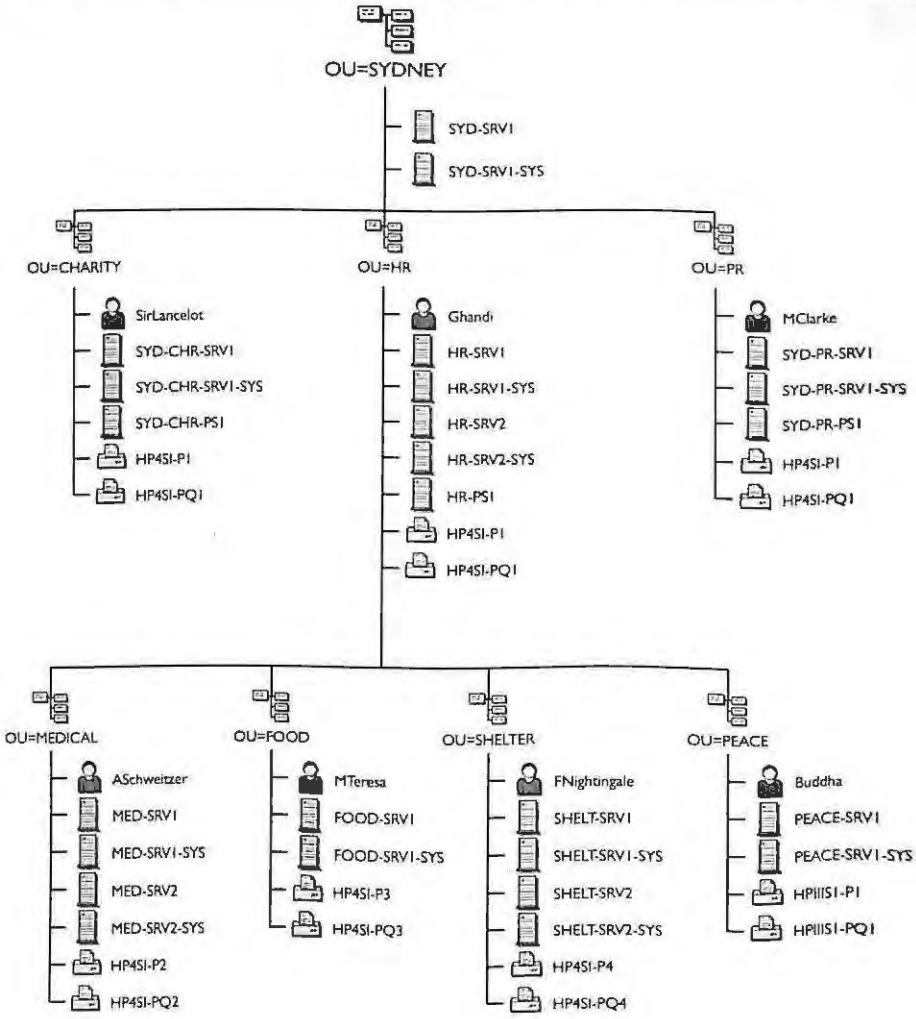
ACME CAMELOT Site



You can also use directory map objects to simplify the administration of your users. For example, the SYDNEY office uses directory maps in all their container login scripts. As versions of their specialized software change, the SYDNEY site administrator changes only the pointer of the directory map object to the new software version. This automatically enables all users in SYDNEY to see the new version of software because all container login scripts use the same directory map. An example of this site is shown in Figure 5.31.

FIGURE 5.31

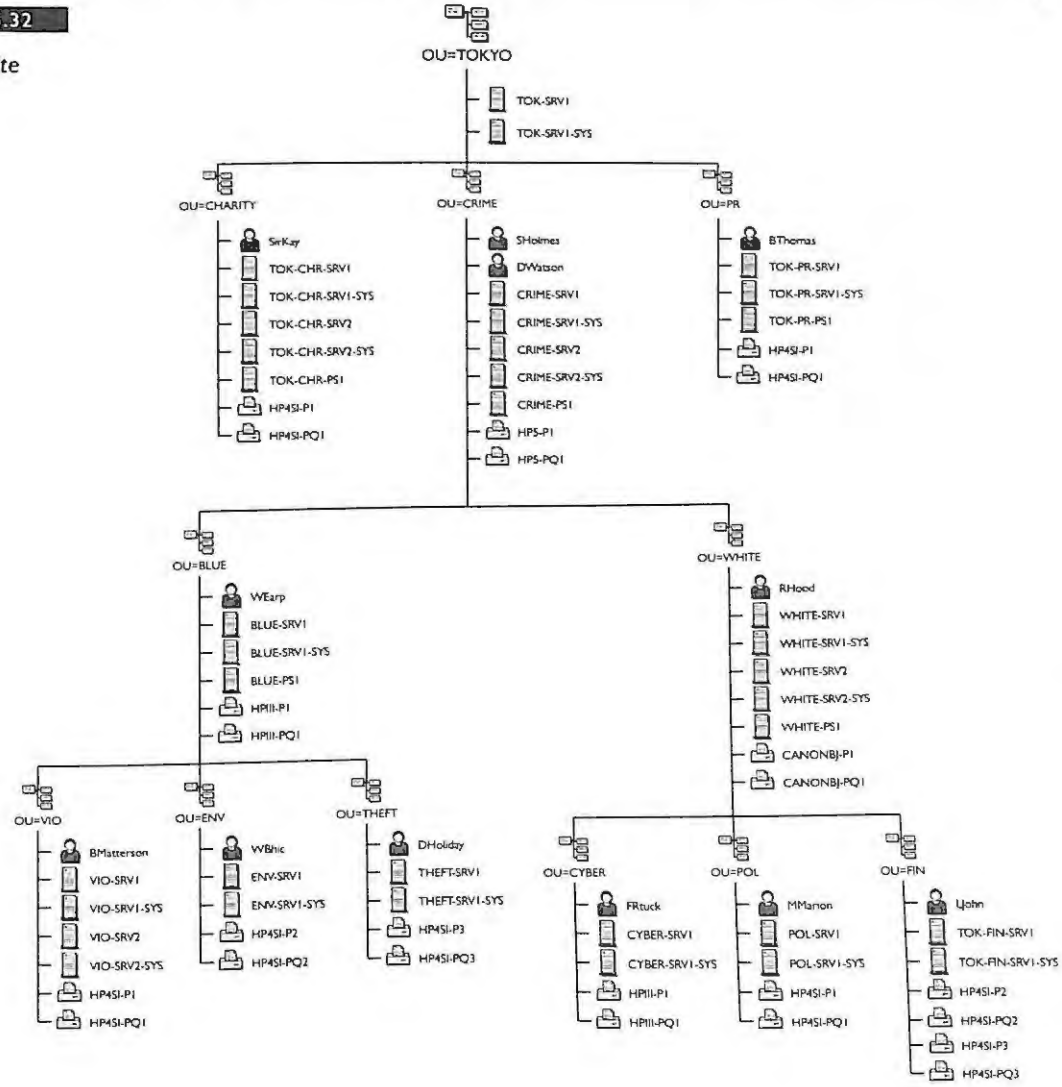
ACME SYDNEY Site



The TOKYO office has traveling users as shown in Figure 5.32. We will create an alias for these users at the top of the tree at O=ACME. With the alias in place a traveling user such as DHOLIDAY in the OU=THEFT only has to remember to log in as DHOLIDAY.ACME. This makes the login process much easier for users who travel but do not carry their own laptop.

FIGURE 5.32

ACME TOKYO Site



Creating Common Resource Containers

Some companies prefer to group similar resources in the same containers, such as a container for all servers, printers, and users. Keep in mind that this approach may work for smaller companies that do not have to group thousands of users in the same container or hundreds of servers together. This design approach works best with smaller companies that want to provide a simple grouping of resources.

DESIGN CONSIDERATIONS

Some LAN administrators may try to design the NDS tree and simultaneously consider all the external factors that may affect the design of the tree. Considering all the factors at once is difficult because the tree will shift and change as you attempt to consider all the design inputs. Some of the most popular distractions are bindery services, partitioning, replication, and login scripts.

Experience has shown that designing the tree is simple if you base the top layers of the tree solely on the WAN infrastructure and the bottom layers according to your organizational information.

When you have completed your first draft design of the NDS tree, you are ready to apply some other design considerations as needed. This process greatly simplifies the tree design effort. It is interesting to note that the design considerations affect only the design of the bottom of the tree, not the top of the tree. This is acceptable because the greatest flexibility for changes in the design is supported at the bottom of the tree.

Again, our approach is to design the bottom level of the NDS tree aligned entirely to the organizations of your company. You can then apply the design considerations as needed. The design considerations that affect the bottom of the tree are:

- Administration
- NDS partitions and replicas
- Login scripts
- Bindery services

Remember that these design considerations apply only to the bottom of the tree; they do not alter the top layers in the tree design. By reviewing each design consideration, you will see how they apply only to the users and other network resources that are contained in the bottom layers.

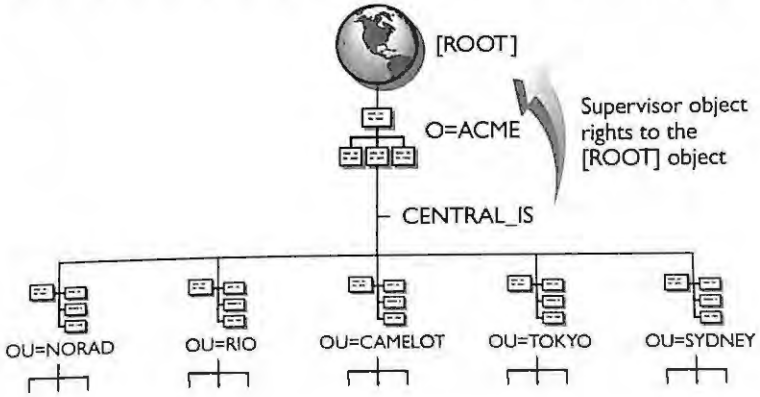
Administration

One of the most important design considerations is how the NDS tree is going to be managed at your company. Are you going to manage the NDS tree as one Information System (IS) group (the centralized approach) or by several different IS groups or people (the decentralized approach)?

Centralized Management The entire NDS tree is controlled by one group in the company. This group manages all the additions and deletions of the NDS objects, partitioning, replication, and everything else related to the NDS database. Figure 5.33 shows how you can centrally manage your tree with one IS group having rights to the top of your tree and down. For more information regarding rights assignments, refer to Chapter 13.

FIGURE 5.33

A centralized management approach with a single group having rights to the entire tree

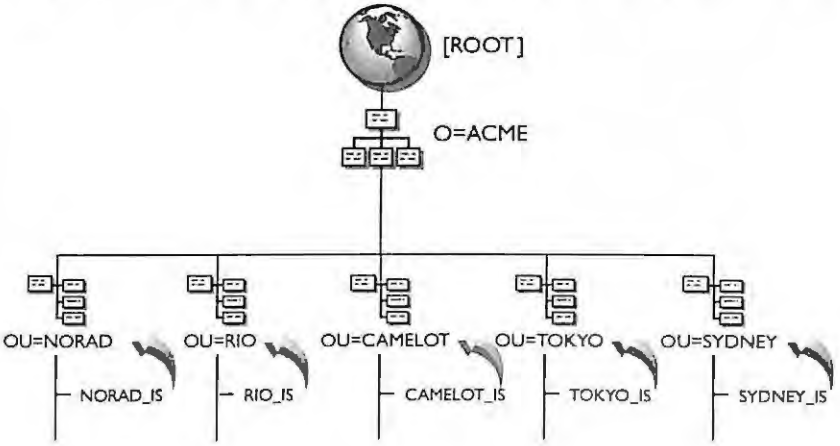


Decentralized Management Portions of the NDS tree are delegated to individuals or independent groups in the company for management. These individuals or groups may be administrators for each department or site administrators responsible for all the network resources in a particular location. All subadministrators should, however, adhere to your previously defined naming standards. Figure 5.34 illustrates how you can set up

administration for the lower containers in your group. You will have a central IS staff managing the upper layers of the tree, with other administrators being responsible for their respective containers.

FIGURE 5.34

Creating a distributed administration approach by creating container administrators. Each subadministrator has rights to his or her own organizational unit.



If the NDS tree is going to be centrally managed, it makes sense to further the tree design to the bottom layers. The central team that provides administration has control of all the objects in the tree from top to bottom.

However, if the tree is going to be decentrally managed, each department administrator or site administrator will decide independently how the tree is organized in that portion of the tree. The top administrators have full responsibility to create the tree down to the department or site and then relinquish control at that layer to each of the independent LAN administrators. Top administrators of the tree will still want to give design guidelines and suggestions to the bottom administrators on organizing the lower containers and grouping network resources. The following is a list of suggestions that you can give your administrators as guidelines. These ideas can also be mandated through the use of access controls as explained in Chapter 13.

Guidelines for Subadministrators

- Subadministrators will have sufficient security over their container to create, delete, or change all objects within their subcontainer.
- Subadministrators will carefully determine if more levels need to be created beneath their container before making changes.
- Subadministrators will do their part to maintain the naming standards as defined by the corporation.
- Subadministrators will not further partition their OU without the assistance of the central IS department nor will they be granted rights to do so.
- Subadministrators will inform the central IS staff before adding a new server into the corporate tree.

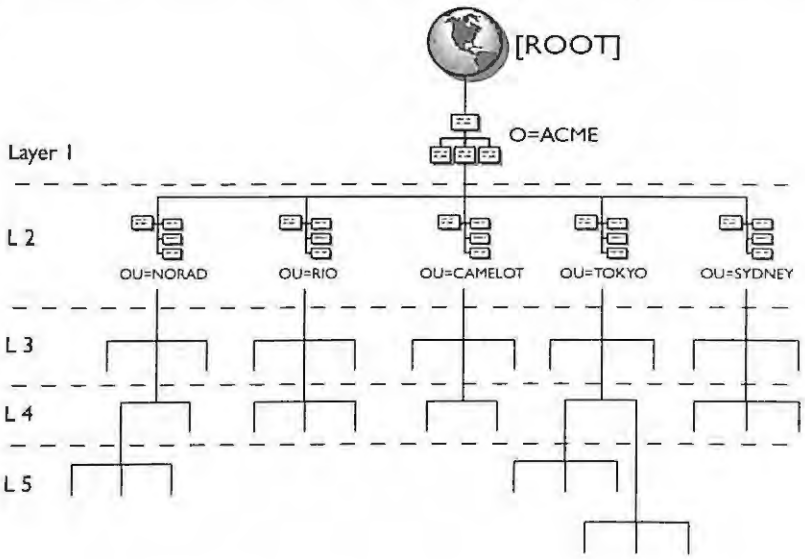
The depth of the tree or number of layers in the tree may be affected by whether your administration is centralized or decentralized. Remember, the recommendation is to build the NDS tree like a pyramid with fewer layers at the top and more layers at the bottom. Centralized administration may imply that a tree designed flat and wide would be easier to administer. If your company has only a few servers and users, you can build a shallow tree that is suited to centralized administration.

This may not be possible if your company is large with many servers, users, and geographic sites. In this case you will need to design the NDS tree with more layers, which means a deeper tree.

For decentralized management of NDS trees, individual administrators along with central administrators can determine the depth of their portion of the tree. Although there is no hard and fast rule regarding the total number of layers in the tree, the NDS tree is more flexible and easier for the user to find information if the tree has three to five layers. Typically, even the largest companies can design a very useful tree with five layers or fewer. Notice in Figure 5.35 that the ACME tree consists of five layers, not including [ROOT].

FIGURE 5.35

The ACME tree consists of five layers, not including [ROOT].



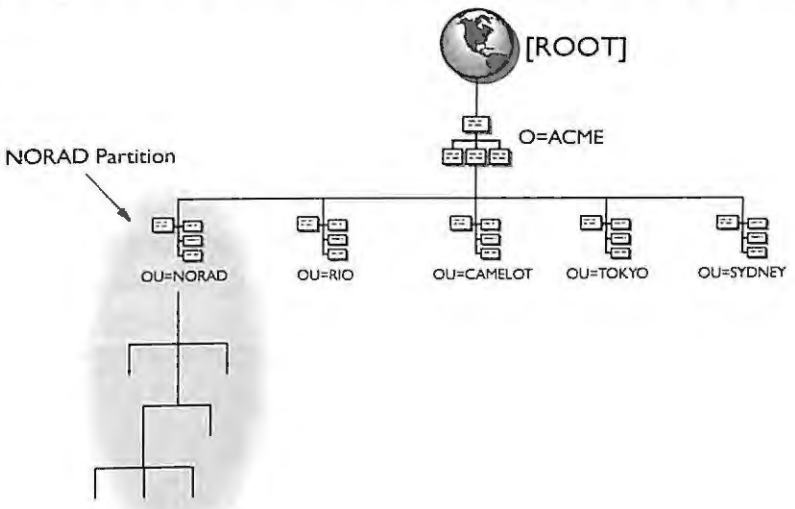
There is a logical limit to the total number of layers in the tree you can access. For example, NDS has a limit of 255 characters for a distinguished name. Thus, the actual limit for the number of layers is dependent on the number of characters in the names of your objects. If your OU names are long, your tree will not be able to have as many layers as it would with shorter OU names. For example, if all the OU names were just two characters then you would be able to have 51 layers (OU=US, 256/5 = 51). We recommend that you give the OUs in your tree short, descriptive names. For more information on naming standards, see Chapter 4.

NDS Partitions and Replicas

The next design consideration you need to address is how you will split the NDS database into partitions. For this discussion, we will consider the size of the partition (total number of objects), the total number of replicas, and where in the tree the partition is to be created. A container object is required for the creation of a partition and is designated the root-most object of the partition. Figure 5.36 shows a partition root called NORAD in the NORAD facility. The partition root is named NORAD because that is the starting point of that partition.

FIGURE 5.36

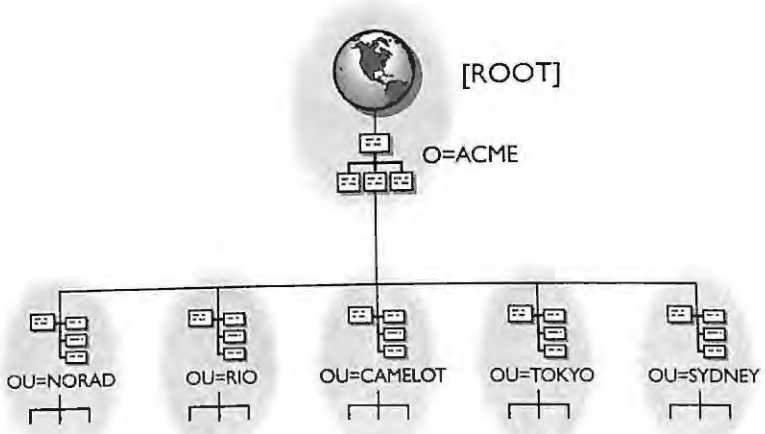
A partition root object is named NORAD in the ACME tree.



When deciding where to create a partition, you should follow the physical network infrastructure. Like the top layer of the tree, the partitions of the tree should represent the WAN, making each site or location its own partition. The benefit of partitioning the NDS database according to the WAN is that the information needed by the local users stays inside that location. We have partitioned each of the location organizational units by site as illustrated in Figure 5.37.

FIGURE 5.37

Each site is its own partition and maintains its portion of the NDS database on its own servers within the site.



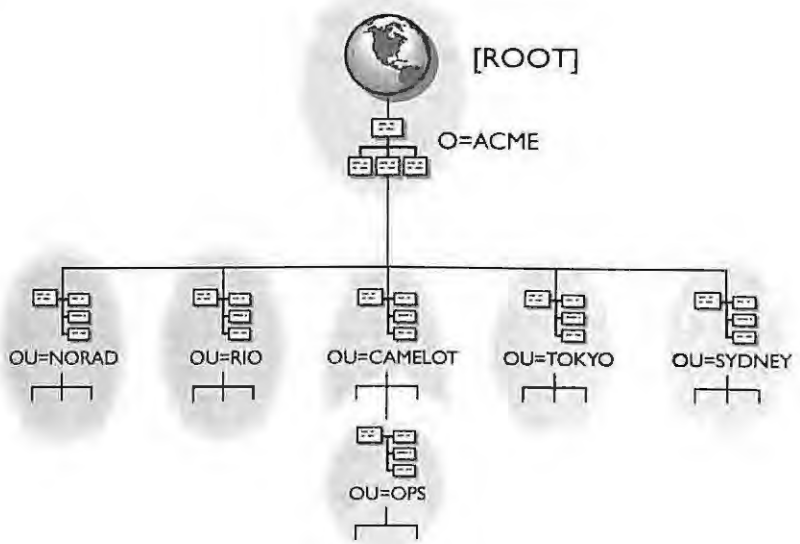
The size of your partitions and the total number of replicas is a design consideration for the bottom of the tree. Typically, partitions range in size from 50 to 3,500 objects. If the partition grows to be significantly larger than 3,500 objects you should probably split the partition in two. Therefore, in Figure 5.38 we have created a new partition called OPS under CAMELOT because that location's partition has grown beyond 3,500 objects. More partitions in the right places provide greater efficiency in your tree design. Remember, a partition contains all the objects in a defined subtree, not just the objects in a single container.

CONSULTING EXPERIENCE

We are suggesting that NDS is more efficient using partitions less than 3,500. Therefore, when your partitions reach this size you can begin to assess the need to split the partitions. Check your user and synchronization performance and use that as a guide. These recommendations are dependent upon the speed of your server hardware. We recommend that your server hardware be a Pentium class machine with 64MB of RAM.

FIGURE 5.38

A new partition is created in the OPS department. It is now a child partition of its parent named CAMELOT.



The next consideration is the total number of replicas of a partition. If the number of replicas is greater than 10, consider creating additional partitions to reduce the total number of replicas. Novell recommends three replicas for each partition. The primary reason you would need more than three replicas of any partition is for bindery services. Bindery services requires a writable copy of the replica. Refer to the "Bindery Services" section later in this chapter.

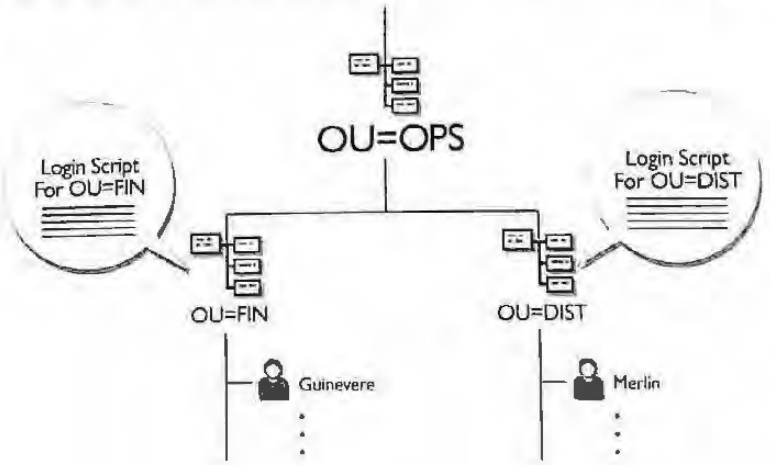
Decide who is responsible for the partitioning of the NDS tree. If you manage the tree centrally, all the partitioning decisions are made by the central IS department. If the tree is decentralized then you may turn over the rights of partitioning the tree to each of the local site or facility administrators. Whichever way you decide to handle partitioning, make sure you decide before installation of NetWare 4.1 begins and make a company policy stating who will handle the partitioning.

Login Scripts

Another design consideration for the bottom of the tree is designing how the users will access the information in the tree. The users will primarily access NDS through the use of login scripts. Remember, the users need login scripts to map network drives and applications, capture to print queues, and set other variables. Thus, the login scripts become a very important design consideration. Typically, the users needing the same login script will be grouped together in the same OU container. You can then use the OU login script to provide users access to the NDS tree. Figure 5.39 shows a container login script that will be used by everyone in the container.

FIGURE 5.39

All users execute the login script in the container where they reside.



You will separate the users that need different login scripts for the same reason. As you design the login scripts for your users, you are in fact designing the organization of the bottom level of the tree.

Another strategy for organizing the login scripts is to have the same login script for all users and copy it to multiple containers. In this manner, the user placement in the tree is not affected. However, this strategy requires that the network administrator be responsible for keeping all copies of the login script the same.

It is recommended that you use the OU container login script to replace the functionality of the NetWare 3 system login script. This will help you organize the bottom layers and containers in your tree.

You can also make use of the profile login script whenever possible for configuring the users for access to resources that are more global in nature. The profile login script enables you to span a single login script across multiple OU containers and assign it to specific users. For example, the container SYDNEY has a profile script created for the HR department, and its subordinate departments of MEDICAL, FOOD, and SHELTER as shown in Figure 5.40 all use the script. The script can reside in any container, and users from any container can execute it.

Through the login script, each user maps network drives to the appropriate network server and establishes access to specific network applications and services. Most login scripts depend on groups and directory map objects for these drive mappings. These groups and directory map objects must be accessible so that any users needing them can find them during the login process.

In order to simplify the mapping to generic network applications (ones needed by all network users), it is appropriate to place the applications in the same subdirectory structure on all the servers. In other words, use the same file structure for all servers. Then when a user maps the drive, he or she does not care which server responds.

For example, assume that each network user needs to have drive mappings to both a word processor and spreadsheet software. These two software packages are installed on all file servers in the same place on the file system (SYS:APPS\WP and SYS:APPS\QPRO). All servers that have a generic file system structure enable the users to map these applications regardless of their location or the server to which they are physically attached. An example of how you can standardize your file system is shown in Figure 5.41.

FIGURE 5.40

A profile script used by users from MEDICAL, FOOD, and SHELTER

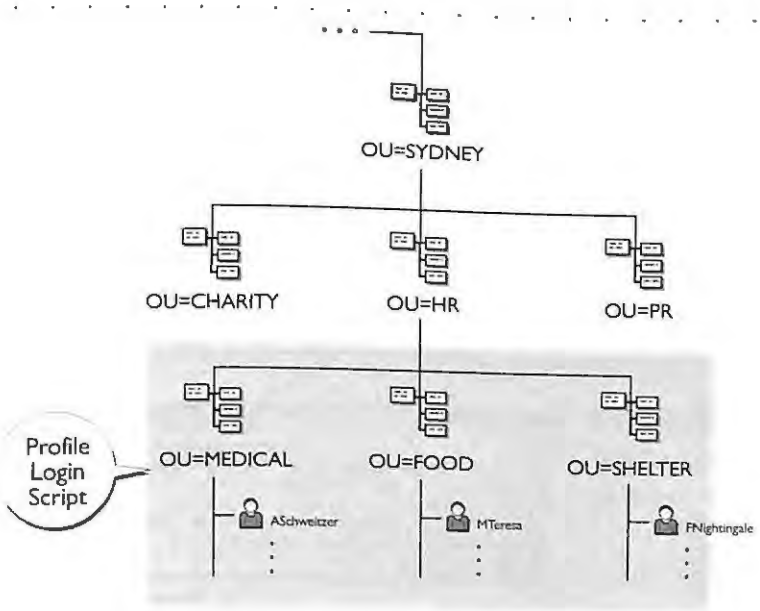
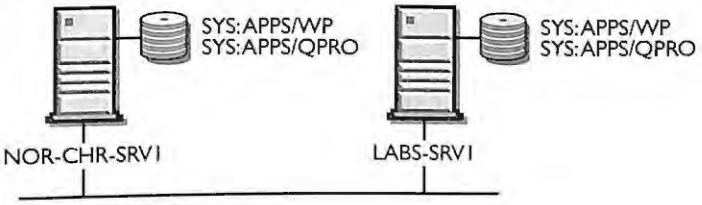


FIGURE 5.41

A standard file system on all your NetWare 4.1 servers will make administration of your network easier.



Bindery Services

NDS provides compatibility with NetWare 2 and NetWare 3 using a feature called bindery services. This feature allows bindery versions of NetWare applications and other third-party software that require the bindery to access the NDS database as if it were the bindery. For example, a client can use the NETX shell (NetWare 3 client) to log in to a NetWare 4.1 server and run any bindery-based application that may exist on the NetWare 4.1 server.

Bindery services can be enabled through the server SET Server Bindery Context command. The server can select one OU, Organization, or Locality container or many containers as the bindery context. The server bindery context is simply the containers the server sees as the bindery. All the leaf objects in the NDS container(s) that are also objects in the NetWare 3 bindery (for example, users, groups, queues, print servers, and

profiles) are seen as objects through the bindery application programming interfaces (APIs). The following figures show how you can set a server's bindery context(s). Figure 5.42 shows how you can use the SERVMAN utility to set the bindery context(s). Figure 5.43 shows how you can verify on a server that the bindery context(s) have been set.

Typing Set Bindery Context at a server shows the string of contexts as valid or invalid. If you want to see only the valid (effective and active) contexts you must type CONFIG at the server console.

FIGURE 5.42

Using the SERVMAN utility to set the server context at OU=CRIME. OU=TOKYO. O=ACME

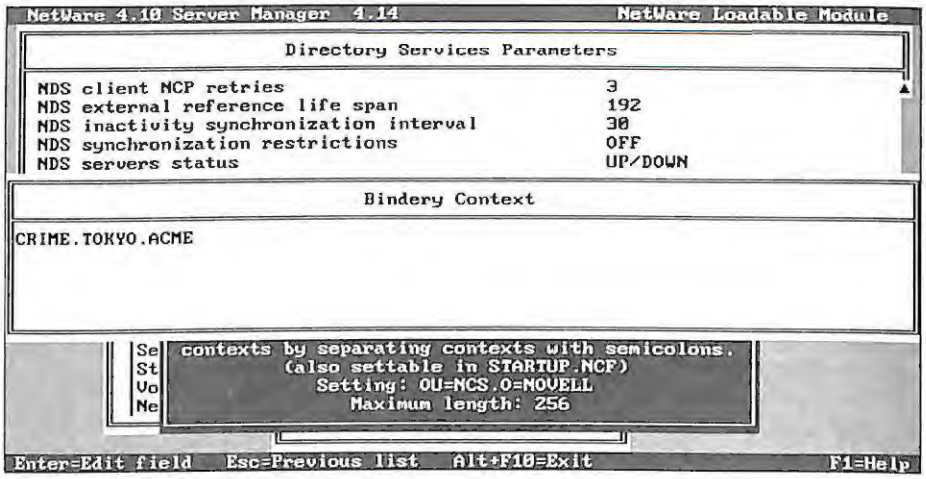


FIGURE 5.43

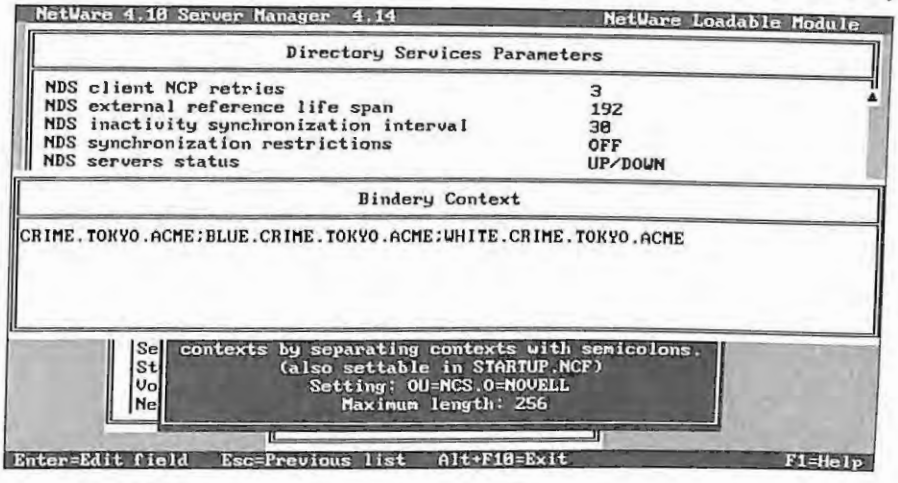
*Typing **SET BINDERY CONTEXT** at a NetWare 4.1 server console will display a server's bindery context(s) if set.*

```
NWIPDEMO:set bindery context
Bindery Context:  OU=NCS.O=NOVELL
Maximum length:  255
   Can be set in the startup ncf file
   Description: The NetWare Directory Services container where bindery
                services are provided.  Set multiple contexts by separating
                contexts with semicolons.
NWIPDEMO:
```

Bindery services in NetWare 4.1 lets you select up to 16 containers as the server bindery context. The major requirement for bindery services is that the server must store at least a read/write replica of the partition where the bindery context is set. If a server has the maximum 16 bindery contexts set, the server would have to store 16 separate replicas just to support bindery services on all contexts. Figure 5.44 shows how we have set multiple bindery contexts on a server in TOKYO to search the organizational units of CRIME, BLUE, and WHITE when a bindery request is made.

FIGURE 5.44

Multiple bindery contexts are separated by semicolons on a server in TOKYO, which will view the CRIME, BLUE, and WHITE containers as its bindery.



As you can see, placing replicas on servers to support bindery services will increase the total number of replicas for each partition. This will affect the tree design because you may be forced to split a partition to reduce the number of replicas. Bindery services is the principal reason for maintaining more than a few replicas of any partition. Refer back to the section "NDS Partitions and Replicas" earlier in this chapter.

Another design consideration is that you should place all the clients or users requiring bindery services from a particular file server in the OU container where you have set the server bindery context. This consideration can affect the NDS tree design at the bottom level because it may require you to combine users and resources of multiple departments or workgroup into one OU. You may need to separate users from the resources they don't use for the same reason. In any event, the organization of the bottom layers of the tree may be affected because the users and resources need to be arranged for bindery services.


NOTE

Before you change the bottom of the tree design to accommodate bindery services, determine whether you even need bindery services for the servers. Remember, bindery services is an optional feature that does not have to be enabled at each server. You should determine if the clients are using NETX or applications that require bindery services. You could also identify the users and key applications and force them to use bindery services on specific servers.

CONSULTING EXPERIENCE

Here is a brief list of some of the applications you may encounter that make bindery calls. This is a brief list, and you need to check all your applications to determine if they require the bindery.

- NetWare 3 Print Services
- Backup Utilities
- Host Connectivity Products
- Menuing Systems
- Network Management Utilities
- Other NetWare 3-Based Applications and Utilities

CHAPTER 12

Understanding and Managing Client Access

"Good counselors lack no clients." Shakespeare

NetWare 4.1 Network Client Software

The network client provides an extension to each network user to access the corporate resources on the servers. The network client is tasked to communicate with the desktop operating system and the network operating system and serves as a liaison between the two. If a desktop application requires the use of network services, the client redirects the output to a server. If a server application needs to communicate with a client, it directs that communication to the network client in response to the client request.

Because of the variety of clients that may exist in your network, your responsibility as a LAN administrator will be to determine what type of client access is needed for a workstation to connect to NetWare 4.1, and to create the appropriate accessibility to meet those needs. Providing access to NDS will be accomplished through the different client software components available in conjunction with NetWare login scripts. The client software provides access to NetWare 4.1, and the login script creates the environment for the user.

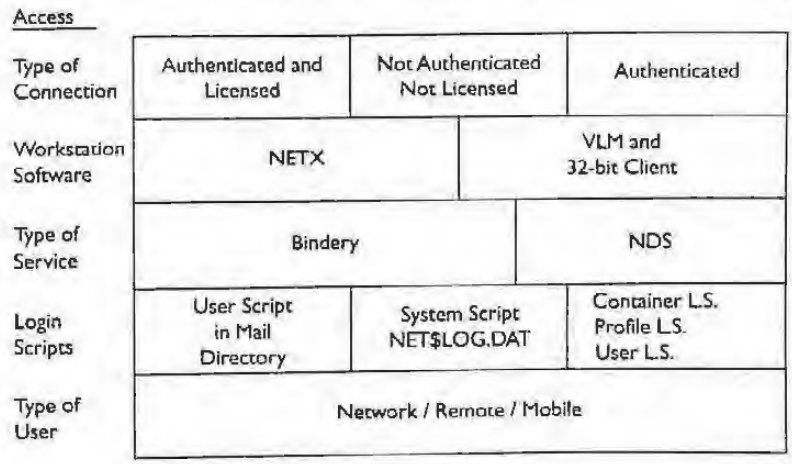
The NetWare 4.1 client software is designed to support connectivity to workstations including DOS/Windows, OS/2, Macintosh, NT, and UNIX operating systems. Through various connections discussed in this chapter, you can connect to a NetWare 4.1 server and access files, applications, and NetWare Directory Services.

As a network designer, you will probably have most of your users running on the same desktop operating system, with a few exceptions that use other systems across your departments. As you design for user access to the NDS tree and NetWare 4.1 servers, you should first begin by designing access (login scripts and so on) for the majority of your user community that uses the same operating system. You can then create your access designs for the less commonly used workstations on your network. You may also have some mobile users that require special login scripts.

This chapter reviews the user login scripts for setting up user environments. We will review container, profile, and user login scripts and how they can be used to meet any user need. Figure 12.1 illustrates the different client access mechanisms and will serve as the basis of our discussion. As shown in the diagram, client access can be broken down into multiple categories. Each of these categories is discussed in this chapter.

FIGURE 12.1

The components of client accessibility for NetWare 4.1 Networks are based on a variety of connections, workstation software, services, login scripts, and users.



Desktop support is key to NetWare 4.1. A server cannot serve the user community unless access is provided for the clients to communicate with NetWare servers across the entire network. This access requires speed and reliability for all users and must make good use of limited memory at each workstation. Once a connection is made, the login script can set up the user's environment. Chapter 18 provides a list of currently supported clients with their latest version of software.

WORKSTATION SOFTWARE

The newest client software is Novell's Client32 architecture, which provides a connection for both DOS/Windows and Windows 95 users. Currently, the most common workstation software used for connecting to NetWare 4.1 is the NetWare DOS Requester, which supports both DOS and Windows clients. The DOS Requester is actually a group of virtual loadable modules that work together to provide client connectivity to a NetWare 4.1 server. The great advantage of this type of software is that you can install only the VLM modules that are required for your user environment. You can create a standard configuration for the majority of your users.

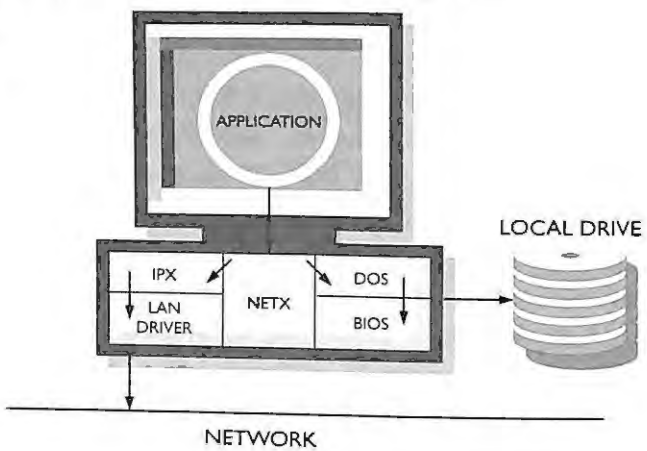
An important difference between the older NetWare 3 NETX.EXE shell and the NetWare DOS Requester is how each client handles network requests. Because NETX.EXE is a shell, all calls from an application to the workstation operating system are intercepted by the shell and then directed to either the network or to DOS.

The NetWare DOS Requester, on the other hand, receives all calls from DOS through the DOS Redirector Interface known as Int2Fh. Therefore, any calls sent to this interrupt are always intended for the network. Each of these approaches has its advantages and disadvantages in terms of memory usage and performance.

A visual example of how the NETX.EXE shell works is found in Figure 12.2.

FIGURE 12.2

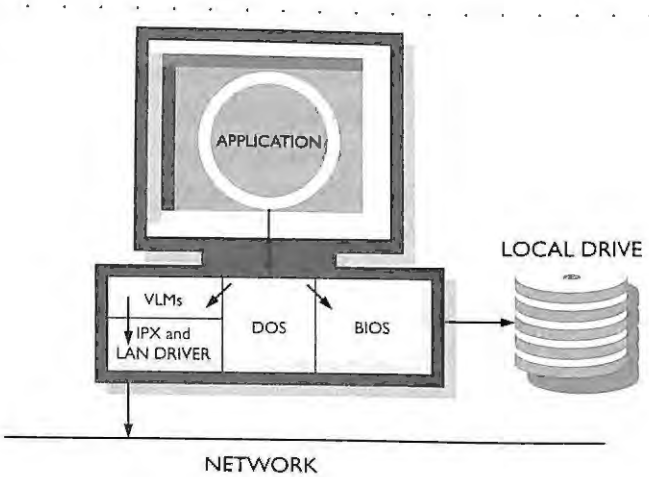
A view of NETX.EXE routing operating system requests from an application



A visual example of how the NetWare DOS Requester receives the network requests from DOS is shown in Figure 12.3.

FIGURE 12.3

A view of NetWare DOS Requester routing operating system requests from an application



TYPES OF SERVICE

Two types of connections exist for NetWare 4.1 — the NDS connection and a bindery services connection. The basic difference between these two connections is that a bindery services connection is server centric. Server centric means that connections to multiple servers require a username and password at each server, and the login process is repeated at every server.

With the NDS connection, on the other hand, you can have a single login to multiple NetWare 4.1 servers. A single login enables the user to enter their name and password once. Any additional drive mappings to other NetWare 4.1 servers will be handled in the background by NDS. The great benefit of the NDS connection is that administrators need to manage only a single user account if they are operating completely on NetWare 4.1 servers.

NDS Connections

An NDS connection requires the use of the VLM client or the NetWare 32-bit Client software for authentication to a NetWare 4.1 server. An NDS connection provides a security mechanism known as RSA encryption between the client and server to provide background authentication for a single sign-on to multiple NetWare 4.1 servers.

An NDS connection is said to be in one of three states:

- Connected but not logged in
- Authenticated
- Licensed/Authenticated

Connected but Not Logged In This state occurs when a user has attached to a NetWare 4.1 server through either the NETX shell or the VLM client. An NDS connection that is not logged in can exist for either NetWare 3 or NetWare 4.1 users to the first attached server. If a connection is made after walking the tree, the state can exist after the first attached server.

For example, when a connection is neither authenticated nor licensed, the users can navigate the NDS tree through the CX (Change conteXt) command. They have attached to a server, but have not yet authenticated.

Authenticated Authentication is a process of proving identity to a server. In NetWare 3, this meant logging in. In NetWare 4.1, the authentication process happens as a "behind the scenes task" at the client.

This type of connection indicates that a NetWare 4.1 server has established a user's identity after the user has entered a correct name and password. Authentication occurs for both NetWare 3 and NetWare 4.1 users, but NetWare 4.1 adds more security to this process.

Authentication is invisible to the user. During the login sequence, the user will enter a password when prompted, and the remaining process occurs behind the scenes. All sensitive data are never transmitted across the wire for security purposes. Authentication relies on encryption algorithms that are based on a public/private key system.

After successful authentication has taken place, a process known as background authentication may occur if the user's login script specifies connections to other servers. A connection to another NetWare 4.1 server, for example, does not require the user to reenter his password. However, all connections are authenticated the same way; the process makes no distinction between the first and subsequent server logins.

Licensed A connection is said to be licensed when a user has made a request of the server, such as mapping a drive or capturing to a printer. Each user will cause the user license to decrement by one after a connection has been licensed. Only an authenticated connection can be licensed.

A combination of these states determines what level a user currently has in a NetWare 4.1 environment.

If users are licensed and authenticated, they can access NDS and file system information to the extent allowed by their rights.

Additive Licensing

Additive licensing increases the total number of licenses on any given NetWare 4.1 server. This enhancement enables administrators to more closely match the number of licensed users to their company's needs. A company that currently has a 100-user license can add a 25-user license to the NetWare 4.1 server to accommodate increased growth. NetWare 4.1 supports 5, 10, 25, 50, 100, 250, 500, and 1000 user versions in any combination.

Bindery Services Connections

The NetWare client software provides compatibility to previous versions of NetWare through bindery services connections. This connection does not provide the capability of a single login to the network. For example, a client using the NETX.EXE shell or the

LOGIN/B option with VLMs to log in to a NetWare 4.1 server must enter a username and password for that server. Additional connections to other NetWare 4.1 servers would require the user to enter another username and password.

Bindery services can be enabled on any NetWare 4.1 server through the SET BINDERY CONTEXT command. The server can select one container or multiple containers in the NDS tree to set as the bindery context. All the leaf objects in the NDS container(s) that are also objects in the NetWare 3 bindery (for example, users, groups, queues, print servers, and profiles) are seen as the bindery.

Bindery services in NetWare 4.1 allows you to select up to 16 containers as the server bindery context. Bindery services requires that the server store at least a read/write replica of the partitions where the bindery context is set.

LOGIN SCRIPTS

Network users will execute login scripts to access NetWare 4.1 servers and other network resources. Traditionally, login scripts were used to establish the user's network environment. Login scripts for NetWare 4.1, however, are used to map network drives, map to applications, capture to printers and print queues, and set other important environment variables. Login scripts are the standard mechanism for user access and may require careful consideration.

When a user logs into the NetWare 4.1 network or server, login scripts associated with the user are executed. There are two categories of login scripts available to the user of the NetWare 4.1 network — NDS login scripts and bindery-based login scripts. NDS login scripts support the Directory connections, and bindery-based login scripts support the bindery services connections.

Our focus on login scripts in this chapter is to provide information on designing access to NetWare 4.1, not to encompass every login script variable and command. (Definitions and functions of all variables and commands are included at the end of this chapter.) Well-designed login scripts will help you create effective working environments for your users.

It is recommended that you execute the login scripts before launching Windows 3.1. If there are users who access Windows immediately after they boot their workstations, you need to have them log in to the network, which runs the login scripts before

launching Windows 3.1. The exception to this tip is if you are running the new NetWare Client32 software that provides a Windows login utility. See Chapter 18 for more information on NetWare Client32.

Login scripts execute in a specific order as shown in the following sequence of login script execution for NetWare 4.1:

1 • User logs in to a server.

2 • User executes container script if available.

3 • User executes Profile if that user's Profile property is set.

4 • User executes User Script if available.

5 • If user login script is not available, then the user executes the default script.

Bindery-based Login Scripts

The bindery-based login scripts that you may place in NetWare 4.1 are the same bindery login scripts found in NetWare 2 and NetWare 3. These login scripts can be copied onto the NetWare 4.1 servers to provide bindery services scripts to your NETX.EXE clients. For example, a user attaching to a NetWare 4.1 server with the NETX.EXE workstation software will have a bindery connection and look for the system and user login scripts on the server. The user script is in the SYS:MAIL directory, and the system login script would be placed in the SYS:PUBLIC directory as NET$LOG.DAT. Even if the user is using the VLM workstation software and selects the LOGIN/B option, that user will be attached to the server as a bindery connection.

The system login script is used for commands that affect all the bindery-based users on that server. Commands that might be placed in the system login script include the commands for displaying messages, mapping network drives and search drives, and setting environment variables. The system login script is the best place to manage the mapping and capture statements for all the bindery users that may still exist on your NetWare 4.1 network.

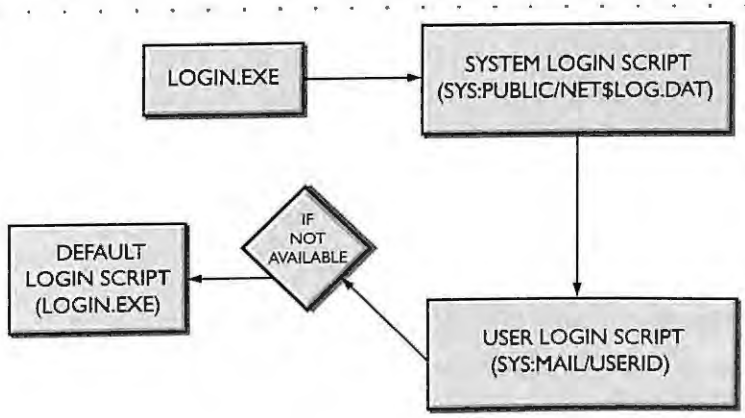
After a user successfully attaches to the server with a bindery connection, the system login script will execute from SYS:PUBLIC\NET$LOG.DAT if it exists. If the user login

script is present it will execute from the SYS:MAIL\USERID subdirectory. If the user login script does not exist then the default login script is executed. The default login script is hard coded into the LOGIN.EXE program.

Figure 12.4 shows the order of execution for bindery-based login scripts. If you are familiar with NetWare 3, notice that the bindery-based login scripts for NetWare 4.1 are executed in the same order.

FIGURE 12.4

The bindery-based login scripts in NetWare 4.1 are executed in the same order as NetWare 3 login scripts.



The individual user login scripts are stored in each user's mail subdirectories on the SYS volume of any server where a bindery account exists. For example, user GWASHINGTON (with object ID of 19000023) stores the bindery-based user login scripts in the SYS:MAIL\19000023 subdirectory.

The individual user login script customizes the user environment to the specific needs of the user. The same commands placed in the system login scripts can also be placed in the individual user login scripts. It is recommended that the user login scripts be used only in situations in which the system login script will not suffice.

The bindery-based login scripts are server centric, meaning that they are used only if a bindery user logs in to the server that is holding them. Because the login scripts are server-centric, there are not alot of design issues to consider. However, you should try to move all the users to the NDS login scripts as soon as possible so that you have fewer scripts to support.

You can make changes to both the system and user login scripts using the NetWare 3 SYSCON.EXE utility. You can also edit the NET$LOG.DAT file (or any script) directly with any text editing program. Although the bindery-based login scripts can be edited, any changes you make to the scripts are not automatically synchronized to the

corresponding NDS login scripts. You can use Novell's NETSYNC utility to synchronize login scripts if you need to maintain consistency between NetWare 3 and NetWare 4.1. Refer to Chapter 2 for more information on NETSYNC.

NDS Login Scripts

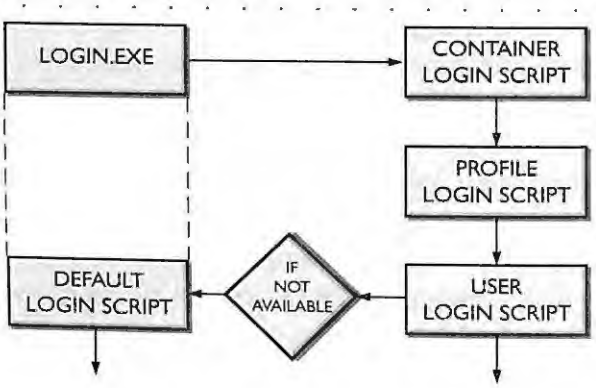
The login scripts used in NDS are different than those used by bindery services. The NDS login scripts are a property of an object and are accessible only through an NDS connection. The only NDS objects that have the login script property are the container objects (O=Organization and OU=Organizational Unit), profile objects, and user objects.

Users who obtain an NDS connection to the network and run LOG.EXE will execute the container login script in which the users reside. The container script is roughly equivalent to the NetWare 3 system login script. After the container login script is executed, a profile login script can be executed if the user is associated with one. The user may also have a user script, which is executed after the container and profile scripts. If no user login script exists, the user will execute a default script. Again, the default login script is hard coded into the LOGIN.EXE program.

Figure 12.5 shows the order of execution for the NDS login scripts. Notice the profile login script, which falls between the container and user login scripts.

FIGURE 12.5

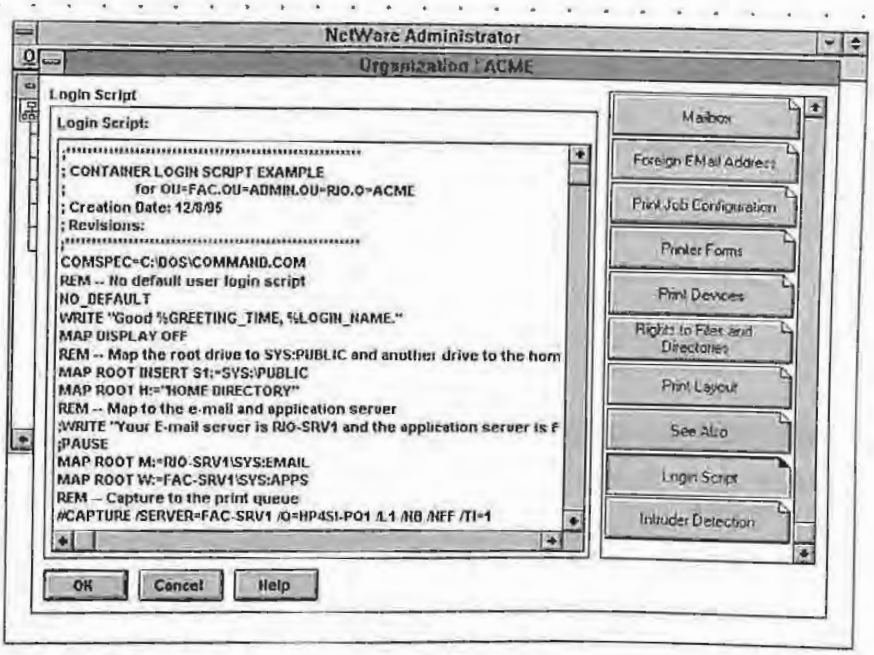
The order of execution for the NDS login scripts in NetWare 4.1



In order to manage the NDS login scripts, you can use either the NWADMIN or NETADMIN utility that ships with NetWare 4.1. With these utilities, you can create and edit all the login scripts, except for the default login script. You can also add a profile login script for execution by selected users. An example of editing a login script is shown in Figure 12.6. In this example, the administrator is editing a container login script for the FAC container.

FIGURE 12.6

Adding or modifying an NDS login script can be done by using the NWADMIN utility.



NDS Container Login Script Typically, the NDS users needing the same network resources will be grouped together in the same NDS container. These users will probably need similar drive mappings and capture statements to establish access to the network resources with which the users are grouped. You can then use the container login script to provide users access to the NDS tree in this fashion.

CONSULTING EXPERIENCE

It is highly recommended that you use the NDS container login script to replace the functionality of the NetWare 3 system login script. The container login script is stored as a property of the O=Organization object, the OU=Organizational Unit object and other container objects. Maintaining one set of login scripts allows for easier administration.

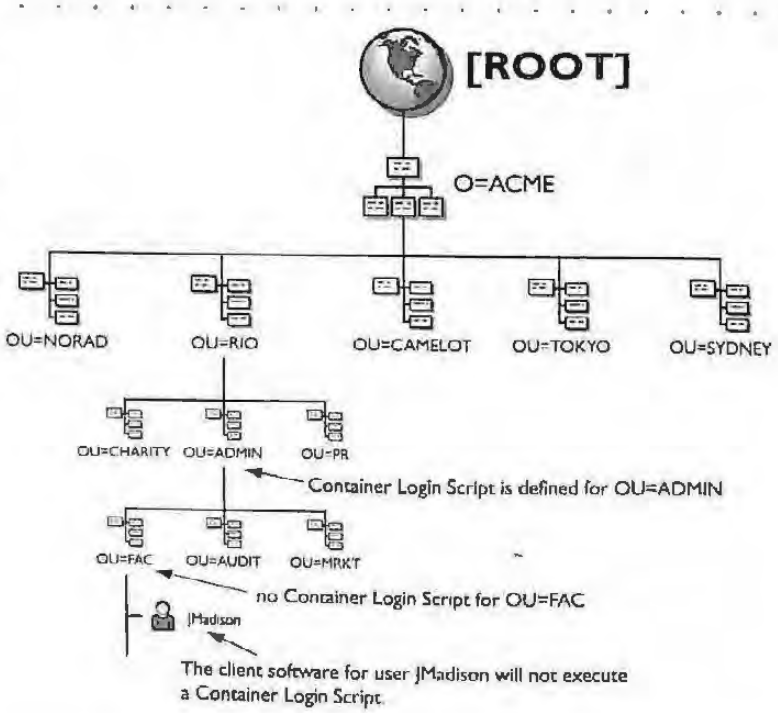
After the LOGIN utility has authenticated a user to the NetWare 4.1 network, the program checks the container login script in which the user resides. If the container

login script exists it will then be executed. NDS will search only the immediate container (O or OU) in which the user is a member. If the container login script is not defined, the system will not automatically search higher in the tree for another container login script.

For example, Figure 12.7 shows the user JMadison in the ACME tree named under the container OU=FAC.OU=ADMIN.OU=RIO.O=ACME. The container OU=FAC does not have a login script. There is, however, a container login script defined higher in the tree at OU=ADMIN. The user JMadison is an occupant of OU=FAC and will not execute a container login script because there is not one currently in FAC. The user JMadison will not search up the tree for a container login script. In other words, no other container login scripts above the user JMadison will be executed.

FIGURE 12.7

Container login scripts are executed only for the immediate occupants of a container. In this example, the user JMadison will not have a container login script.



The NDS container login script commands should establish the network environment of the users. These commands include the network drive mappings, printer and print queue captures, and other environment settings. The users in the container are best managed by using the NDS container login script.

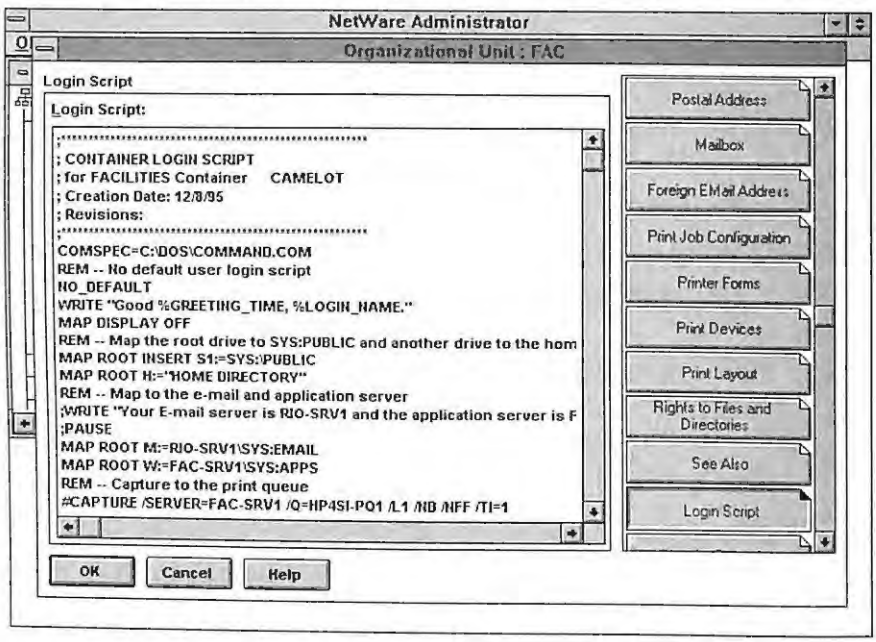
The following is an example of a container login script that we have just added for the OU=FAC container in the RIO location of ACME tree.

```
;*********************************************************
; CONTAINER LOGIN SCRIPT EXAMPLE
; for OU=FAC.OU=ADMIN.OU=RIO.O=ACME
; Creation Date: 12/8/95
; Revisions:
;*********************************************************
COMSPEC=C:\DOS\COMMAND.COM
REM - No default user login script
NO_DEFAULT
WRITE "Good %GREETING_TIME, %LOGIN_NAME."
MAP DISPLAY OFF
REM - Map the root drive to SYS:PUBLIC and another drive to the
home directory
MAP ROOT INSERT S1:=SYS:\PUBLIC
MAP ROOT H:="HOME DIRECTORY"
REM - Map to the e-mail and application server
;WRITE "Your E-mail server is RIO-SRV1 and the application server
is FAC-SRV1"
;PAUSE
MAP ROOT M:=RIO-SRV1\SYS:EMAIL
MAP ROOT W:=FAC-SRV1\SYS:APPS
REM - Capture to the print queue
#CAPTURE /SERVER=FAC-SRV1 /Q=HP4SI-PQ1 /L1 /NB /NFF /TI=1
```

The previous script can be created in either the NWADMIN utility or the NETADMIN utility as shown in Figure 12.8.

FIGURE 12.8

Using the NWADMIN utility to create a script for the FAC container



NDS Profile Login Script If a user has a profile script assigned, it will be executed immediately after the container login script. The profile login script is optional and is used in special cases or for groups with special needs. The profile has the capability to include users that are in different containers in the tree. Its purpose is to assign additional environment settings that you may not want to assign to everyone in the profile. The scripts and the commands used in the profile login script are identical to the NDS container login scripts.

Because the profile login script is a special-purpose login script it can provide you with greater flexibility during the login process. Multiple users can be associated with the profile login script and they can reside in different containers in your tree. Figure 12.9 shows the creation of a profile login script through the NWADMIN utility. You can enable users to execute the profile script by assigning individual users to the script by making the assignment to the user object at creation. This method can be accomplished through the NWADMIN or NETADMIN utilities as shown in Figure 12.10.

FIGURE 12.9

Assigning an individual user to execute a profile login script

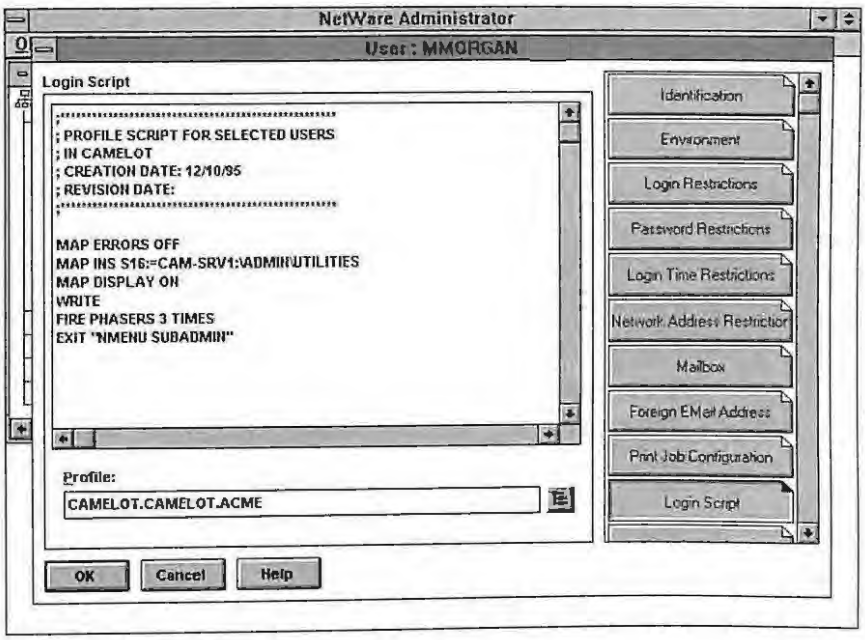
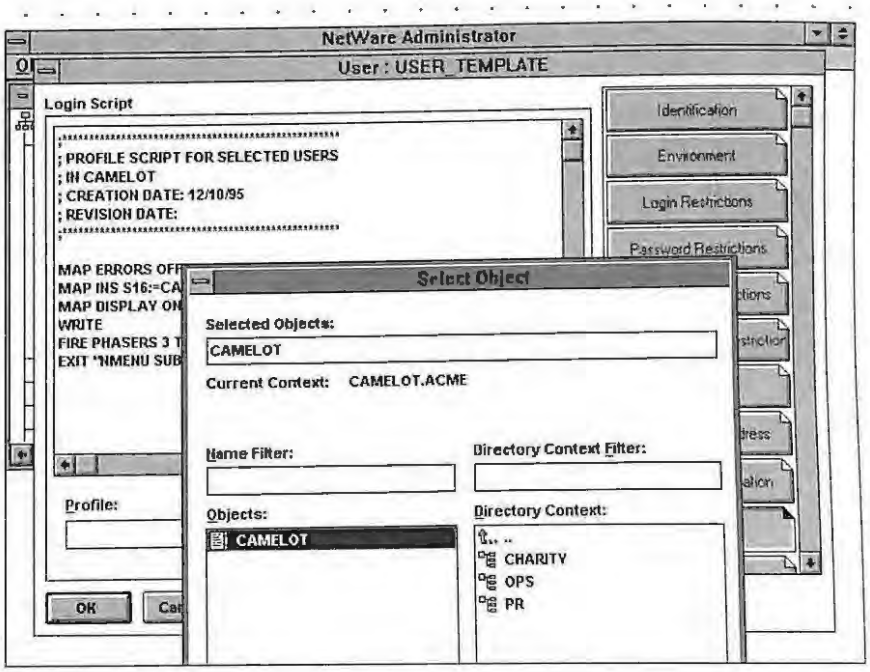


FIGURE 12.10

Creation of a profile login script through NWADMIN



Typically, the profile login script is used:

- For an entire company (if the company and tree are small)
- To create a user environment based on location
- For a special group of users

The profile script is represented by the profile object, which can be placed anywhere in the Directory tree. Using the profile login script to span all the users in the entire company is recommended only for small networks. A small network consists of fewer than ten servers and is not widely distributed over wide area links. We recommend using the profile login script across small networks only because this method is expensive in terms of the NDS traffic that will be generated to manage this object.

The profile login script is also used to create a user environment for a location. This method is similar to simply using a container script, except that you may want to create an environment for specific users within the container. Users that are organized around a particular site, building, or floor can have specific environment settings based on their particular needs. With the profile login script it is not necessary to create floor or building containers in your tree just to have a common login script.

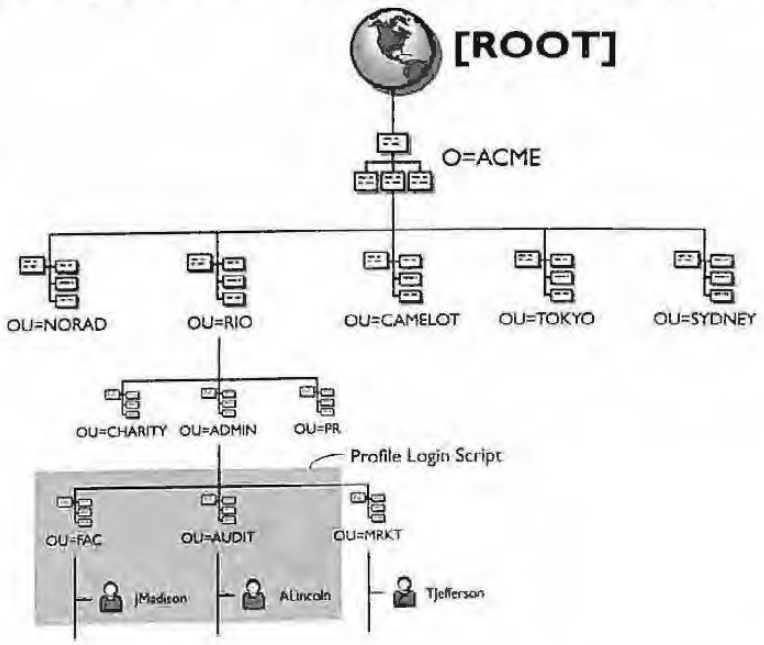
For example, you have a specific set of users in a building and you want them to always map to the same e-mail and applications servers. You may also have users on one floor who want to capture to the same printers. You can accomplish these tasks using the profile login script. Using the profile login script in this fashion is considered to be a locational use of the script.

Figure 12.11 illustrates a profile login script being used as a locational login script that includes the containers in the ACME tree below the OU=ADMIN container in the RIO location. Specifically, the OU=FAC and OU=AUDIT are in the same building and need similar environment settings.

The profile login script can also serve as a special-purpose login script for a group of users. The group members can all be in the same NDS container or they can span across a number of OUs. If the profile login script makes assignments for users within a single OU, then the profile login script is similar to a group object with its sole purpose of executing a script.

FIGURE 12.11

The users in both the OU=FAC and OU=AUDIT containers are using the same profile login script.



A more powerful use of the profile login script is to span more than one NDS container. In Figure 12.12, you will see three users in a different NDS container. These users are members of a special group of administrators who need specialized access to resources in the NDS tree. When each user logs in to the network, he receives the additional drive mapping to perform various job functions. Users JMadison, ALincoln, and TJefferson are each associated with the same profile login script that gives them the extra drive mappings.

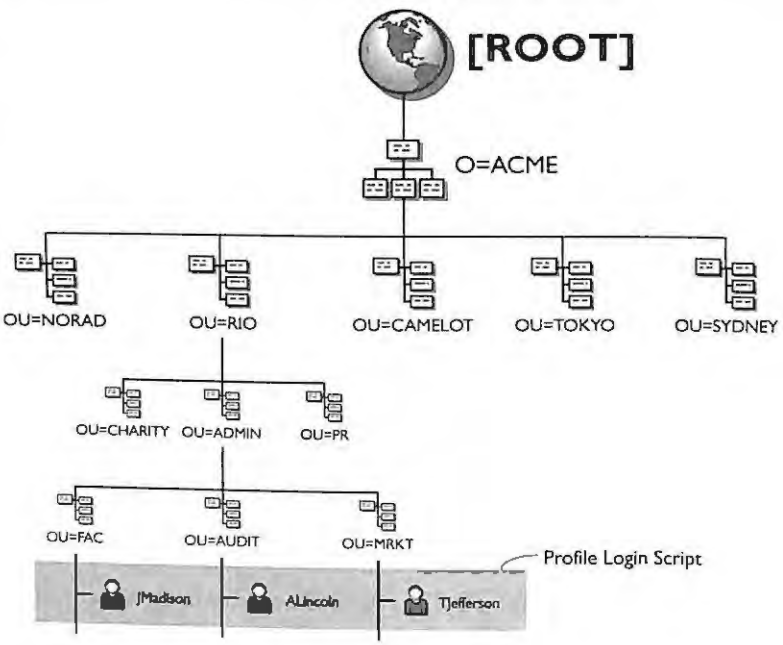
NDS User Login Script The NDS user login scripts are stored as a property in each of the user objects. Like the bindery-based user login script, the NDS user login script customizes the user environment to the specific needs of that user. All the login script commands and variables can be used in the individual user login scripts. However, we strongly recommend that user login scripts be used only when the commands in the container system login script are not adequate.

Most, if not all, scripting can be accomplished through container login scripts. For most large NetWare environments it is generally

unfeasible to implement user scripts because of the difficulty in maintaining them. As a network administrator you don't need the extra work.

FIGURE 12.12

The users JMadison, ALincoln, and TJefferson are each associated with the same profile login script.



If you decide to let your users have personal login scripts, keep in mind that maintaining all the users' login scripts will be a difficult task.

The following is an example of a user login script for JMadison in the OU=FAC container in the RIO location of the ACME tree. These users support their own user login scripts and do not request the assistance of a network administrator.

```
MAP DISPLAY OFF

MAP ERROR OFF

MAP F:=FAC-SRV1\SYS:USERS\JMADISON

;***** EMAIL *****

SET EMAILUSER = "JMADISON"

MAP M:=RIO-SRV1\SYS:POSTOFF
```

```
;***** PRINTERS *****
#capture L=1 Q=HP4SI-PQ1 NB NFF TI=1
;***** WINDOWS *****
MAP ROOT W:=FAC-SRV1\SYS:APPS\WINDOWS
;***** BRIEF EDITOR FLAGS *****
MAP S16:=FAC-SRV1\SYS:APPS\BRIEF
SET BPACKAGES = "c:t;h:t,r"
SET BPATH ="z:\\apps\\brief\\macros"
SET BHELP = "z:\\apps\\brief\\help"
SET BBACKUP = "c:\\backup"
SET BFLAGS = "-i70 -u300 -l200 -Dega -kl M"
SET BFILE = ""
SET BTMP = "c:\\tmp"
SET BCC = "\"cl /c %s.c\""
SET BCH = "\"cl -c -Tc %s.h\""
;***** WP *****
MAP S16:=FAC-SRV1\SYS:APPS\WP\6.1
SET WP = "/U=JM"
;***** Misc. *****
MAP S16:=FAC-SRV1\SYS:APPS\PROGRAMS\BIN
MAP
```

Default Login Script The default login script is executed only when the user login script does not exist. The default login script is hard coded into the LOGIN.EXE program and tries to create enough drive mappings to the server so that the user can function properly. The purpose of the default login script is to back up the absence of a user login script. The default login script will execute even if you have a container or profile login script.

If you do not want to run the default script, then you need to place the NO_DEFAULT command in the container login script or in a profile script. The container login script can also have an EXIT command at the bottom of the script. The EXIT command prevents any other login script from running, including the default login script.

The following commands are executed as the default login script:

```
WRITE "Good %GREETING_TIME, %LOGIN_NAME."

MAP DISPLAY OFF

MAP ERRORS OFF

MAP *1:=%FILE_SERVER\SYS:

MAP *1:=%FILE_SERVER\SYS:%LOGIN_NAME

IF "%1" = "SUPERVISOR" || "%1" = "ADMIN" THEN MAP
*1:=%FILE_SERVER\SYS:SYSTEM

MAP INS S1:=%FILE_SERVER\SYS:PUBLIC

MAP INS

S2:=%FILE_SERVER\SYS:PUBLIC\%MACHINE\%OS\%OS_VERSION

MAP DISPLAY ON

MAP
```

MOBILE OR TRAVELING USERS

NetWare 4.1 lets the user log in and access resources from anywhere in the network. This feature helps you to manage the mobile or traveling user more easily. In order to completely support traveling users and their specific computing requirements, you will need to consider the following questions:

- Does the user carry a laptop computer?
- Where is the user geographically located?
- Where is the user's home office?

As users move from one location to another, they access the network and its resources differently. Knowing how each user wants to access the network will help you set up the user environments. For example, some users just want dial-in access to the network from remote locations. These locations can range from their homes, hotel rooms, and even airplanes. Typically, this type of user dials into the network from a laptop or home computer.

Some users may travel from one office to another and need full access to all the local network resources of the office they are visiting. Although the users need full access to the local resources, they still want the data from their home directory and server. Essentially, the definition of a traveling user is broken into two types: remote users and mobile users.

CONSULTING EXPERIENCE

Your approach to designing access for NetWare 4.1 should be to first design for the majority of the users and then design for the traveling users. In order to design the access properly, you need to know how many users in the network are traveling users. Then determine from the total number of traveling users how many are remote users and how many are true mobile users.

Remote Users

The remote users are the individuals who travel or carry a laptop computer and simply access the network resources through dial-in. The remote user who takes a laptop on the road is usually self-contained, meaning that the laptop computer is configured with all the necessary applications software. The user can continue to work when on the road and merely dials into the network to transfer e-mail messages, download files, or briefly access other resources.

Remote users require less design considerations for access because they will access the NDS tree only as needed for a connection to the network. Supporting remote users will not impact the design of the Directory tree or require you to create any special NDS objects. Users simply dial into specific predetermined access points in the network and use their normal NDS context or location. After the normal login to the network, the users can download files and access other necessary resources.

Some remote users dial in just to transfer their e-mail messages. Typically, a company may dedicate special phone lines for just the remote e-mail users. These lines may have their own security and access method and would not affect the Directory tree access.

If a remote user travels to another office and plugs his laptop computer into the network and wants access to all the local resources, he has really become a mobile user. The design considerations for the mobile users are addressed in the following section.

Mobile Users

The mobile users are individuals who travel from one office to another or from one computer to another. They expect full access to all the local network resources of the office they are visiting while maintaining the ability to access data from their home server. The mobile user may not carry a computer (laptop) with him, but expects to have a computer available at the other site. Some mobile users decide to carry laptop computers and plug them into the network when they arrive. Thus, the best definition of a mobile user is an individual who uses a computer on the network from a location that is away from his home office.

Whether the user travels thousands of miles or across the building, the issues are the same for moblile users. The user wants access to the network applications, such as word processing, spreadsheets, e-mail, and printing from the local servers, but also wants to retrieve the data from his home server. The user wants these capabilities to be as seamless as possible.

CONSULTING EXPERIENCE

Users who carry laptop computers to a new location are not considered mobile users if they do not need access to the local network resources. If the users are content to access their home resources across the network, then they are simply remote users. There are no special design considerations for remote users. Remember, NetWare 4.1 enables the users to login from anywhere on the network. A user simply looks for a network connection and logs in to the server at his home office.

In order to support the needs of the mobile user, you need to answer the following questions:

- Where is the user geographically located?
- Where is the user's home office?

There are several mechanisms in NetWare 4.1 to help you answer each of these questions. These mechanisms include the NDS name context, alias objects, configuration files, login scripts, login script variables, and environment variables.

NDS Name Context The name context in NDS helps you determine where in the NDS tree the user belongs. The name context is important because NDS requires it for every user logging in to the network. The context can be set in the user's NET.CFG file or by typing the user's full name during login as well. While a mobile user's physical location may change, his context will remain constant.

If the mobile user has traveled without a laptop computer, he expects to use any available computer in the office he is visiting and log in to the network. The main issue with this scenario is how to determine the user's name context for login purposes. There are several ways to work around this problem. The mobile user can manually enter the context at the computer console before login, you can create alias objects that point to the user in his normal context, and the name context of the alias can be set in the workstation configuration file.

Manually Changing the NDS Name Context The first option involves the mobile user manually entering his name context into the computer he is using. This option assumes that the user understands how to use the proper utilities and is familiar with their complete context in the NDS tree. The CX (Change conteXt) utility is used to set the user's context before login.

For example, the user JMadison in the ACME tree as shown in Figure 12.13 would need to set his name context by typing:

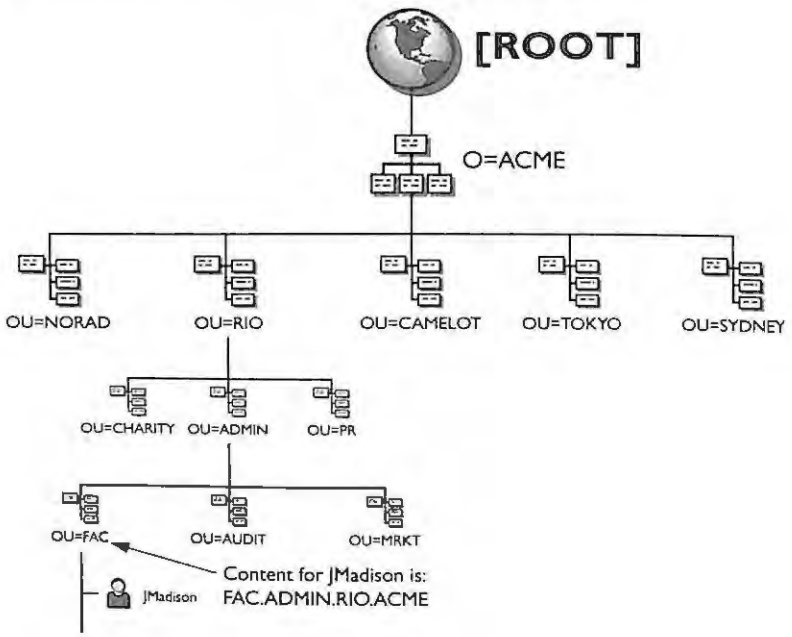
```
CX .FAC.ADMIN.RIO.ACME
```

Notice the leading period in the CX command line. The leading period tells the utility that this is a distinguished name and to start at the [ROOT] object when setting the name context. This is a little easier than trying to figure out where your current context is set in the tree.

The CX utility is stored in the LOGIN subdirectory on the server. The user must have a connection to a server and be in the directory (typically the F: drive) or type the path before running this utility.

FIGURE 12.13

The name context for the user JMadison in the ACME tree



Using an Alias Object to Help Set the Name Context The second option that could help set the name context is the use of alias objects. If you have a small number of mobile users you can create an alias object below the O=Organization for each mobile user. The alias would point to the user's primary object in the appropriate container.

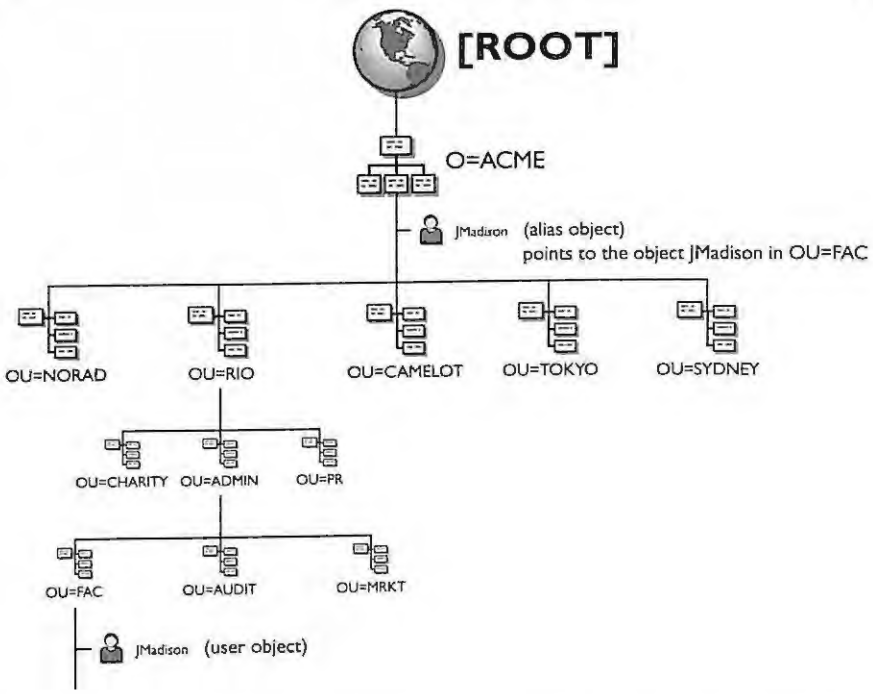
The value of this strategy is that it creates a simple context for each of the mobile users. The users do not need to know where the context is or even how to set it. The users would simply enter the name of the alias object during the login process.

For example, an alias object has been created for the user JMadison in the ACME tree. Figure 12.14 shows that the alias object called JMADISON was created directly in the O=ACME container. The alias object points to the real object in the OU=FAC in the RIO location. When the user JMadison wants to log in to the network from any site, he uses the name of the alias object as follows:

```
LOGIN .JMADISON.ACME
```

FIGURE 12.14

An alias object called JMADISON has been created under the O=ACME container.



Notice the leading period in the LOGIN command line before the name of the alias object. The leading period instructs the utility to start at the [ROOT] object when looking for the alias object.

This method of using the alias object to support the mobile users works well if you have a small number of mobile users at your site. Setting up an alias object for each individual mobile user is feasible if the total number is small. This method may not work if your mobile user population is high. You will need to determine how many alias objects you can manage.

NOTE

You can also use third-party utilities that search the tree for a user object and set the approximate context for your mobile user. You should consider these types of utilities when you have many mobile users.

Using the Configuration Files to Set the Name Context The name context for the user can be set using the standard workstation configuration file called NET.CFG. The

NET.CFG file is read during the loading of the workstation client. The following is an example of setting the name context for the user JMadison in the ACME tree. Within the NET.CFG file there is a section called the NetWare DOS Requester where the NAME CONTEXT = "OU.FAC.OU=ADMIN.OU=RIO.O=ACME". Users traveling to different locations with their own notebook or laptop will typically have the NET.CFG file set already. When they arrive on site and connect to the network, their name context is resolved from setting in the NET.CFG file on the laptop.

```
Link Support
    MemPool 6192
    Buffers 10 1580
    MAX STACKS 8

Link Driver NE2000
    INT 5
    PORT 300
    MEM D0000
    FRAME Ethernet_802.2

NetWare DOS Requester
    NAME CONTEXT = "OU=FAC.OU=ADMIN.OU=RIO.O=ACME"
    PREFERRED SERVER = FAC-SRV1
    FIRST NETWORK DRIVE = F
    NETWARE PROTOCOL = NDS,BIND
    SHOW DOTS = ON
    USE DEFAULTS = ON
    PB BUFFERS = 10
```

You can use the PREFERRED SERVER variable in the NET.CFG file (in the NetWare DOS Requester section) to connect the mobile user to the server that has this user's bindery context. If you force the user to connect to the proper server using the PREFERRED SERVER variable, the user can log in using bindery services. This enables users running the older NETX.EXE workstation client to participate in mobile computing if needed. For example, in the previous NET.CFG file, the variable is set as PREFERRED SERVER = FAC-SRV1. This enables the user JMadison to log in and access the server using bindery services if he is still running NETX.EXE.

Login Scripts for Mobile Users

The two mechanisms for creating a mobile user login script are the login script variables and an environment variable that can be called NW_SITE.

The following is an example of a container login script for mobile users. In this example, we have defined a mobile script that will allow the user to easily log into the network from any of the five major sites shown in the ACME tree. The script demonstrates how a user is mapped to the local e-mail server and the local application server. This login script is used as a container login script and also requires the NW_SITE DOS environment variable to be set on the user's workstation in the CONFIG.SYS file.

```
;*********************************************************
; MOBILE CONTAINER LOGIN SCRIPT
; for OU=FAC.OU=ADMIN.OU=RIO.O=ACME
; Creation Date: 10/8/95
; Revisions:
;*********************************************************
REM Do not execute default script
NO_DEFAULT
Write "Good %GREETING_TIME, %LOGIN_NAME"
REM Map public drive to local server
MAP S16:=SYS:\PUBLIC
REM Map F drive to the user's home server
```

```
MAP F:="HOME_DIRECTORY"
REM Map NetWare Drives according to the NW_SITE variable

IF <NW_SITE> == "NORAD" THEN BEGIN
     MAP ROOT M:= NOR-SRV1\SYS:MAIL
     MAP ROOT W:= NOR-SRV1\SYS:APPS\WP
     MAP ROOT O:= NOR-SRV1\SYS:APPS\OPRO
     END
IF <NW_SITE> == "RIO" THEN BEGIN
     MAP ROOT M:= RIO-SRV1\SYS:MAIL
     MAP ROOT W:= RIO-SRV1\SYS:APPS\WP
     MAP ROOT O:= RIO-SRV1\SYS:APPS\OPRO
     END
IF <NW_SITE> == "CAMELOT" THEN BEGIN
     MAP ROOT M:= CAM-SRV1\SYS:MAIL
     MAP ROOT W:= CAM-SRV1\SYS:APPS\WP
     MAP ROOT O:= CAM-SRV1\SYS:APPS\QPRO
     END
IF <NW_SITE> == "TOKYO" THEN BEGIN
     MAP ROOT M:= TOK-SRV1\SYS:MAIL
     MAP ROOT W:= TOK-SRV1\SYS:APPS\WP
     MAP ROOT Q:= TQK-SRV1\SYS:APPS\QPRQ
     END
IF <NW_SITE> == "SYDNEY" THEN BEGIN
     MAP ROOT M:= SYD-SRV1\SYS:MAIL
```

```
MAP ROOT W:= SYD-SRV1\SYS:APPS\WP
MAP ROOT O:= SYD-SRV1\SYS:APPS\QPRO
END
EXIT
```


NOTE

In the previous script you can specify the actual NDS volume object names instead of the server names for mapping drives. The drives M and W in our example default to searching in the bindery, which relies on SAP because we do not specify the NDS volume object names in the login script.

NetWare Login Command Switches

Table 12.1 lists the NetWare general login command switches and their syntaxes.

TABLE 12.1 *NetWare Login Command Switches*

COMMAND	COMMAND SYNTAX	PURPOSE
General Login Commands	LOGIN [File Server\] username [options]	Log clients into NetWare servers with a bindery or NDS connection.
/NS (No Script option)	LOGIN /NS	Executes a login without a login script.
/CLS (Clear Screen option)	LOGIN /CLS	Clears screen before execution of the login script.
/S (Script option)	LOGIN /S filename or LOGIN /S objectname	Executes a login script contained in the text filename or NDS object name.
/B (Bindery option)	LOGIN /B	Logs into a server using bindery services.
/TR (Tree option)	LOGIN /TR treename	Specifies a certain NDS tree by its tree name for logging in.

(continued)

TABLE 12.1
NetWare Login Command Switches (continued)

COMMAND	COMMAND SYNTAX	PURPOSE
/SWAP	LOGIN /SWAP	Swaps LOGIN.EXE into Extended RAM if you use external commands from inside the login script.
/? (help option)	LOGIN /?	Provides a help screen.

Login Script Commands

There are many login script commands that can be used in any of the login scripts. Table 12.2 lists each login script command and its function.

TABLE 12.2
Login Script Commands

COMMANDS	FUNCTION
ATTACH	The ATTACH command allows you to attach workstation to other NetWare servers. The ATTACH command only provides a bindery services connection to the selected server. This command enables you to have login scripts that are compatible and coexist with previous versions of NetWare (specifically NetWare 2 and NetWare 3). This command no longer works on the command line for NetWare 4.1 but is still provided in login scripts as a means of backward compatibility with NetWare 3 scripts that may be migrated into NetWare 4.1.
BREAK	The BREAK command has two settings, either ON or OFF. The ON setting enables you to execute your login scripts by pressing CTRL+C or CTRL+BREAK. The default setting is always OFF.
CLS	The CLS command clears the workstation screen or display. It is functionally similar to the CLS command that you can execute from DOS.

TABLE 12.2

Login Script Commands (continued)

COMMANDS	FUNCTION
COMSPEC	The COMSPEC command specifies the subdirectory where DOS should load the command line processor called COMMAND.COM.
CONTEXT	The CONTEXT command is a smaller version of the CX.EXE command line utility that enables you to set the workstation context in the NDS tree. An example of using the CONTEXT command is as follows: CONTEXT .R&D.NORAD.ACME
# CHARACTER	The # character provides external program execution. For example, to execute the capture program from within a login script type the following: #CAPTURE L=1 Q=HP4SI NB NFF TI=1 You can enter a complete pathname/filename for the external program or make sure that the proper drive mapping and search drives have been set. The # character must be the first character on the line.
DISPLAY	The DISPLAY command enables you to shows the contents of any file on the screen. The following syntax is used: DISPLAY [pathname]\filename The difference between this command and the DISPLAY command discussed later in this section is that this command shows all the characters in the file including control codes and ESC sequences.
DOS BREAK	The DOS BREAK command has two settings, either ON or OFF. The ON setting enables you to terminate any DOS program that has been executed from the login script by pressing CTRL+C or CTRL+BREAK. The difference between

(continued)

TABLE 12.2
Login Script Commands (continued)

COMMANDS	FUNCTION
	this command and the BREAK command is that this command enables CTRL+BREAK checking for DOS. The BREAK command checks only CTRL+BREAK within the login script itself.
DOS SET, TEMP SET, or SET	These commands can be used to establish the DOS environment variables for the workstation and user. The syntax is as follows: [OPTION] [DOS] SET name = "value" The [OPTION] parameter can be used or replaced with an optional keyword. The keywords are TEMP, TEMPORARY, LOCAL, which means that the variable is set only during the processing of the login script. The variable is not set in DOS.
DOS VERIFY	The DOS VERIFY command has two settings, either ON or OFF. The ON setting means that the data copied to the local drive is written without errors.
DRIVE	The DRIVE command specifies which network drive will be used as the default drive.
EXIT	The EXIT command tells the login script to terminate the processing of the login script and exit immediately. You can use the EXIT command in conjunction with the following program name: EXIT [filename] The login script passes control to the program specified. You should place the EXIT command statement as the last line in the individual login script. The EXIT command can be placed in system or container login scripts to guarantee that no individual user login scripts are executed.

TABLE 12.2
Login Script Commands (continued)

COMMANDS	FUNCTION
FDISPLAY	The FDISPLAY command shows the contents of the specified text file on the screen. The following syntax is used: FDISPLAY [pathname]\filename Only the text or characters in the file are displayed. The control characters in the file are not shown.
FIRE PHASERS	The FIRE PHASERS command produces blasts that sound like you are firing a toy weapon. You can fire 1 to 9 phasers as dictated in the following syntax: FIRE PHASERS n TIMES (where n is the number of times) You typically use this command for special effects during the login process.
GOTO	The GOTO command enables you to program the login script and jump or repeat specific locations in the login script. You can place labels in the login script to control the GOTO statements.
IF . . . THEN . . . ELSE	The IF . . . THEN . . . ELSE command enables you to build conditional logic into the login scripts. This command lets you execute certain portions of the login scripts conditionally.
INCLUDE	The INCLUDE command enables you to direct the login script to specific files that you have predefined. The following syntax is used: INCLUDE [pathname]\filename The contents of the file specified in the INCLUDE statement are the next lines processed in the script. After the file has been processed, control is returned to the statement immediately following the INCLUDE statement in your login script.

(continued)

TABLE 12.2
Login Script Commands (continued)

COMMANDS	FUNCTION
LASTLOGINTIME	The LASTLOGINTIME command checks or displays the last date and time of login.
MACHINE	The MACHINE command sets the hardware machine type. This variable receives its value from the LONG MACHINE TYPE variable in the NET.CFG. The default value is IBM_PC.
MAP and MAP DISPLAY	The MAP command is equivalent to the MAP.EXE program. The command enables you to establish network drive mapping. The MAP DISPLAY command can be set to either ON or OFF. The ON setting shows all the drive mapping during the login process. The default is ON. The OFF setting will not show the drive mapping during the login procedure.
NO_DEFAULT	The NO_DEFAULT command disables the execution of the Default Login Script, which is part of the LOGIN.EXE program. Disabling the Default Login Script can be useful when you want to control all the drive mapping for the user. This includes the drive mapping to the SYS:PUBLIC subdirectory.
PAUSE or WAIT	The PAUSE command causes the login script to stop execution until a key is pressed. This command is useful so that long messages can be read without scrolling off the screen. This command can also be helpful when debugging the login script.
PCCOMPATIBLE	The PCCOMPATIBLE command indicates that the workstation hardware is compatible with an IBM PC. For some workstations, if you do not use this command, then some NetWare utilities such as NETADMIN and FILER will not work.

TABLE 12.2
Login Script Commands (continued)

COMMANDS	FUNCTION
REMARK or REM	The REMARK command documents the lines or place comments in the login script file. The use of comments will always improve the readability and maintenance of the scripts. To place comments in the login scripts the following syntax is used: REM [text] or * text or ; text
WRITE	The WRITE command displays text messages to the workstation screen during the login process. The following syntax is used: WRITE [text] You can use the semicolon (;) to join text messages together.

Login Variables

NetWare 4.1 has always had the capability to use login variables in login scripts to help you manage the login scripts and make them more efficient and flexible. Table 12.3 lists the login script variables and their definitions that can be used to enhance your login scripts. These variables can be used in the login script to help you:

- Build conditional statements
- Provide date and time functions
- Establish DOS environment and workstation settings
- Provide NDS properties to the user

Most of the variables can be displayed using the WRITE login script command preceding the variable. There are also some examples provided in the use of these variables. In Table 12.3, note that some of the variables have underscores in them and some do not.

TABLE 12.3
Conditional Statement Login Script Variables

VARIABLE	FUNCTION
%ACCESS_SERVER	Displays or checks if the access server is functional (TRUE=functional, FALSE=not functional).
%ERROR_LEVEL	Displays or checks the DOS error level. A value of 0 indicates that no DOS errors have occurred. For example, this variable can be used to check and see if a drive mapping was successful.
%MEMBER OF "group"	Tests to see whether user is a member of the "group." Returns TRUE or FALSE.
%NOT MEMBER OF "group"	Returns TRUE if the user is NOT a member of the "group."

Table 12.4 lists the date and time login variables.

TABLE 12.4
Date and Time Login Script Variables

VARIABLE	FUNCTION
%AM_PM	Displays time as day or night, using a.m. or p.m.
%DAY	Displays the current day value ranging from 01 to 31.
%DAY_OF_WEEK	Displays the written day of the week.
%GREETING_TIME	Displays time of day as morning, afternoon, or evening.
%HOUR	Displays time of day in hours ranging from 1 to 12.
%HOUR24	Displays the hour in 24-hour time ranging from 00 to 23.
%MINUTE	Displays the minutes ranging from 00 to 59.

TABLE 12.4
Date and Time Login Script Variables (continued)

VARIABLE	FUNCTION
%MONTH	Displays month (from 01 to 12).
%MONTH_NAME	Displays name of the month.
%NDAY_OF_WEEK	Displays number of week day.
%SECOND	Displays the seconds ranging from 00 to 59.
%SHORT_YEAR	Displays year in short format (92, 93, 94, 95, 96, 97, and so on).
%YEAR	Displays year in full format (1992, 1993, and so on).

Any DOS environment variable can be used in a login script if you place angle brackets (< and >) around the variable. A common example of a DOS variable used in login scripts is:

```
<COMSPEC>
```

In order to use a DOS environment variable with login script commands, you need to add a percent sign (%) in front of the variable. For example, to map a drive to the COMSPEC DOS environment variable type the following:

```
MAP S16:=%<COMSPEC>
```

The following list of DOS Environment and Workstation variables in Table 12.5 will help to set up the workstation for the specific network users. Some variables have more than one key word, which is shown in parentheses.

TABLE 12.5
DOS Environment and Workstation Variables

VARIABLE	FUNCTION
%LAST_NAME	Displays the user's last name (surname) in NetWare Directory Services, or full login name in bindery-based NetWare. This value returns the same result as the SURNAME variable in the user properties.
%LOGIN_ALIAS_CONTEXT	Displays the context of the alias object user logged in with. This variable is valid only with NDS.

(continued)

TABLE 12.5

DOS Environment and Workstation Variables (continued)

VARIABLE	FUNCTION
%LOGIN_CONTEXT	Displays the context for the user. Returns the context where user exists in the NDS tree. This variable works only with NetWare 4.1.
%LOGIN_NAME	Displays the user's login name. This returns the same result as the CN variable in the user property list, although CN is multivalued.
%MACHINE	Displays the machine type of a workstation (IBM_PC, and so on).
%NEW_MAIL	Displays the status of the variable.
%OS	Type of operating system on the workstation (MSDOS, OS2, and so on).
%OS_VERSION	Operating system version on the workstation (3.30, and so on).
%P_STATION (PHYSICAL_STATION)	Workstation's node number shown as a 12-digit hexadecimal.
%PASSWORD_EXPIRES	Displays the number of days before the user password will expire.
%REQUESTER_VERSION (optional names: NETWARE_REQUESTER, REQUESTER)	Displays the version of the VLM requester.
%REQUESTER_CONTEXT	Displays the context that is found in the workstation's NET.CFG file at the time of login.
%SHELL_TYPE (SHELL_VERSION)	Version of the workstation's DOS shell (1.02, and so on); supports NetWare 2 and 3 shells and NetWare 4 Requester for DOS.
%STATION (CONNECTION)	Displays the workstation address for that user.

NetWare 4.1 extends the list of login variables, as shown in Table 12.6, through the use of the user properties found in NetWare Directory Services. If the property includes a space, enclose the name in quotation marks or replace the spaces with an underscore.

TABLE 12.6

NDS User Properties Variables

VARIABLE	FUNCTION
%ACCOUNT_BALANCE	Displays account balance information if being used.
%ALLOW UNLIMITED CREDIT	Displays whether unlimited credit has been assigned for that user. The value returned is "Y" or "N".
%BACKLINK	Established for any user object in which there is an associated external reference on a different server.
%BINDERY PROPERTY	Used to emulate the bindery properties that are not represented by the other user properties.
%CN	Displays the login name of the user who logs in to the network.
%DESCRIPTION	Displays any value contained in the description property for the user.
%EMAIL ADDRESS	Displays the first value in the e-mail address property for the user.
%EQUIVALENT TO ME	Displays only the first value in the list.
%FACSIMILE TELEPHONE NUMBER	Displays the first number in the fax number property for the user.
%FULL NAME	Displays the user's full name value. This value is the property stored in NDS and the bindery if the server is bindery-based NetWare. Spaces are replaced with underscores.
%GROUP MEMBERSHIP	Displays the values of the membership attributes.
%HOME DIRECTORY	Displays the complete path for the home directory property set for the user who has logged in.
%INITIALS	Displays the value of the user's middle initial property.
%LANGUAGE	Displays the current language being used by the user.

(continued)

TABLE 12.6

NDS User Properties Variables (continued)

VARIABLE	FUNCTION
%L	Displays the first value of the location property for the user.
%LOCKED BY INTRUDER	Displays status of locked by intruder property. The value returned is "Y" for yes or "N" for no.
%LOGIN DISABLED	Displays account disable status. The value returned is "Y" for yes or "N" for no.
%LOGIN GRACE LIMIT	Displays the value of the login grace limit property.
%LOGIN GRACE REMAINING	Displays the number of remaining grace logins for the user.
%LOGIN INTRUDER ATTEMPTS	Displays the number of incorrect login attempts for the user.
%LOGIN MAXIMUM SIMULTANEOUS	Displays the value of the maximum simultaneous connections for the user.
%LOGIN TIME	Displays both the date and time of the login time for the user.
%MAILBOX ID	Displays the mailbox ID for the user.
%MAILBOX LOCATION	Displays the mailbox location for the user. MHS can, but does not have to, be installed in order to have a value for the mailbox location.
%MESSAGE SERVER	Displays the default server or message server name.
%MINIMUM ACCOUNT BALANCE	Displays the value of the minimum account balance or low balance limit.
%NETWORK ADDRESS	Displays the physical network address, node, and socket number for the workstation.

TABLE 12.6

NDS User Properties Variables (continued)

VARIABLE	FUNCTION
%OBJECT CLASS	Displays the base class for the user object.
%OU	Shows the first value defined in the Department list for the user.
%PASSWORD ALLOW CHANGE	Shows the value of this user property "Y" or "N".
%PASSWORD EXPIRATION INTERVAL	Displays time in total seconds before the user password will expire.
%PASSWORD MINIMUM LENGTH	Displays the minimum password length setting for the user.
%PASSWORD REQUIRED	Displays the value of the password required. Displays or returns "Y" or "N".
%PASSWORD UNIQUE REQUIRED	Displays the property value of unique password required. Displays or returns "Y" or "N".
%PHYSICAL DELIVERY OFFICE NAME	Displays the value of the city property for the user.
%POSTAL OFFICE BOX	Displays the user's postal office box value if any.
%POSTAL CODE	Displays the value of the user's postal zip code, if any.
%POSTAL ADDRESS	Displays the value of the users's postal address property, if any.
%PROFILE	Displays the name of the profile object if the user is associated with a profile.
%REVISION	Displays the value of the revision property for the user. The revision increments each time the user is accessed.
%S	Displays the value of the state or province property for the user.

(continued)

TABLE 12.6

NDS User Properties Variables (continued)

VARIABLE	FUNCTION
%SA	Displays the value of the street address for the user.
%SECURITY EQUALS	Displays security equivalence assignments made for that user. Only displays the first value in the list.
%SEE ALSO	Displays the first value in the see also property for the user.
%SERVER HOLDS	Displays the number of accounting charges pending while the server performs a chargeable action.
%SURNAME (LAST_NAME)	Displays the user's surname property value, if any. User's last name (surname) in NetWare Directory Services, or full login name in bindery-based NetWare.
%TELEPHONE NUMBER	Shows the user's phone entered in his phone number property. Only displays the first value in the list.
%TITLE	Displays the title for the user if one has been entered as a user property. Only shows the first value in the list.
%UID	Displays a unique user ID assigned to the user for use by UNIX clients.

CHAPTER 13

Managing NetWare Security

"Even in the common affairs of life, in love, friendship, and marriage, how little security have we when we trust our happiness in the hands of others!" William Hazlitt

As networks become increasingly more distributed the exposure of information has also increased. Therefore, it makes sense that in large networked environments, you'll have to pay more attention to security and take measures to ensure that your data is protected.

Securing your network actually encompasses many areas of security including physical access, login, NDS, and file system restrictions. Other areas that can be classified within the physical access category include natural disasters and hardware failures. Another area of security violations is caused unintentionally by users who have authorized access and simply make mistakes. Computer viruses also pose a threat to security and can be caused intentionally or unintentionally.

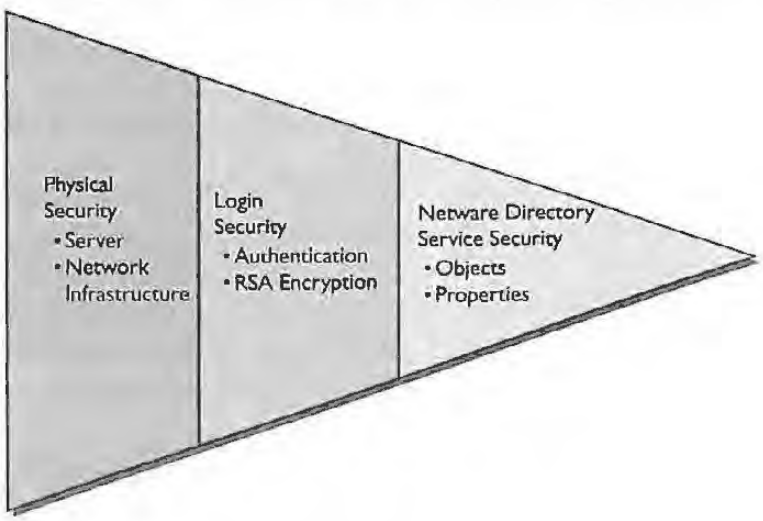
While each of these areas is important, the goal is to secure your important data from theft, eavesdropping, or destruction. Every organization may place greater emphasis on a particular aspect of security depending on the sensitivity of the data. Some organizations may attempt to enforce all available aspects of security, while other companies may only concentrate on a few areas. As a network administrator you must decide along with your other managers what is an acceptable level of security versus risk for your department or company's data. Any level of security you implement is more than what your users will want because of additional burdens placed on their ability to freely work on the network.

Some companies may be especially prone to disasters because of their geographical location and natural weather conditions. Other companies may be at a greater hardware risk because they are running their network in a factory or plant where the possibility of fire or other damage is greater.

This chapter discusses these topics including securing the physical access to hardware, understanding the authentication and login process, and applying NDS restrictions. NDS restrictions include security for objects and properties as well as for the file system. These various levels of security are represented in Figure 13.1.

FIGURE 13.1

Security can be broken down into the categories of physical security, login security, and NDS security.



Designing a Security Model

One of the first steps to implementing security for your environment is to design what is commonly referred to as a security model. A security model is basically your own company's road map for implementing security. Consider the following questions when implementing your security plan:

- What security threats do you have at your company?

  You must evaluate all aspects of threats including your building's age and entrance policies, the geographic location, the experience level of your users, the number of users, the age of your server hardware, and access points to your network from locations outside your company. Access points refer to gateways or other routing devices connected to the Internet or other wide-area links.

- What security do you currently usually have in place and what improvements can be made?

  Most companies usually have some form of network security in place. If your network is growing or if you have recently connected to the Internet, you

should consider increasing your security. If your company is in the process of relocating to a new building you should reevaluate your security measures.

- What are the costs associated with increasing your security?

  You have to take into account what the costs will be when increasing your security. Will your added security measures require more staff to assist with this aspect? Will the changes require the purchase of newer, more expensive hardware such as tape backups and virus protection software? How much more administration time is needed to service each user? What is the cost of your downtime if your network should have a failure?

- Are you reviewing your security processes on a continual basis?

  Each network administrator should perform a regular security check on the network to determine if new workstations or any other new devices have been added to the network. A review will also determine if any security policies need to be modified based on any other changes that may have occurred on the network.

Controlling Physical Access to Your Hardware

NETWORK HARDWARE SECURITY

One of the most basic aspects of providing a secure network is to physically secure your servers and workstations. If possible, your servers should be protected in a locked room such as a data center or wiring closet. Many large companies already place servers in their data centers. Smaller companies may not have this option, but there are other steps that can be taken to help safeguard your servers as explained at the end of this section.

Network administrators can to secure their physical networks by encasing their network wiring in conduit to prohibit intrusion. Again, your company and administrators must determine if such an enormous expense is needed to secure your particular data.

Most network administrators going to this length are probably considering installing a network that is C2 compliant. For more information on a trusted environment, you can refer to Novell's Application Notes April 1994 Special Edition "Building and Auditing a Trusted Environment with NetWare 4" and August 1994 "An Introduction to Novell's Open Security Architecture." At the time of this writing NetWare 4.1 was nearing C2 compliance for NetWare 4.1 servers and workstations running NetWare client software. For information on security and related topics you can also refer to the National Computer Security Center Document NCSC-TG-028, which is known as the "Orange Book."

Network administrators may also want to consider enabling auditing on various activities for servers that are designated as highly secure. For more information on the auditing functions using the AUDITCON utility, see Chapter 2. There are also additional third-party products that monitor network activity as well.

You also should consider taking the following steps to secure your servers:

1. Keep all servers in a data center or an air-conditioned locked room and limit access to these areas.
2. Provide a tape backup for all servers on your network to back up files as well as NDS. Restore data to a test directory to periodically check that the restore procedure is working properly.
3. Remove DOS from the file server by typing REMOVE DOS at the server console.
4. Use an auditing tool such as Novell's AUDITCON or another third-party utility to track events on highly sensitive servers.
5. Lock the file server console using MONITOR.NLM and do not store the password in the AUTOEXEC.NCF file.
6. Require passwords for print servers that need to log in to the NetWare 4.1 servers.
7. Periodically check containers that are under your responsibility for new objects. Investigate all new objects added to your containers.

8 • Carefully control the use of RCONSOLE and its password. You can encrypt the RCONSOLE password for your AUTOEXEC.NCF by following these steps:

a • At a NetWare 4.1 console type REMOTE ENCRYPT.

b • You will then be prompted for an RCONSOLE password to be encrypted.

c • After entering the password, the file LDREMOTE.NCF will be created.

d • You can call this file in your AUTOEXEC.NCF or copy the contents out of the file into your AUTOEXEC.NCF.

CONTROLLING PHYSICAL ACCESS TO WORKSTATIONS

Workstations are more problematic to secure because some of your employees may use notebook computers that are transported away from the office each day. Also, very few companies ever go to the bother of physically securing workstations to desks or tables. Even if your users have stationary computers, there is always the possibility that unauthorized individuals will have access to your data.

You may also want to consider using a protocol analyzer such as Novell's LANalyzer, which provides a feature known as "new station" alarm to notify network administrators when a new node address has been discovered on your network. This application could be run on a nightly basis to search for any new hardware that may have been added to your network. It is usually quite easy for anyone to attach a notebook computer to your network infrastructure.

You can also limit access to your servers through Novell's station restriction and time restrictions procedures, which can be accomplished with the NWADMIN or NETADMIN utility. These procedures limit a user to a particular workstation for logging in and also to specific times of the day.

You should consider taking the following steps to secure your workstations:

1 • Encourage users not to leave their workstations unattended while logged in to the network unless they are using password protected screen saver software.

2 • Always log out of the network before leaving your work location for the day.

3 • Keep office doors locked when not in use and workstations powered off if possible.

4 • Administrators especially should not leave their workstations unattended while logged in with supervisor rights to the tree, its containers, or servers.

5 • At a minimum you should have an asset tag for each piece of hardware on your network in case of theft or other damage.

6 • Use an antivirus product for your network that is loaded from the container login script.

7 • Carefully monitor the use of the ADMIN or any tree administrator user object password. Change the password frequently.

8 • Require periodic changes in users' passwords. Do not allow them to use the same password twice.

9 • Have a company policy that no external diskettes are to be brought into the workplace unless they are scanned for viruses first.

10 • Dial in access should be closely monitored and use automatic call back features built into remote software.

Understanding the Login and Authentication Process

After you have physically secured your servers, the NetWare 4.1 login security is the next line of defense in network security. The authentication procedure verifies that any requests the server receives are from legitimate clients. The authentication process consists of the login and the authentication. NetWare 4.1 uses its own authentication process that is compatible with NetWare 3 as well. The mechanism used to make the

authentication process extremely secure is known as encryption and is discussed later in this section.

PASSWORD SECURITY AND VERIFICATION

The purpose in having a password is to prevent unauthorized access to your network resources. NetWare 4.1 security must be reinforced by all network users and administrators practicing good password security. Any disclosure of passwords will allow a user to access that account to the extent of that user's rights. Therefore, you can take numerous precautions to secure your network, but the human factor is always the biggest threat to your security. You can take some steps to help minimize the threat of a password breach in your security:

1. Without exception, require a user password for all users.
2. Always require periodic changes of passwords for mobile users dialing into your network or users who work from home.
3. Users should never post their passwords anywhere.
4. Users should be encouraged to avoid easy passwords such as family member names, and so on.

Through the NetWare 4.1 utilities you can do the following:

1. Require a minimum password length of five to eight characters.
2. Enforce periodic changes of passwords.
3. Enforce unique passwords.
4. Frequently change the password for your ADMIN or tree administrator user objects.

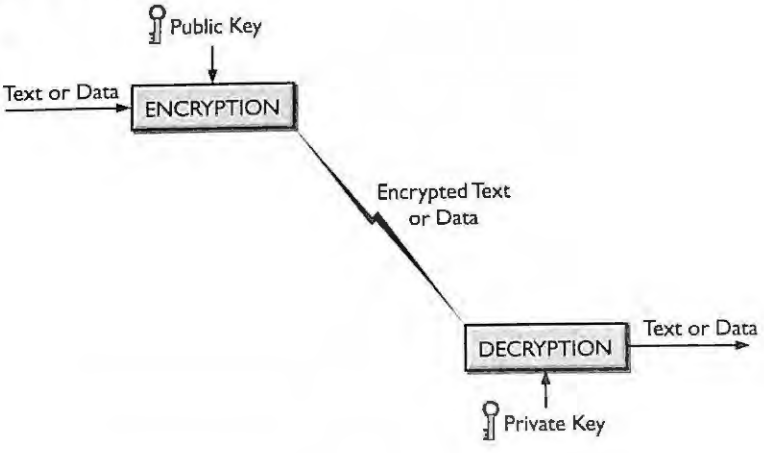
RSA ENCRYPTION/DECRYPTION

NDS uses encryption to secure authentication information that is being sent across the network from one server to other servers or from a workstation to a server for authentication purposes. NetWare 4.1 uses the RSA encryption technology to provide public and private key encryption. The encryption process produces transmissions across the network that are unreadable except to the receiving entity.

The public and private key encryption is also known as asymmetrical cryptography because there is a mathematical relationship between the two keys. If data is encrypted with the object's public key, the receiving object will use its private key to decrypt the information. The public key can be read by any requesting object. The receiving object holds the private key but never discloses it. Figure 13.2 shows the relationship between the public and private keys during transmission of authentication data. Keep in mind that only the server objects keep their own private key. Users obtain their encrypted private key during the login process. This key is then decrypted, used to generate a signature, and then discarded.

FIGURE 13.2

The use of public and private key encryption of authentication data



Directory Services uses encryption to ensure that authentication information is secure while it is being transmitted on the wire. There will always be a public key/private key pair for each encrypted transaction. To prevent intrusion during a transmission by capturing and replaying encrypted messages, the NetWare authentication process uses what is known as a nonce value. A nonce value is a random value that is generated for each encrypted transaction. The nonce value is associated only once with each encrypted

message. Because the nonce is used only once, it will not do an intruder any good to capture an encrypted message and attempt to impersonate the sender.

Each transaction uses the network layer infrastructure to send and receive packets. This means that for authentication IPX and NCP requests are made between a client and a server, but for NDS authentication the data is encrypted before transmission on the wire. Keep in mind that the term "client" can mean a Directory Services server communicating with another server. As discussed previously in Chapter 12, there are varying degrees of authentication access. We list them here again briefly.

Connected but Not Logged In

This state is a client who has attached to a NetWare 4.1 server either through the NETX shell or the VLM client. It could also be a DS server that does not use NETX or VLMs. A connected but not logged in state can exist for either NetWare 3 or NetWare 4.1 users to the first attached server or, if a connection is made for tree walking, after the first attached server. This state is seen in the MONITOR utility as NOT LOGGED IN and does not take a licensed connection.

Authenticated

Authentication is a process of proving identity to a server. In NetWare 3, this meant logging in. In NetWare 4.1, it happens as a "behind the scenes task" at the client and is called authentication.

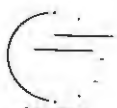
This type of connection indicates that a NetWare 4.1 server has established a user's identity after the client has entered a correct name and password to obtain the encrypted private key. Authentication occurs for both NetWare 3 and NetWare 4.1 users, with NetWare 4.1 adding more security to this process. The authentication process includes creating a proof and a server verifying that proof. Proof is constructed with the client's public key, signature, and credential and is built at the time of login.

Licensed

A connection is said to be licensed when a client has made a request of the server such as mapping a drive or capturing to a printer. At that time the client requests the server to license the connection. This will cause the license count on the server to be decremented by one. Only an authenticated connection can be licensed.

A combination of these states determines what level a user currently has in a NetWare 4.1 environment. For example, when a connection is neither authenticated nor licensed, users can navigate the NDS tree through the use of the CX (Change conteXt) program (presuming [PUBLIC] has the Browse right at the [ROOT] object of the tree) and the execution of the LOGIN.EXE utility as well. They have attached to a server, but have not yet authenticated.

If a user is licensed and authenticated, he can access NDS and file system information to the extent allowed by his rights.


NOTE

The Change Connection State NCP call switches the connection between Not Licensed and Licensed.

THE USER LOGIN

The first step to authenticating in a NetWare 4.1 environment is the identification phase, also known as the login phase. When the client first logs in to a NetWare 4.1 server he must establish his identity with the server and then proceed through the authentication phase. This is accomplished by the client broadcasting a Service Request, with the broadcast type 0x0278 if the preferred tree is set on, or the broadcast type 0x0004 if the preferred server is set. A server or router that receives this request will reply with a response. The client will examine the response and accept the first response with the correct tree/server being sought. This gets the client a connection to begin the tree walking (resolve name operation). NDS will then search the tree to find a writeable replica of the user's object. The following steps then occur during login and authentication phases:

Login Phase

1. Once a writeable replica is found with the user object, the user is prompted for a name and password.
2. After successfully proving knowledge of the password, the client receives the encrypted private key from the server.
3. The private key of the client is used to generate a signature.
4. The public key of the client is used to generate a credential.

5 • The signature and credential are used to build a proof that is used later in this sequence.

Authentication Phase

6 • The public key attribute of the server is read by the requesting client.

7 • The proof previously generated is encrypted with the server's public key and sent to the server.

8 • The server decrypts the proof with the server's private key.

9 • The proof is verified by the server through a mathematical computation of the client's public key and the client key stored on the server.

10 • If the proof is correct, then authentication is successful.

11 • The finished authentication request will then call Directory Services for the user's security equivalence vector.

12 • The login process is then passed back to Directory Services to execute the applicable login script and apply NDS access controls.


NOTE

For more information on user login and authentication, refer to Novell's October 1994 Application Notes "Identification and Authentication in NetWare 4."

Keep in mind that the client's password is never transmitted across the wire. Therefore, it is not possible for someone to capture a password packet on the wire. In addition, the authentication data is valid only during the current login session. If a user terminates the session and then reconnects, the authentication process is repeated.


NOTE

Another feature, known as Packet Signature, requires each packet to have a valid signature in order to be executed by the server. Packet signing makes it far more difficult for someone to forge NCP packets and send them on to the server for processing.

BACKGROUND AUTHENTICATION

During the initial login process a NetWare client will identify a hosting server. Although the login process in NetWare 4.1 is performed only once, the authentication process may occur throughout the entire session in order to enable services from other servers.

Background authentication is the ongoing identification process that occurs after the initial login. If a connection needs to be made to other network services, authentication is done through the process of background authentication. Background authentication can occur because the NetWare 4.1 servers can verify the proof provided by the client from the client's public key without additional user intervention. A server making a request of another server is also considered a client. Keep in mind that the proof is constructed at the client using the signature, credential, and public key.

When a user logs out of the network, the NetWare 4.1 license manager is notified and makes the license available to any other validated user needing a license. The logout includes destroying the service connection to all but one server along with all bindings that were part of the user connection. The authentication data is also destroyed on the workstation at logout.

NDS Access Control

NDS security actually consists of two parts known as file system security and object security. Both aspects of NetWare 4.1 security work together to provide a flexible and effective method for controlling access to your network. The file system security provides access control to files and directories. The object security provides access control to NDS objects and associated operations. You must determine to what extent you want to enable the many file system and NDS security features at your disposal. NetWare is well regarded for providing a high degree of network security, and much of the security administration happens by default as explained later in this chapter.

The first step to understanding NetWare security is to begin with the file system security. The file system security consists of the security that was introduced in previous versions of NetWare. Your familiarity with the security concepts in NetWare 3 will be a great help to you in understanding security in NetWare 4.1.

UNDERSTANDING FILE SYSTEM SECURITY

Very little has changed in file system security from NetWare 3. All the rules governing rights administration are the same in NetWare 4.1. As an administrator you don't have to learn any new concepts to manage your NetWare 4.1 files. However, if you are new to NetWare you will want to read this section and refer to Novell's documentation on managing the file system. NetWare 4.1 does introduce some additional file system attributes that may be useful for particular situations.

File system security basically consists of assigning trustee rights and file/directory attributes. The trustee rights assignments can be applied to any NDS object including containers, user objects, group objects, and organizational roles. Table 13.1 shows the file system rights available in NetWare 4.1.

TABLE 13.1
File System Rights

RIGHT	DEFINITION
Access Control	Adds/modifies rights to files and directories
Supervisor	Enables all file and directory assignments to be made and grants all rights listed in this table
Read	Enables the trustee to open, read, and execute application files
Write	Enables the trustee user to open, write to, and modify a file
Create	Enables the trustee to create subdirectories and files
Erase	Enables the trustee to delete directories and files
Modify	Enables the trustee to modify, rename directories and files, and change file attributes
File Scan	Enables the trustee to view file and directory names in the file system

File Attributes

NetWare 4.1 file system security includes the capability to manage access at file and directory levels just as it did in previous versions of NetWare. Attributes control what actions can or cannot be taken on a file or directory. For certain files, such as application files on the network, you may want to make sure they are flagged as Read Only and Shareable so that no unintentional or intentional deletions occur.

Additional file and directory attributes have been added to the NetWare 4 file system to provide more functionality to NetWare 4.1. These new file system attributes are listed below in Table 13.2. For a complete list of all attributes associated with NetWare 4.1 files and directories, refer to Novell's documentation.

TABLE 13.2

New file system attributes that have been added to NetWare 4

ATTRIBUTE	ABBREVIATION	DEFINITION
Compress	Co	Status attribute that indicates the file is compressed.
Can't Compress	Cc	Status attribute that indicates the file cannot be compressed because of limited space savings.
Don't Compress	Dc	Added to a directory, this attribute keeps all files within the directory from being compressed. This attribute can also be added to a specific file.
Immediate Compress	Ic	Added to directories or files, this attribute alerts the file system to compress a file as soon as the operating system can handle the action.
Migrated	M	This status attribute indicates that the file has been migrated.
Don't Migrate	Dm	Added to a directory, this attribute will not allow files within the directory to be migrated to secondary storage. This attribute can also be added to a specific file.

For more information on administering file system security, refer to Novell's NetWare 4.1 manual Supervising the Network.

UNDERSTANDING OBJECT SECURITY

Your understanding of NetWare 3 file system security will assist you in mastering NetWare 4.1 because both use the same terminology and the same rules. The rules for file system security in both versions of NetWare are identical. NetWare 4.1 extends security to the NDS environment by adding access controls to all objects and properties found in your tree.

As shown below in Figure 13.3, security features in the NetWare 4.1 environment are similar to that of NetWare 3.

FIGURE 13.3

Security features between the two versions of NetWare are similar.

NETWARE 3	NETWARE 4.1
BINDERY	NETWARE DIRECTORY SERVICES
FILE & DIRECTORY RIGHTS FILE ATTRIBUTES	FILE & DIRECTORY RIGHTS FILE ATTRIBUTES
SUPERVISOR OPERATOR OBJECT RIGHTS ONLY	OBJECT & PROPERTY RIGHTS
SUPERVISOR USER	ADMIN USER
GROUP EVERYONE	O=ORGANIZATION
GUEST USER	[PUBLIC] TRUSTEE
INHERIT RIGHTS MASK (IRM)	INHERITED RIGHTS FILTER (IRF)
DIRECTORY ENTRY TABLE (DET)	DET, ACL (ACCESS CONTROL LIST)

The exceptions between NetWare 3 and NetWare 4.1 are as follows:

- **Inherited Rights Filter (IRF)** terminology is used instead of Inherited Rights Mask (IRM). In NetWare 4.1, the term IRF describes the operation of filtering out rights for a particular object. IRFs are explained in detail below.
- **NetWare Object classes** have been expanded from four in NetWare 3 to approximately 32 in NetWare 4.1.
- **The SUPERVISOR user** is used only for bindery requests in NetWare 4.1.

Another NDS user object (typically called ADMIN) is granted Supervisor object rights at the [ROOT] of the tree. This user object is the functional equivalent of the Supervisor found in NetWare 3 and has default Supervisor rights over all NDS objects in the tree as well as the file system for any NetWare 4.1 server installed. One other major difference to note is that you cannot filter Supervisor rights in the NetWare 4.1 file system, which is just like NetWare 3. You can, however, filter Supervisor rights in the NetWare Directory Services tree.

- **Guest** is not automatically created in NetWare 4.1. The [PUBLIC] Trustee is similar to Guest by enabling users to see the NDS tree before logging in to a server. However, Guest is a real object and [PUBLIC] is not, so the similarities are few.

- **Group Everyone** is not automatically created in NetWare 4.1. However, an equivalent feature is available by using either the [ROOT] object or O=Organization object. For example, O=ACME includes every object in the NDS tree and rights can be assigned to O=ACME, which all users receive automatically.

- **Operators** for print queues found in NetWare 3 are now an attribute of the Queue object found in NetWare 4.1 Directory Services.

- **Directory Entry Tables** are still used to store file system trustees in NetWare 4.1. In addition, NetWare 4.1 uses the Access Control List (ACL) to store NDS trustee information.

Object Rights Defined

Object rights are simply rights granted to a particular object to access or manage another object. In NetWare 3 servers the Supervisor object has rights to manage all other bindery objects on the server. NetWare 4.1 has expanded on this concept by allowing all NDS objects rights to other objects. As shown in Table 13.3, NDS objects can receive many different rights.

TABLE 13.3

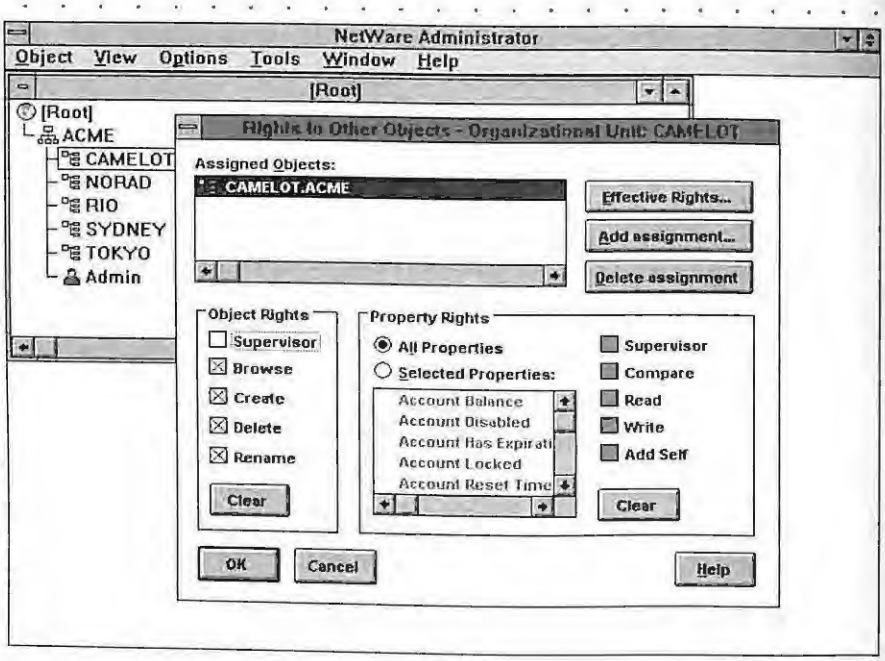
An object can receive many different object level rights to manage other objects in the Directory.

NDS OBJECT	OBJECT LEVEL RIGHTS
Supervisor (S)	Grants full privileges to the trustee over an object and has complete access to all the object's property rights.
Browse (B)	Enables a trustee to see an NDS object in the tree during a browse of the tree.
Create (C)	Enables a trustee to create objects below this object (applies to container objects only).
Delete (D)	Enables a trustee to delete an object. Subordinate objects must be deleted first if you are deleting a container.
Rename (R)	Enables a trustee to rename an object.

An example of how object rights appear in the NWADMIN utility is shown in Figure 13.4 below.

FIGURE 13.4

An example of object rights as they are displayed in the NWADMIN utility



Property Rights Defined

Property rights enable a trustee to view or change the values of a particular object's properties. You can have rights to certain properties (selected property rights) or to all

properties (all property rights) for a particular object. For example, the Supervisor right over an NDS object also grants Supervisor privileges for all properties in that object. All other rights assignments made at the object level are not affected on the properties. In fact, Supervisor rights at the property level do not grant Supervisor rights at the object level. Only the reverse is true. Table 13.4 shows a list of the available property rights in NetWare 4.1.

TABLE 13.4
NetWare 4.1 property rights

PROPERTY RIGHT	FUNCTION
Supervisor (S)	Grants all rights to the object's properties.
Compare (C)	Enables a test for a value match and returns a true or false. Compare is a subset of read. If you have read/writes you automatically have Compare rights at the property level.
Read (R)	Returns a value (contents) of a property. Read contains the Compare right.
Write (W)	Enables you to modify, add, change, and delete a property value.
Add/Remove Self (A)	Enables you to add or remove yourself as a value of a property. It is a subset of the write right. If you have write rights to a property you automatically have the Add/Remove Self right.

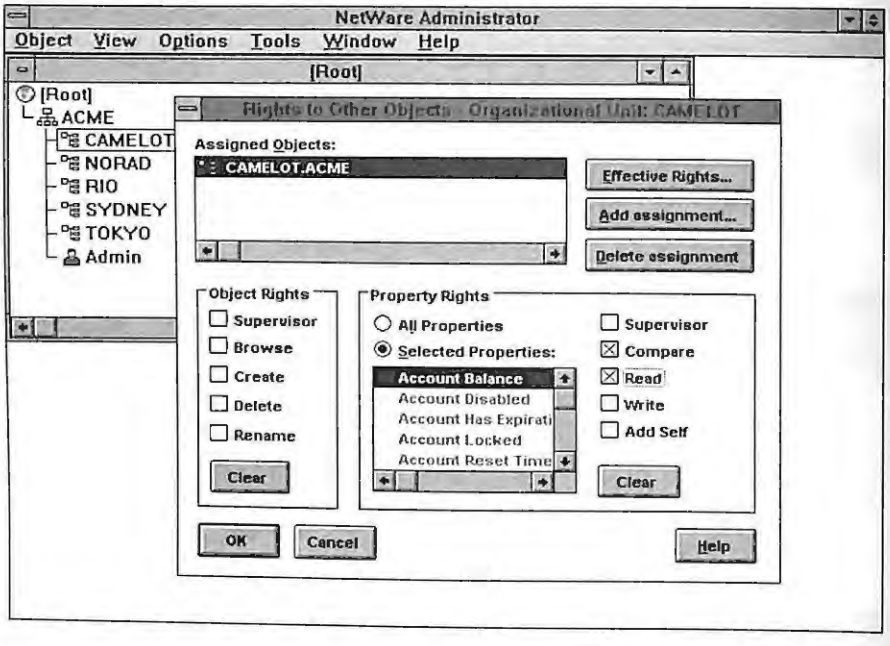
An example of how property rights are displayed in NWADMIN is shown in Figure 13.5 below.

The Access Control List

The Access Control List (ACL) is a special property of every object. It can actually be considered the most important mechanism for determining NDS access. The Access Control List contains trustee assignments for an object and its properties. A user object, for example, with the write right to the ACL of another user object has what is known as managed rights over that user object. This means that an object with the write right of any object's ACL can make any rights assignments to that object.

FIGURE 13.5

Property rights in the NWADMIN utility



Each object can have an ACL. The ACL is a property value and contains three entries: the trustee ID, the type of access (object or property), and the actual rights assignment. This concept is shown in Figure 13.6.

FIGURE 13.6

Every object may have an Access Control List as one of its properties.

ACCESS CONTROL LIST (ACL)

TRUSTEE OBJECT ID	TYPE OF ACCESS (OBJECT OR PROPERTY)	RIGHTS ASSIGNMENT
07721429	OBJECT	[S]
07618424	OBJECT	[CR]
01099600	PROPERTY	{S }

The ACL is extremely powerful, and its access should be closely controlled for every object on your network. Because the ACL is a property of an object, it can be modified by anyone who has write rights to the ACL property for the object. This means that someone with the write rights to the ACL can give rights assignments for that object.

By default, users do not receive write rights to their own ACL. In the NDS schema, some of the object classes specify a default ACL template. The default ACL template grants basic access control to newly created objects. If the object (for example, the organizational unit object) contains a default ACL template when created, it would have information in its ACL as shown in Table 13.5 below.

TABLE 13.5
Container object default ACL

OBJECT NAME	DEFAULT RIGHTS	AFFECTED ATTRIBUTES (PROPERTIES)
[ROOT]	Read	Login script and print job configuration

When a container object is created, the [ROOT] object automatically obtains the Read property right to the container's login script and print job configuration.

The user class is another example of an object that receives a default ACL during initial creation. Its default ACL would appear as shown in Table 13.6 below.

TABLE 13.6
User object default ACL

OBJECT NAME	DEFAULT RIGHTS	AFFECTED ATTRIBUTES (PROPERTIES)
[PUBLIC]	Read	Message server
[ROOT]	Browse	[Object rights]
[ROOT]	Read	Group membership
[ROOT]	Read	Network address
[self]	Read	All attributes
[self]	Read/Write	Login script
[self]	Read/Write	Print job configuration

Therefore, as was mentioned earlier, access control is limited, and your default security is a closed-door approach. As a NetWare administrator, you must open up security doors (granting additional rights to administrators) only when necessary. Otherwise, the majority of your users receive by default sufficient access rights when they are created.

As we have stated earlier, an object may or may not have an ACL. If an object already has the effective right, the ACL is not created. For example, all objects have the default ACL that specifies that the ADMIN (or appropriate object) has object Supervisor rights to the user object. If the ADMIN already has the Supervisor object right (and it does if it creates a user object), the default ACL is not created when the NDS user object is created.

Learning the Rules of NDS Security

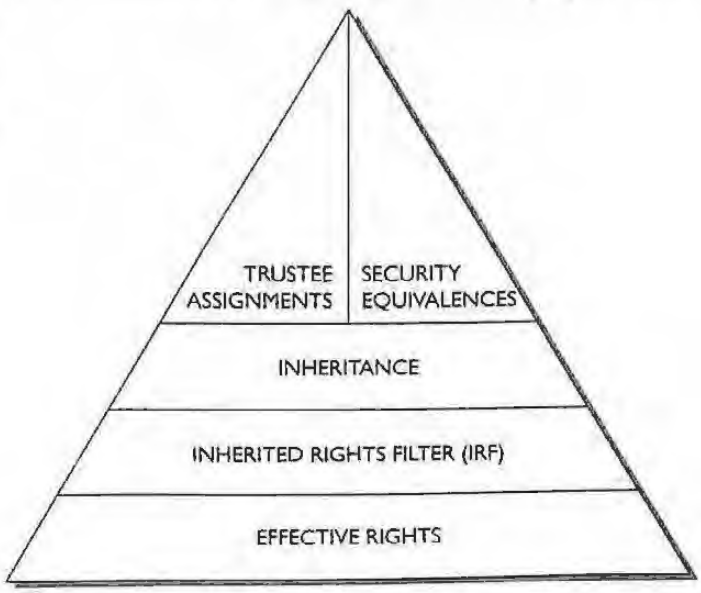
The first step in understanding NDS security is to understand the rules that govern it. This section will outline the concepts and rules for each area of security. With the groundwork in place we will then focus on some specific examples and explain how security is implemented for each.

NDS security uses the same terminology as file system security. In fact, your familiarity with NetWare 3 file system security will provide you with a great foundation for understanding NetWare 4.1 security. The following concepts will be discussed and are shown in Figure 13.7:

- Trustee Assignments
- Security Equivalence
- Inheritance
- Inherited Rights Filter
- Effective Rights

FIGURE 13.7

The NDS security pyramid serves as a visual basis for understanding the order and concepts of NDS security.



TRUSTEE ASSIGNMENTS

A trustee assignment indicates the rights granted to an object for a specific file, directory, object, or property. An object that has been granted rights to manage another object is said to be a trustee of that object. A trustee assignment is a direct, explicit assignment of rights to a particular object. Sometimes you will hear the term explicit trustee assignment, which means the same thing. A trustee assignment is listed first in our pyramid diagram because it is the first point at which rights assignments are made. It is the basis for all subsequent security assignments, such as security equivalence and inheritance. Security always begins with a trustee assignment. For example, the use of a group object requires you to grant the group a trustee assignment that the members of the group receive through security equivalence.

The installation of NetWare 4.1, as another example, causes some default trustee assignments (templates) to be made for user and server objects. As an administrator you will most likely make additional trustee assignments for groups, containers, and other administrators as explained in the following section.

Default Trustee Assignments for Users

During the installation of your first NetWare 4.1 server the ADMIN user object (if you've named it that) receives an explicit assignment of object Supervisor at object

[ROOT] as shown in Figure 13.8 This assignment is the first trustee assignment made by the NetWare 4.1 installation software and initially is the only object in the tree with object Supervisor rights at [ROOT].

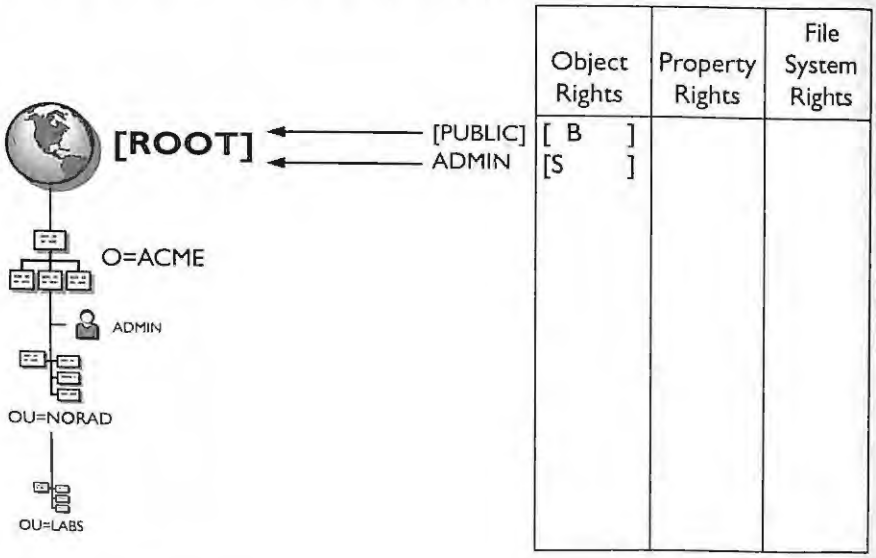
NOTE

The ADMIN object is just a user object like any other object, but it has been granted object Supervisor rights at [ROOT]. It can be renamed, deleted, and moved like any other user object.

Also, during installation of NetWare 4.1, [PUBLIC] receives an object trustee assignment of object Browse at object [ROOT]. With this right all users can browse the tree after attaching to a server before logging in. This enables users to use the CX command to browse the tree and discover object names once they have loaded the NetWare client software and have attached to a NetWare 4.1 server. Figure 13.8 shows a graphic representation of the PUBLIC rights received at the time of installation of NetWare 4.1.

FIGURE 13.8

This diagram shows the default trustee assignments made at the installation of NetWare 4.1.



NetWare 4.1 Server Default Trustee Assignments

The NetWare 4.1 installation utility also makes trustee assignments at the file system level. The ADMIN object has object Supervisor rights to the tree. The server object is in the tree and therefore has rights to your servers. Because the ADMIN object has object

Supervisor rights to the server object, the ADMIN object also receives Supervisor rights to the NetWare file system of that server. This is the only instance in NetWare 4.1 security where object rights have an impact on file system rights. In fact, any object with write rights on a server's ACL has Supervisor rights on the file system of that server.

[PUBLIC] receives the Read property right to the server object's Messaging Server property right so that if the server is being used for the default login server, its property can be located. This assignment is in the template for the server object.

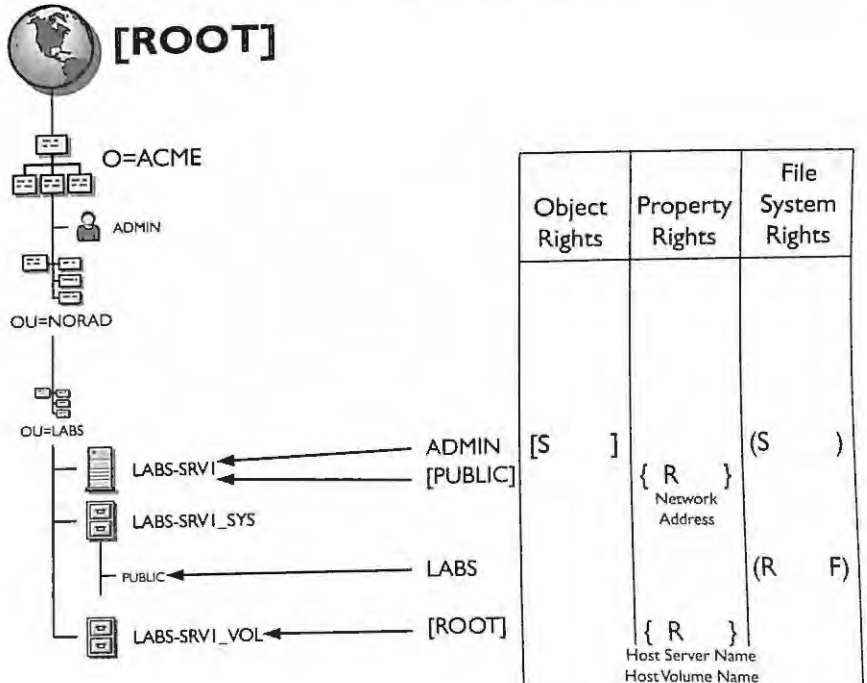
The container object that the new server is installed into receives Read and File Scan rights to the server's \PUBLIC directory. This default access enables all users in the container to execute any files stored on the server's \PUBLIC directory.

The container also receives Create rights to the server's \MAIL directory.

Figure 13.9 shows how these trustee assignments are made when a server object is first created.

FIGURE 13.9

This example shows how trustee assignments are made for a server object during installation of NetWare 4.1.



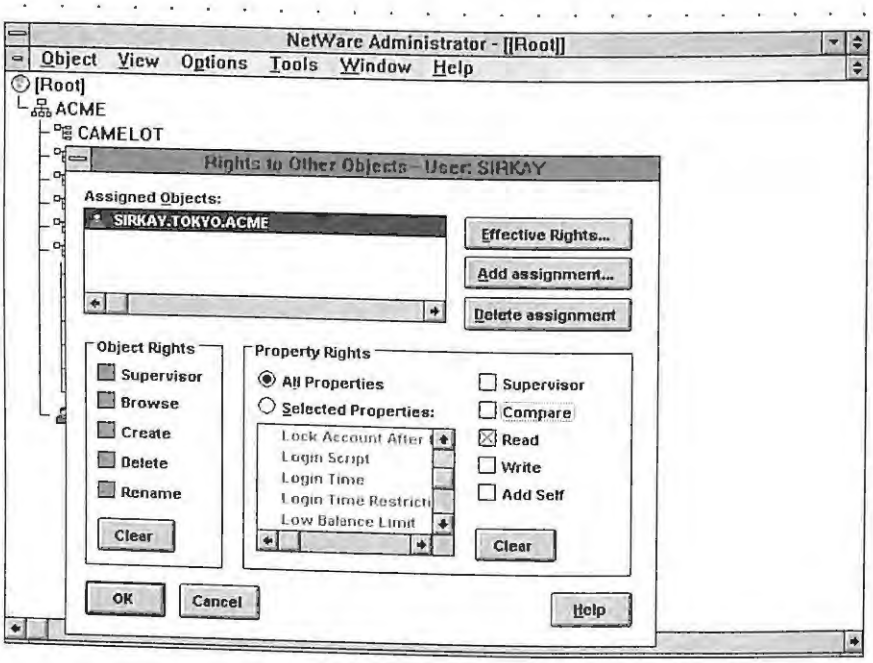
Default Trustee Assignments for Users

When user objects are created they also receive some default trustee assignments (refer back to the section on ACLs and default ACLs). These assignments greatly reduce the amount of work required by a NetWare administrator to set up user accounts and provide for their access. The following access is automatically granted during creation of a user object.

For our purposes let's assume that a new user object has been created in the TOKYO container. The newly created user object receives the Read and Compare rights to All Properties by default. The All Properties is a category that is visible with either NWADMIN or NETADMIN by selecting Rights to Other Objects as shown in Figure 13.10. Having the Read right to All Properties allows the user to read the values of his or her own user properties. The Compare right is a subset of Read and enables the value of the property to be compared with another value.

FIGURE 13.10

The user object SIRKAY.TOKYO.ACME receives by default the Read and Compare rights to All Properties for its own object.



The user object is granted read and write rights to its own login script and print job configuration. These rights permit the users to change their own login script and print job configuration if they want. Figure 13.11 shows these rights assigned along with the others when a user object is created.

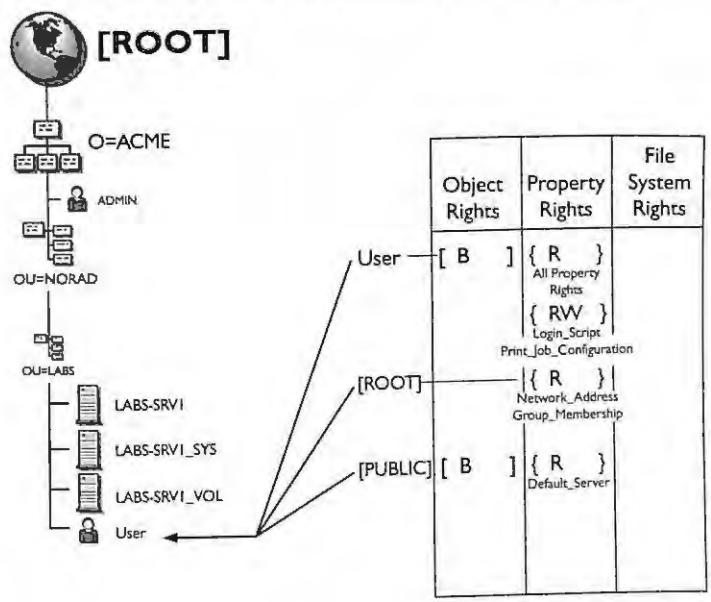
Understanding the Rules of Trustee Assignments

As was mentioned earlier, there are rules that govern the functionality of trustee assignments. Learning these rules can make it much easier for you to understand and use NetWare security.

- **Trustee assignments flow down the tree** — A trustee assignment for objects and All Property rights flows down the tree unless it is blocked by an Inherited Rights Filter (IRF). The IRF is explained later in this chapter. For example, if user SIRKAY were granted Supervisor rights to the TOKYO container, these rights would flow down to any subsequent containers and objects below TOKYO unless an object has an inherited rights filter.

FIGURE 13.11

Rights assignments are made when a user object is created.



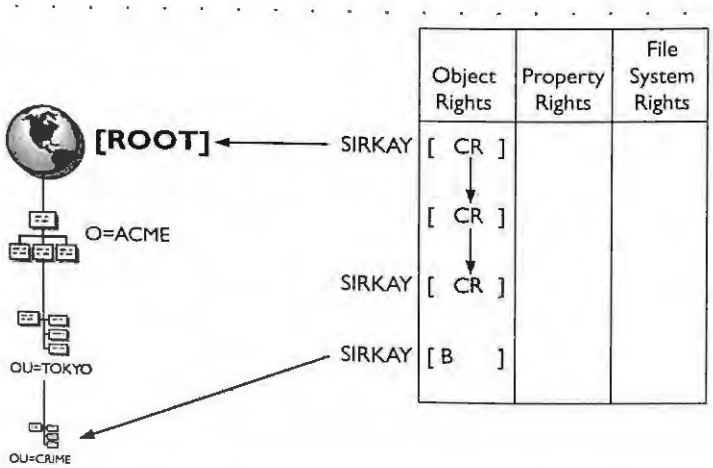
- **An explicit trustee assignment at a lower level in the tree replaces all previous trustee assignments** — As shown in Figure 13.12, user SIRKAY has been granted explicit Create and Rename rights beginning at object TOKYO. This trustee assignment flows down until it is blocked by an IRF or reassigned by another explicit assignment. In this example, we reassign the object the

BROWSE right at the OU=CRIME. This explicit assignment will replace all other higher assignments at the OU=CRIME level in the tree.

- **Selected property rights override any assignment made in the All Properties category** — At the time of user creation, a user object receives the Read property right to all of its own properties. Any selective assignment of a property right using the Selected Properties category will override anything assigned through the All Properties category. For example, by default all users have the Read property right to all of their own user object properties. Notice also that a user also receives by default the read and write rights to his login script and print job configuration. The fact that the read write is given again in the selected properties assignment indicates that it has overridden the previous assignment made in the All Properties category.

FIGURE 13.12

An explicit rights assignment made at a lower level in the tree will replace any previous explicit assignment made to that object.



- **The Access Control List (ACL) property of every object stores trustee assignments to that object** — Each object can contain a property known as the ACL. A user by default does not have write rights to its own ACL or to that of any other object. Keep in mind that some objects may not have ACLs if they already have the explicit right.

Do not grant users the write rights to any ACL, including their own user object, because the write right to the ACL controls all access to that particular object.

For example, a user possessing the write right to a container ACL has the ability to make any changes to that object's ACL. The user could assign anyone Supervisor object rights to that container and could modify the object as well.

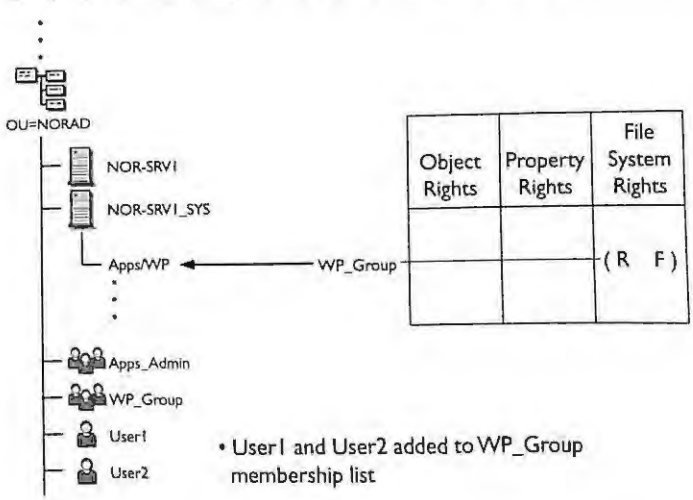
UNDERSTANDING SECURITY EQUIVALENCE

Security equivalence simply means that an object can be equivalent in rights to another object. The majority of rights assignments should be made by administrators through the use of security equivalence. It is quick and easy to use security equivalence because you can deal with a large number of users rather than a single user at a time. Time is always a factor for network administrators, and we recommend that you assign security equivalence to groups or containers as the best way to make rights assignments to large numbers of people.

To meet the needs of many users requiring the same rights to a directory or file, you can create a group, assign rights to the newly created group, and add members to the group. The members of the group are security equivalent in rights to the group object. Therefore, any assignment made to a group will be received by its members through security equivalence. An example of this process is shown in Figure 13.13.

FIGURE 13.13

An example of creating a group and granting rights to the group. The users receive rights through security equivalence.



In addition, a container functions much the same way as a group except that the group is used in your login scripts and a group can span multiple containers. If all users in a container access the same resources, it may not be necessary to use groups. However, if you want to further differentiate your environment setting within a container, the group object is an effective way to go.

Rules Governing Security Equivalence

Because the use of security equivalence will be so common on your network, it is very important to understand how NetWare 4.1 security functions. You will save time as a network administrator if you understand the rules that govern security equivalence.

- **Security equivalent rights cannot be masked** — If you receive a security equivalence, this assignment cannot be masked by an IRF.

- **Every object is security equivalent to all container objects that are part of its distinguished name. This security is known as implied security equivalence.** — For example, the user GUINEVERE.FIN.OPS.CAMELOT in the ACME tree is security equivalent to every object in its name. Therefore, if you were to grant the container FIN rights to a particular e-mail server, user GUINEVERE would receive those rights through security equivalence.


NOTE

You cannot single out users to not receive rights granted to a container by using an IRF. If you grant rights to a container, *all* users in or subordinate to that container will always receive those rights.

- **Every object is security equivalent to [ROOT]** — Once a user has successfully logged in to a server, that user is security equivalent to [ROOT].

- **Every object is security equivalent to [PUBLIC]** — [PUBLIC] with the default rights of Browse enables users to browse the tree before logging in to a server. Each user is security equivalent to [PUBLIC], and [PUBLIC] has been granted Browse rights at object [ROOT]. This assignment can be changed if you like.

CONSULTING EXPERIENCE

Be very careful with assigning rights to the [PUBLIC] trustee because of the security equivalence with all users since they do not need to be authenticated to receive those rights. The assignment of file system rights is nonfunctional when using the [PUBLIC] trustee. This means that you cannot grant access to files before the user has successfully logged in to the server. Generally, the use of the [PUBLIC] trustee for granting rights should be avoided.

- **An object is security equivalent to all objects listed in its Security Equals property** — An NDS object will keep a list (known as the Security Equals Property) of all objects that it equals in rights.

Keep in mind that when a user logs in to a NetWare 4.1 server and authenticates to the Directory, Directory Services creates what is known as a security equivalence vector that is stored in the connection table on the server. The security equivalence vector contains a list of that object's security equivalencies and is created on every server that the client authenticates to.

INHERITANCE

Inheritance is the method by which rights to objects and files flow down to subordinate levels of the tree. As previously stated, explicit trustee assignments at a higher level in your tree will flow down. The rights you receive at lower levels without assignment are known as inherited rights. Inherited rights include only the object rights and the All Properties rights. Selected property rights are not inherited.

Sometimes there is the tendency to confuse security equivalence and inheritance. Keep in mind that inheritance is simply the way that previously granted rights flow down the tree to subordinate levels.

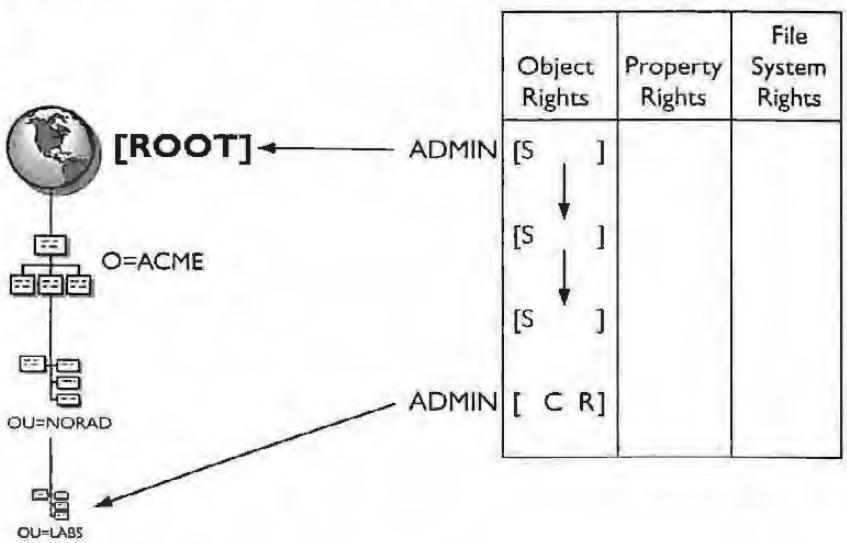
Earlier in our discussion we mentioned that explicit rights, such as the ADMIN user object possessing the Supervisor right at the [ROOT] object, flow down the tree. As shown in Figure 13.14, the Supervisor assignment continues to flow down the tree

unless it is otherwise blocked or reassigned. Therefore, at each subsequent level in the tree the ADMIN object's rights are being received through inheritance.

Inherited rights also flow down independently of other rights assignments, such as those obtained through security equivalence. This means that the rights received through inheritance are not affected by actions you may take on other explicit rights assignments. The two operate under separate rules. Do not mix up your security rules. Figure 13.15 shows how explicit rights assignments flow down independently of security equivalence rights.

FIGURE 13.14

Inheritance of the Supervisor right at the second and third levels of the tree



Understanding Inherited Rights Filters (IRFs)

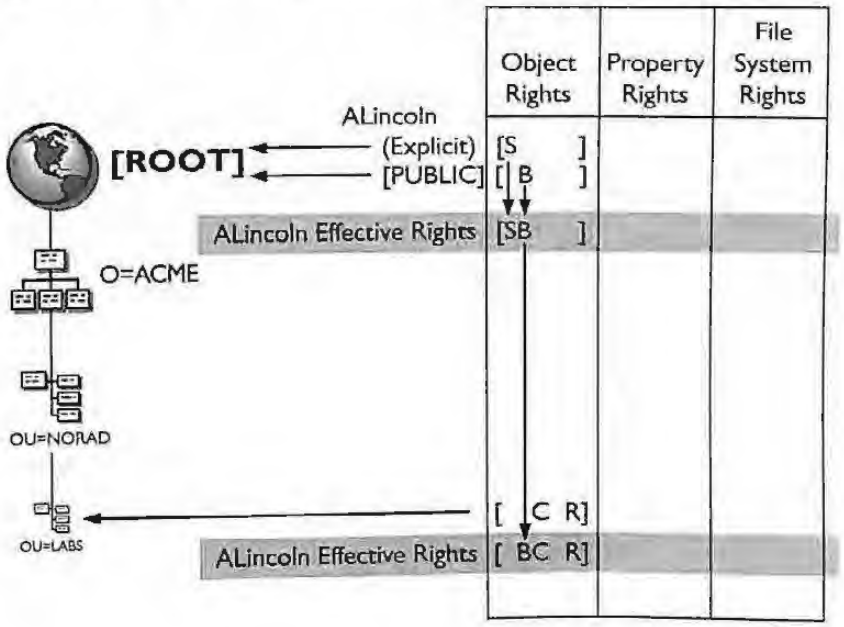
The filter known as the Inherited Rights Filter (IRF) is used to block inheritance. The IRF can be applied to object rights, the All Properties category, and the Selected Properties category.

NOTE

As mentioned earlier, you cannot place an IRF on rights received through security equivalence. You can apply the IRF only to object rights, the All Properties category, and the Selected Properties category.

FIGURE 13.15

The explicit assignment of the Supervisor right flows down the tree independently of the Browse right, which was received because user ALincoln is security equivalent to the [PUBLIC] trustee.



The IRF enables the NetWare administrator to specify which rights can be inherited from an object. It is easier to understand the concept of the IRF if you compare it to a shell around an object. When you place an IRF on an object you are placing a shell around the object. The rights that are enabled in the IRF are the only rights that users will have to an object. For example, you could place an IRF of Browse on a server object in a container. One user must maintain Supervisor rights over the object, however. All other users can inherit only the Browse right because of the IRF that is placed around the server object.

Inherited ACLs

Each partition [ROOT] object contains a property known as the inherited Access Control List. For more information on the partition [ROOT] object, refer to Chapter 6. The inherited ACL property contains the summation of ACLs from parent containers. Unless an IRF is in effect all objects in the partition will receive the rights contained in the inherited ACL. NDS can then calculate rights for objects in its partition without having to walk the Directory tree. As changes to ACLs are made to the Directory tree, NDS will update the multivalued inherited ACL property.

The NDS janitor process has the responsibility to maintain the inherited ACLs by recalculating inheritance if any changes are made to the inherited ACLs. For more information on the janitor process, refer to Chapters 8 and 9.

The ACL is both the trustee assignments made and the filters applied. The same attribute (property) name is used for both.

Rules that Govern the IRF

- **The IRF cannot grant rights; it only revokes previously assigned rights** — Keep in mind that the IRF is a shell wrapped around an object.
- **You can enable an IRF for every Object, Property, File, and Directory** — In most cases you will not need to use that many IRFs because the user's default rights are limited to begin with. As shown in the scenarios at the end of the chapter, most IRFs are used to protect servers and to separate file system and NDS administration.
- **The Supervisor Object/Property rights can be revoked by an IRF** — An IRF can be applied to all objects, including the server object. Therefore, you can limit a person's Supervisor access to a NetWare 4.1 file server by applying an IRF to the server object. Remember that a user possessing the managed right (write right to the ACL) to a server object also has rights to the file system volumes for that server as well.
- **The Supervisor File/Directory rights cannot be revoked by an IRF** — This feature is identical to NetWare 3 Supervisor rights in that any user that has Supervisor rights to a file system directly cannot have file system rights masked on that file server.

UNDERSTANDING EFFECTIVE RIGHTS

The last step in the security pyramid is the calculation of effective rights. Effective rights are what an object can actually do after all other security factors are calculated

against the object. The following sources are used in the calculation of effective rights of one object to another:

- The object's ACL
- The object's explicit assignments
- All security equivalent access privileges

For example, we will discuss an object A with access to object B using security equivalence calculated at the time of authentication. The rights would be calculated as follows:

1. The sum of explicit assignments would be calculated back to partition root. Object B would be calculated back to object B's partition root. Of course, an IRF would negate some assignments.
2. Add in the inherited ACLs from partition root.
3. Object A receives all explicit and inherited ACLs to which A is security equivalent.

UNDERSTANDING MANAGED RIGHTS

Managed rights (or management rights) is a term used to describe an object (ADMIN, for example) that has the write right to an object's ACL. Managed rights means that the trustee has all power over an object and can modify anything pertaining to that object. For some operations in NDS you must have managed rights to perform that operation. Below is a list of NDS operations and the managed rights that are required to perform them:

- All partition operations including Create, Merge, Add replica, and Move Subtree require the trustee to have write rights to the target partition. The Merge operation requires managed rights to the [ROOT] objects of both trees.
- A schema modification requires the trustee to have write rights to the ACL of the Directory's [ROOT] object.

- Any modifications to the following properties require write rights to that object's ACL:
  - Security Equals
  - Group Membership
  - Profile Membership
- Backup requires managed rights on the object(s) being backed up.
- The Add/Remove replica operation requires the following:
  - Managed rights on the partition root
  - Managed rights on the target server

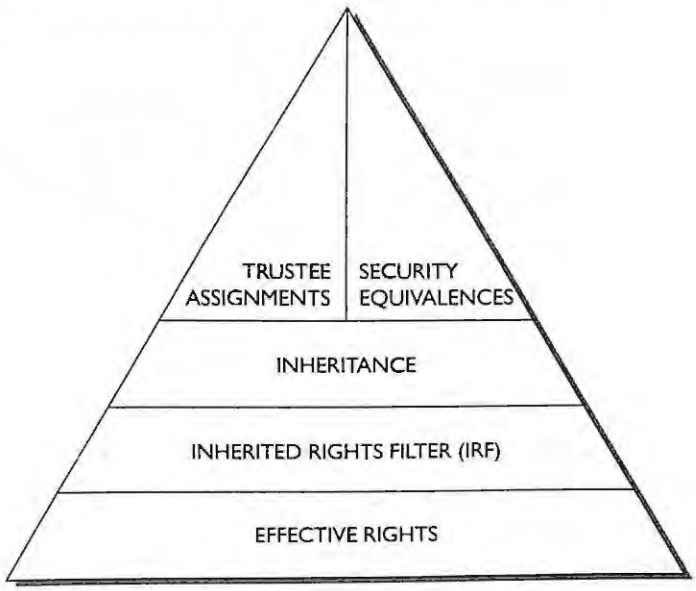
Implementing NDS Security

With an understanding of the basic concepts of security, we can now begin a discussion of how to use and implement NDS security for your network. Figure 13.16 is the basis for our security discussion. As stated earlier, the pyramid shown in this figure shows a very logical approach for understanding NetWare 4.1 security. Each section of the pyramid will now be explained with examples on how you can implement security in your environment for the greatest benefit.

For our examples we will refer to the ACME tree to implement security procedures throughout the entire organization. The following scenarios will be discussed in terms of NetWare security. All scenarios make the assumption that you are an administrator with object Supervisor rights at the [ROOT] of your tree.

FIGURE 13.16

The NDS security pyramid shows graphically the order in which rights are assigned.



SECURITY REQUIRED TO INSTALL A NETWARE 4.1 SERVER UNDER THE OU=NORAD CENTER

Security Concepts to Understand

- Trustee Assignments
- Security Equivalence

As a temporary administrator you are asked to install a NetWare 4.1 server in the NORAD container into the ACME tree. There are currently no administrators in your location, and this server must be brought up immediately.

1 • Contact your main administrator to obtain Create rights for the NORAD container. Supervisor rights at your container are needed to install a NetWare 4.1 server into your own container or to add a partition replica to partition root. This can be accomplished in several ways as described in the following steps.

2 • The first way is to simply have the administrator explicitly grant you Supervisor rights to the NORAD container. This method is difficult to track if many similar requests are made to the main NDS administrator.

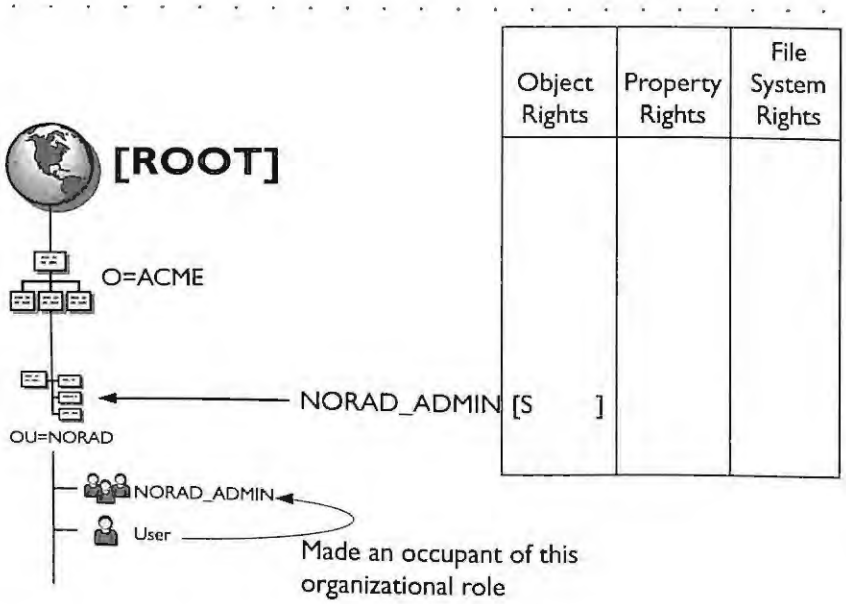
3 • The second and recommended way is to use NWADMIN or NETADMIN to create an organization role in the NORAD location and grant the role Supervisor rights to the NORAD container known as NORAD_ADMIN.

4 • The main administrator can then move you in temporarily as an administrator so that you can install the NetWare 4.1 server. A NetWare 4.1 installation with an add replica will NOT complete unless you have Supervisor rights to the container in which the server is being installed.

An example of this entire scenario is shown in Figure 13.17.

FIGURE 13.17

Rights necessary to install a server in the NORAD container



An example of an organizational role being created with NWADMIN is shown in Figures 13.18, 13.19, and 13.20.

FIGURE 13.18

Creating an organizational role

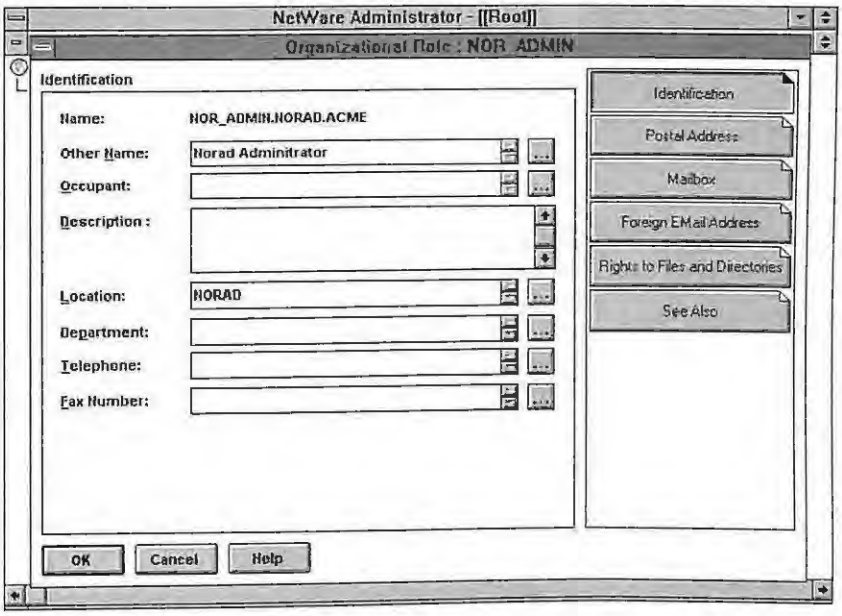


FIGURE 13.19

Assigning the necessary rights

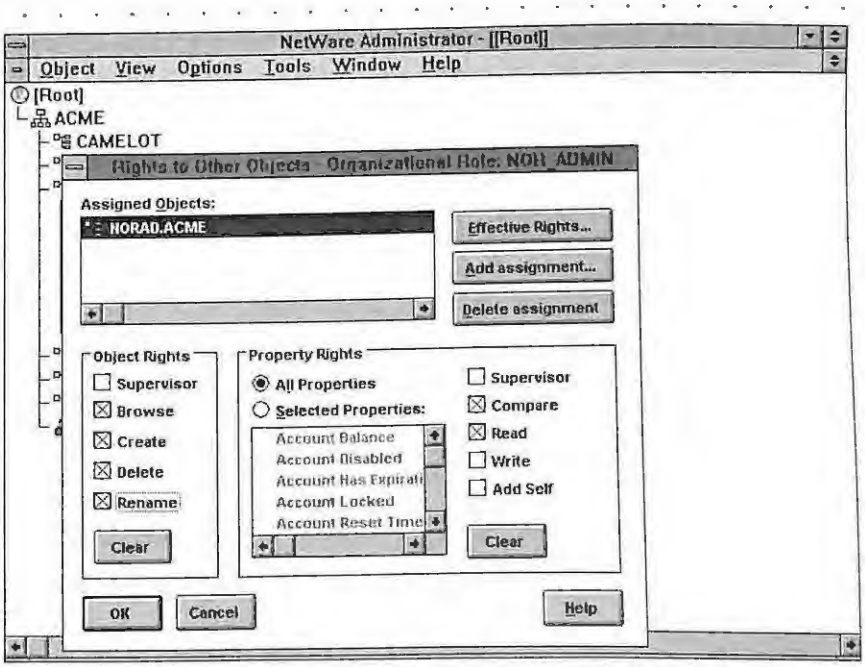
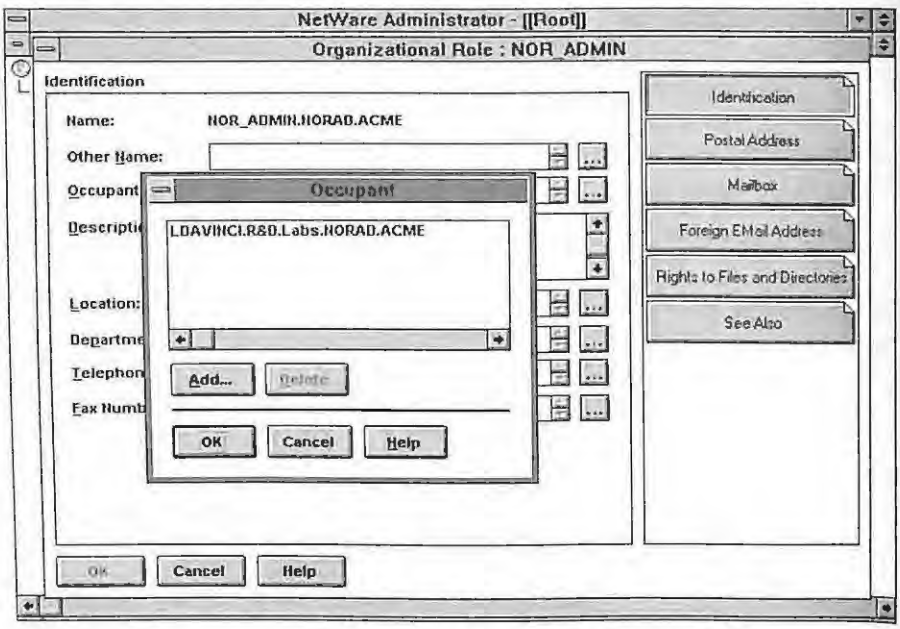


FIGURE 13.20

Moving a user into the role as an occupant



SECURITY REQUIRED TO INSTALL AN APPLICATION ON YOUR NETWARE 4.1 SERVER IN THE CAMELOT CONTAINER AND GRANT APPLICATION ACCESS TO YOUR USERS

Security Concepts to understand

- File System Trustee Assignments
- Supervisor rights to a server object

You are a file system administrator in the CAMELOT location responsible for installing all new applications on the location's file servers.

1. You must have at a minimum the Create right to the APPS subdirectory on your NetWare 4.1, for example. In most cases you will use Supervisor trustee rights to perform these operations.

2. If you have Supervisor object rights over the file server object, you will also have Supervisor rights over the file system.

3 • Install your application according to the directions.

4 • Make sure that all executable files related to this application are flagged as sharable read only. Most application installations automatically do this for you, but it doesn't hurt to check.

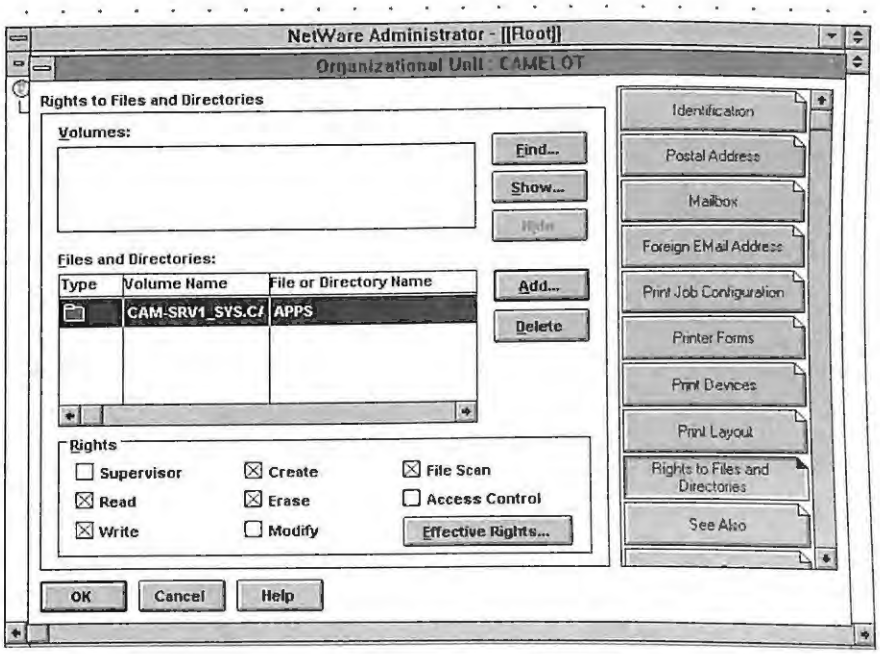
5 • Create any supporting objects that may be needed such as groups or directory maps. This requires Create rights at the container. You may also need file system rights as well.

6 • Consider using Novell's NetWare Application Manager to launch applications as NDS objects from a user's desktop. For more information on the NetWare Application Manager, refer to Chapter 2.

An example of this entire scenario is shown in Figure 13.21.

FIGURE 13.21

Using NWADMIN to assign rights necessary to install an application on a NetWare 4.1 server in the CAMELOT container



SECURITY PROCEDURES FOR GRANTING AN INDIVIDUAL RIGHTS TO MANAGE A HELP DESK CENTER AT THE CAMELOT LOCATION

Security Concepts to Understand

- Trustee Assignments
- Security Equivalence

Your responsibility is to assist in managing a help desk at the CAMELOT location.

1. Using NWADMIN or NETADMIN, create a series of specialized organizational roles for your help desk administration such as user administrators, server administrators, and tree administrators. Although currently Directory Services does not enforce rights to a specific object class, you can create organizational roles that designate these type of administrators.
2. Assign Create, Delete, and Rename rights to the user administrator's role.
3. Assign Supervisor file system rights to the server administrator's role for each server in the container to be managed.
4. Create an organizational role at the top of your tree with explicit Supervisor rights at [ROOT]. Move the top help desk administrators into the organizational role as occupants.
5. Make IRF assignments where appropriate to limit administrator access to certain areas of your system.

Examples of this entire scenario are shown in Figures 13.22, 13.23, and 13.24.

FIGURE 13.22

Creating an organizational role for the user administrator

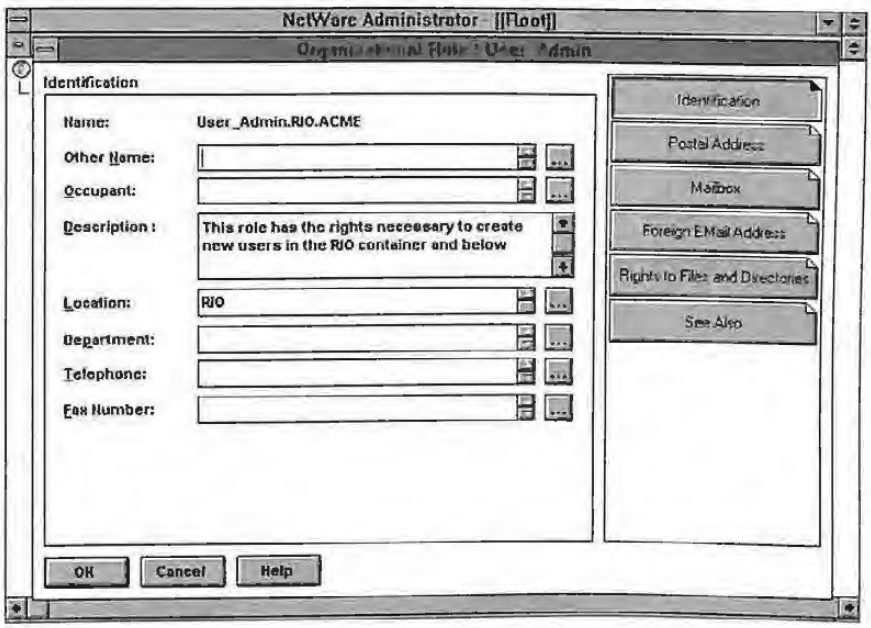


FIGURE 13.23

Creating an organizational role for the server administrator

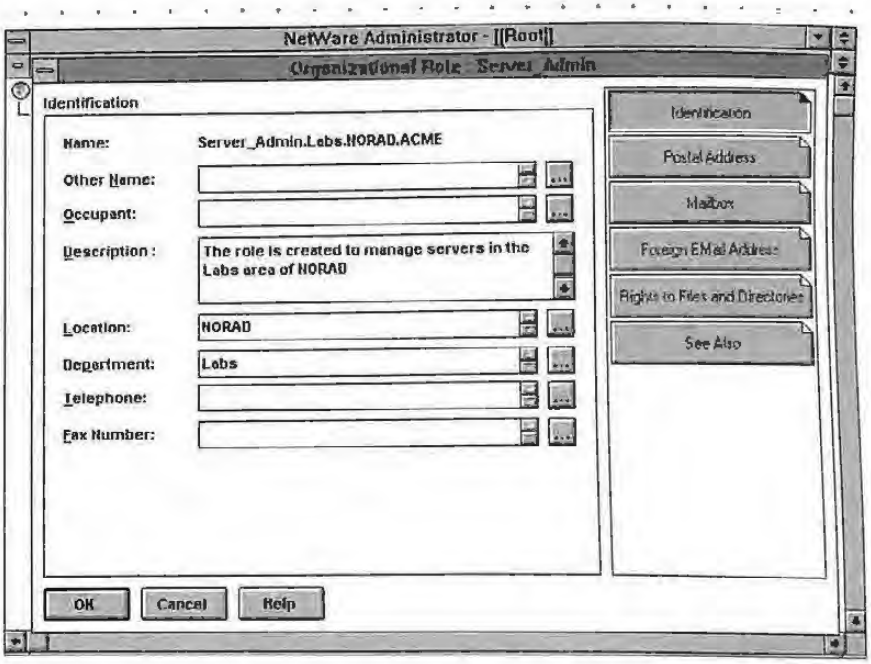
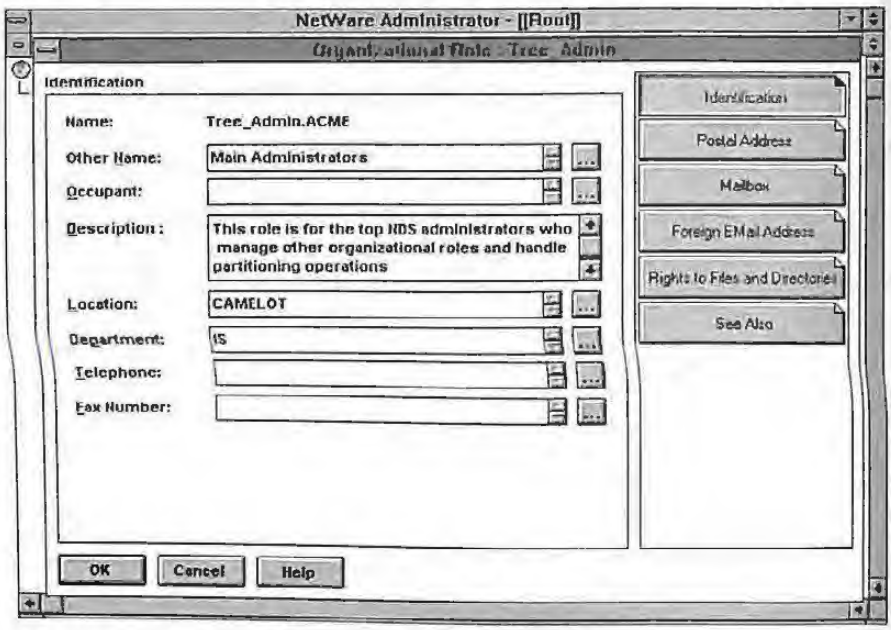


FIGURE 13.24

Creating an organizational role for the tree administrators



CREATION OF SUBADMINISTRATORS FOR EACH MAJOR LOCATION IN THE ACME TREE

Security Concepts to Understand

- Trustee Assignments
- Security Equivalence
- Inheritance

You have been given the assignment to create subadministrators for each major location in the ACME network and assign individuals to manage the network from that level down. You currently manage the network with only the ADMIN user object.

1. Using NWADMIN or NETADMIN, create an organizational role object in the containers NORAD, RIO, TOKYO, CAMELOT, SYDNEY, and TOKYO.

2 • Using NWADMIN or NETADMIN, grant the newly created organizational role objects Supervisor rights to their respective containers.

3 • Choose an administrator or administrators to participate in the organizational role and add them as role occupants using NWADMIN or NETADMIN.

4 • Assign the ADMIN user explicit Supervisor rights to each of the organizational role objects.

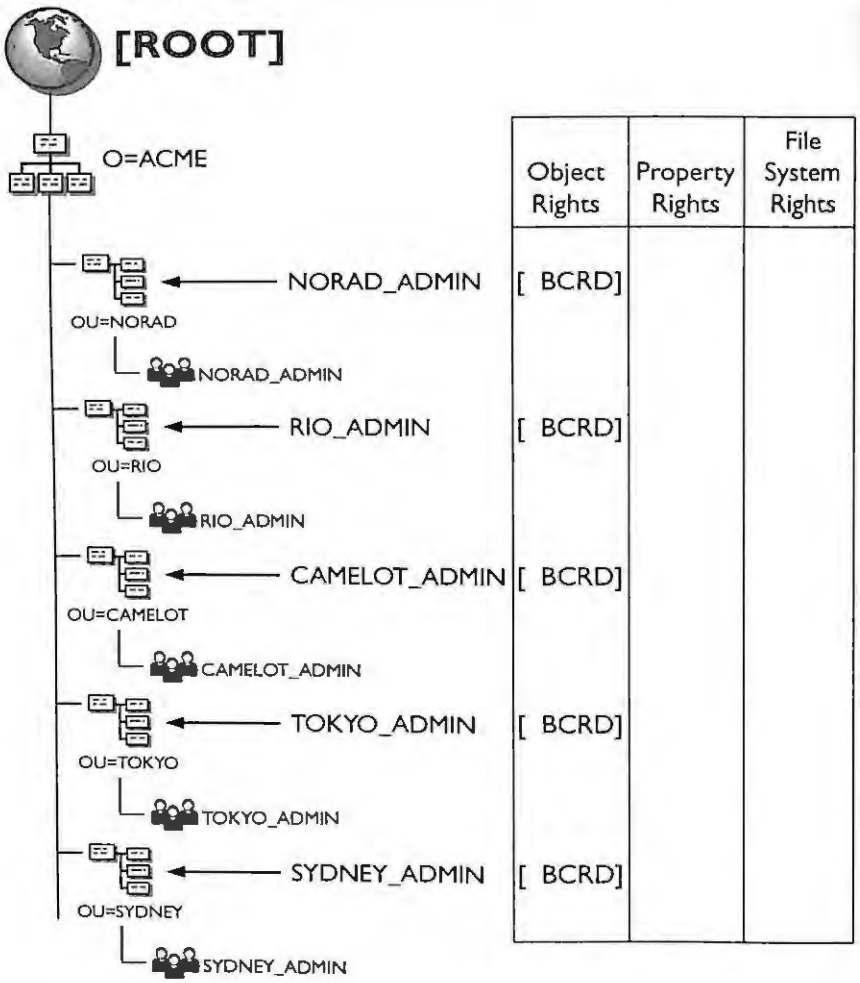
5 • Using NWADMIN or NETADMIN, place an Inherited Rights Filter of Browse (and possibly Read) on each of the organizational roles to prohibit management of the role by the occupants.

6 • These new administrators will have the power to create additional objects, including subordinate containers, in their respective locations. The administrators have Supervisor rights at these lower levels in the tree through inheritance.

An example of this entire scenario is shown in Figure 13.25.

FIGURE 13.25

Creating organizational roles for each location in the ACME tree



CREATING A FILE SYSTEM ADMINISTRATOR AND AN NDS ADMINISTRATOR IN THE OU=TOKYO LOCATION

Security Concepts to Understand

- Trustee Assignments
- Security Equivalence
- Inherited Rights Filter

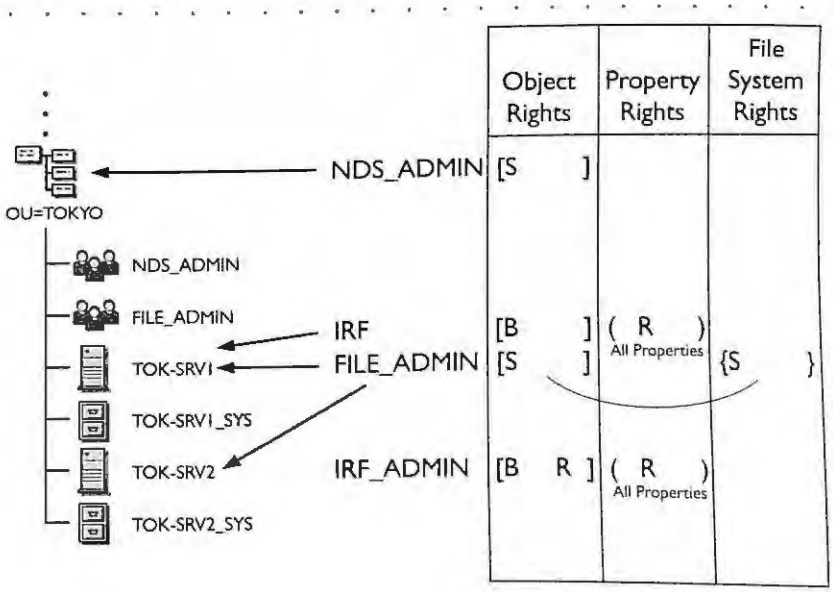
Administration in the TOKYO container is being broken out into two responsibilities. One individual will handle only the file system administration, while the other administrator will handle NDS administration. Each must have separate rights.

1. Using NWADMIN or NETADMIN, create two organizational roles. Name the first role NDS_ADMIN and the second role FILE_ADMIN.
2. Using NWADMIN or NETADMIN, assign the NDS_ADMIN role Supervisor Rights to the TOKYO container.
3. Using NWADMIN or NETADMIN, assign the FILE_ADMIN role Supervisor rights to the server objects TOK-SRV1 and TOK-SRV2.
4. Using NWADMIN or NETADMIN, place an Inherited Rights Filter of Browse (and possibly Read) on the server objects TOK-SRV1 and TOK-SRV2.

An example of this entire scenario is shown in Figure 13.26.

FIGURE 13.26

Separating NDS and file system administration



RIGHTS NECESSARY TO PERFORM PARTITIONING OPERATIONS AT DIFFERENT LEVELS IN THE NDS TREE

You have been made responsible for partitioning operations on the ACME network. You want to prohibit all partitioning operations except by you and another administrator.

1. Using NWADMIN or NETADMIN, grant the other administrators in their organizational roles all rights to their respective containers except the Supervisor object right. Grant Read and Create rights on the container ACL also. (Keep in mind that other administrators will not be able to install NetWare 4.1 servers into the tree without the Supervisor right to the container if they are adding a replica. You can grant them the right temporarily to handle this situation or add the replica for them.)

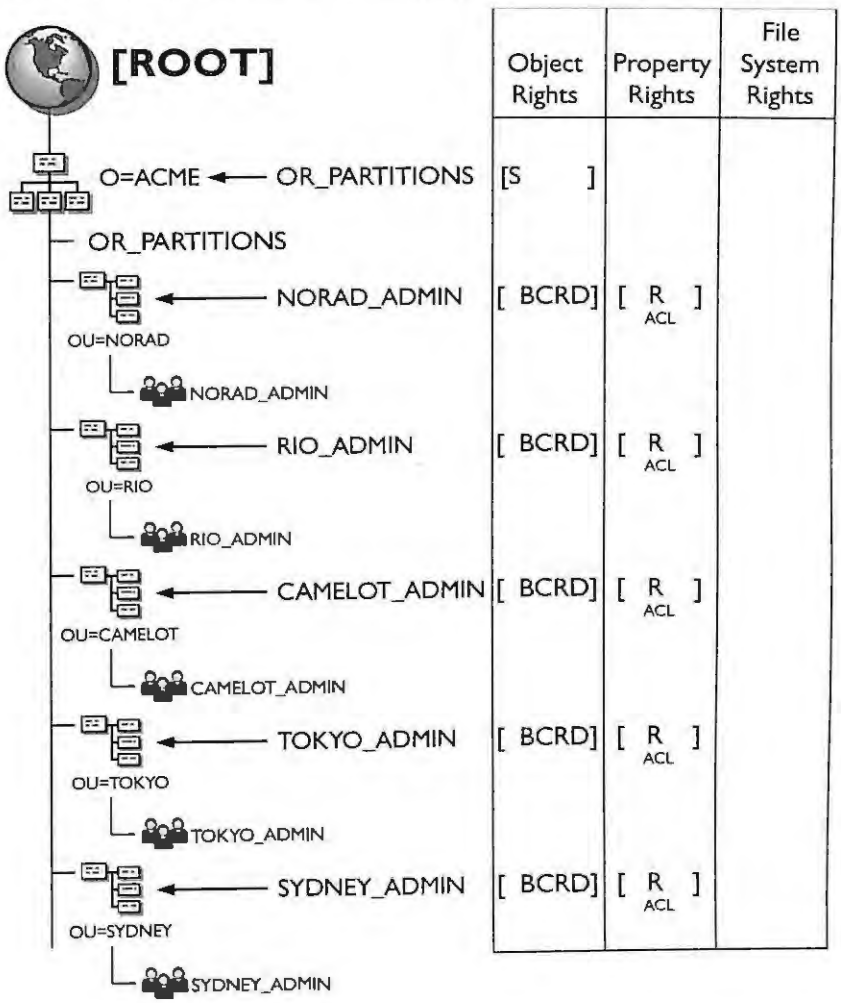
2. Create an organizational role for yourself and other tree administrators for partitioning operations called OR_PARTITIONS.

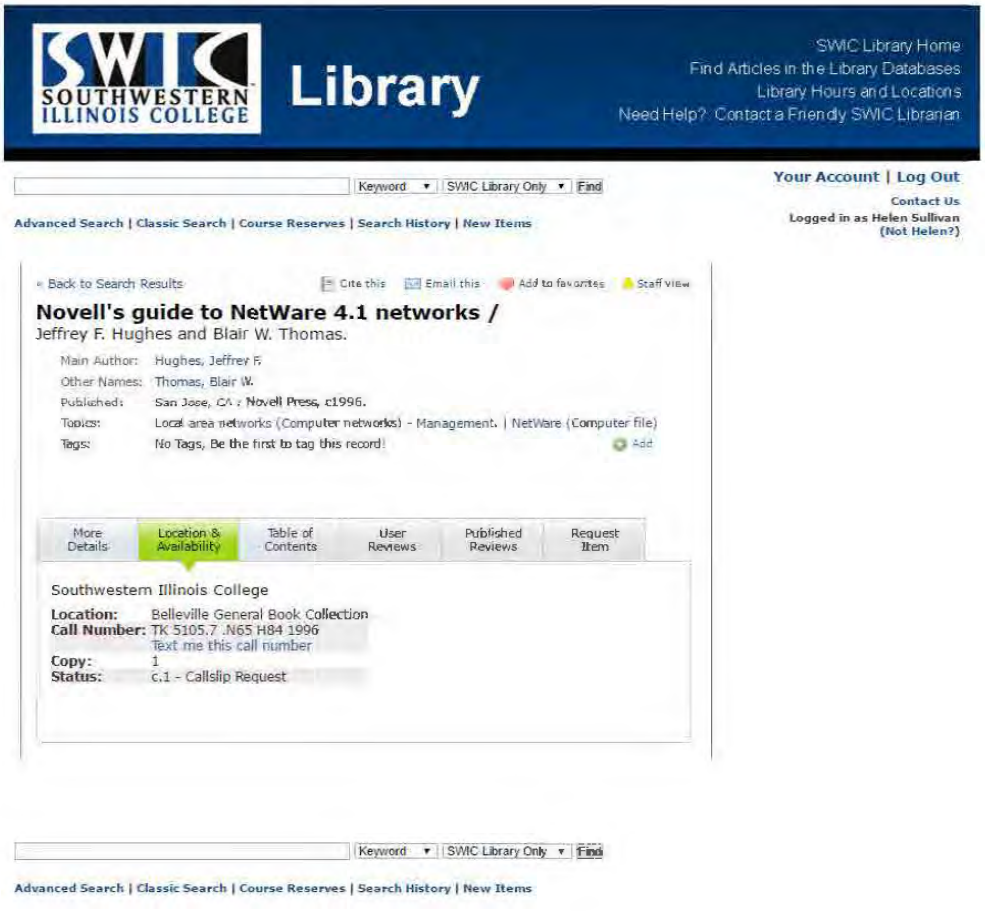
3. If you have not already done so, grant the organizational role explicit object Supervisor rights to each container immediately subordinate to [ROOT].

An example of this entire scenario is shown in Figure 13.27.

FIGURE 13.27

Rights needed to perform partitioning operations on the ACME tree




SWIC
SOUTHWESTERN ILLINOIS COLLEGE
Library
SWIC Library Home
Find Articles in the Library Databases
Library Hours and Locations
Need Help? Contact a Friendly SWIC Librarian
Keyword
SWIC Library Only
Find
Your Account | Log Out
Contact Us
Logged in as Helen Sullivan
(Not Helen?)
Advanced Search | Classic Search | Course Reserves | Search History | New Items
« Back to Search Results
Cite this
Email this
Add to favorites
Staff view
Novell's guide to NetWare 4.1 networks /
Jeffrey F. Hughes and Blair W. Thomas.
Main Author: Hughes, Jeffrey F.
Other Names: Thomas, Blair W.
Published: San Jose, CA : Novell Press, c1996.
Topics: Local area networks (Computer networks) - Management. | NetWare (Computer file)
Tags: No Tags, Be the first to tag this record!
Add
More Details
Location & Availability
Table of Contents
User Reviews
Published Reviews
Request Item
Southwestern Illinois College
Location: Belleville General Book Collection
Call Number: TK 5105.7 .N65 H84 1996
Text me this call number
Copy: 1
Status: c.1 - Callslip Request
Keyword
SWIC Library Only
Find
Advanced Search | Classic Search | Course Reserves | Search History | New Items

Novell's Guide to NetWare 4.1 Networks

Authors: Jeffrey F. Hughes
Thomas

Publication:

Book
Novell's Guide to NetWare 4.1 Networks
1st
IDG Books Worldwide, Inc. Foster City, CA, USA ©1995
ISBN: 156884736X

1995 Book

Title	Novell's Guide to NetWare 4.1 Networks 1st
Pages	942
Publisher	IDG Books Worldwide, Inc. Foster City, CA, USA ©1995
ISBN	156884736X

Novell's guide to NetWare 4.1 networks /

Jeffrey F **Hughes**; Blair W **Thomas**

1996
English Book xxxiv, 942 pages : illustrations + 1 computer optical disc (4 3/4 in.)
San Jose, CA : Novell Press, : ISBN: 156884736X 9781568847368

GET THIS ITEM

Availability: **FirstSearch indicates your institution owns the item.**
- Libraries worldwide that own item: 71 UNIV OF ILLINOIS
- Search the catalog at the Library of University of Illinois at Urbana-Champaign

External Resources:
- Discover full text Discover UIUC Full Text
- Interlibrary Loan Request
- Cite This Item

FIND RELATED

More Like This: Search for versions with same title and author | Advanced options ...

Title: **Novell's guide to NetWare 4.1 networks /**

Author(s): Hughes, Jeffrey F.
Thomas, Blair W.

Publication: San Jose, CA : Novell Press,

Year: 1996

Description: xxxiv, 942 pages : illustrations + 1 computer optical disc (4 3/4 in.)

Language: English

Standard No: **ISBN:** 156884736X; 9781568847368; **National Library:** 156-88473 **LCCN:** 95-82357

Report No: 19582357

SUBJECT(S)

Descriptor: Local area networks (Computer networks) -- Management.
Local Area Networks.
Local area networks (Computer networks) -- Management.
Redes de computadores.
NetWare 4.1.

Identifier: Local area networks (Computer networks); Management; NetWare; NetWare; Computer file

Title Subject: NetWare.
NetWare.

Note(s): CD-ROM: Two user version of Netware 4.1./ One pull-out poster attached to back of book./ Includes index./ **Report:** 19582357

System Info: System requirements for CD-ROM: PC or compatible; minimum 386 with SX or DX processor; minimum 8 MB RAM; 100 MB of available hard drive space; CD-ROM drive.

Class Descriptors: **LC:** TK5105.8.N65; **Dewey:** 005.7; **NLM:** TK 5105.7 H893na 1996

Other Titles: Guide to NetWare 4.1 networks

Responsibility: Jeffrey F. Hughes and Blair W. Thomas.

Vendor Info: Baker & Taylor Baker and Taylor YBP Library Services (BKTY BTCP YANK) 59.99 **Status:** active

Document Type: Book

Entry: 19960105

Update: 20170223

Accession No: **OCLC:** 34619636

Database: WorldCat

Public Catalog

Copyright Catalog (1978 to present)
Search Request: Left Anchored Title = novell's guide to netware 4.1
Search Results: Displaying 1 of 1 entries

previous next

Labeled View

Novell's guide to NetWare 4.1 networks / Jeffrey F. Hughes and Blair W....

Type of Work: Text
Registration Number / Date: TX0004715185 / 1996-04-03
Title: Novell's guide to NetWare 4.1 networks / Jeffrey F. Hughes and Blair W. Thomas.
Imprint: San Jose, CA : Novell Press, c1996.
Description: 942 p. + CD-ROM.
Copyright Claimant: Jeffrey F. Hughes and Blair W. Thomas
Date of Creation: 1996
Date of Publication: 1996-03-08
Previous Registration: Preexisting material: software on CD-ROM.
Copyright Note: C.O. correspondence.
Other Title: guide to NetWare 4.1 networks
Names: Hughes, Jeffrey F.
Thomas, Blair W.

previous next

Save, Print and Email (Help Page)		
Select Download Format	Full Record	Format for Print/Save
Enter your email address:		Email

LIBRARY: Auburn University Libraries
YOU SEARCHED: Keyword = (156884736X)[in]
SEARCH RESULTS: Displaying 1 of 1 entries

previous next

Brief Record | Full Record | Table of Contents | MARC Record

Novell's guide to NetWare 4.1 networks / Jeffrey F. Hughes and Blair W....

000 01202mam a2200349 a 450
001 1252139
005 20130704081824.0
006 m u
008 960424s1996 caua 001 0 eng d
010 __ |a 95082357
020 __ |a ***156884736X***
035 __ |a (OCoLC)34619636
035 __ |9 AHD9609AU
040 __ |a JHW |c JHW
049 __ |a AAAA
090 __ |a TK5105.7 H84 1996
100 1_ |a Hughes, Jeffrey F.
245 10 |a Novell's guide to NetWare 4.1 networks / |c Jeffrey F. Hughes and Blair W. Thomas.
246 30 |a Guide to NetWare 4.1 networks
260 __ |a San Jose, CA : |b Novell Press, |c c1996.
300 __ |a xxxiv, 942 p. : |b ill. ; |c 24 cm. + |e 1 CD-ROM 4 3/4 in.)
336 __ |a text |2 rdacontent
337 __ |a unmediated |2 rdamedia
338 __ |a volume |2 rdacarrier
500 __ |a CD-ROM: Two user version of Netware 4.1.
500 __ |a One pull-out poster in pocket.
500 __ |a Includes index.
538 __ |a System requirements for CD-ROM: PC or compatible; minimum 386 with SX or DX processor; minimum 8 MB RAM; 100 MB of available hard drive space; CD-ROM drive.
650 _2 |a Local Area Networks (Computer networks)
700 1_ |a Thomas, Blair W.
948 __ |a jb
999 __ |a 1252139

previous next

FORMAT Records to *Print*, *Save*, or *E-Mail*

Format Type: Full Record | DISPLAY Reformatted Records

Enter your email address: | Email