HP 9000 Networking

NetWare® 4.1/9000
Introduction to NetWare Directory Services

HP Part No. J2768-90005
Printed in U.S.A.
E1296

HEWLETT® PACKARD

Notice

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material. This product is based in whole or in part on technology developed by Novell, Inc

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard

Restricted Rights Legend

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DoD agencies, Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

**Hewlett-Packard Co.
19420 Homestead Road
Cupertino, CA 95014 USA**

Printing History

New editions are complete revisions of the manual. The dates on the title page change only when a new edition or a new update is published.

Note that many product updates and fixes do not require manual changes and, conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.

First Edition: December, 1996 (HP-UX Release Dart-31)

Conventions

This document uses the following conventions for displaying the syntax of user-entered commands:

- Commands, displays, and user input are shown in bold Courier font, for example: **`sam`**
- Words in ***`italics`*** denote a parameter that must be replaced by a user-supplied variable.
- Elements inside brackets (**`[]`**) are optional. You can select any one or none of the elements within the brackets.
- When multiple elements are enclosed within braces (**`{}`**), you must select one of the elements.

Contents

Contents

Contents

Contents

Contents

Contents

Contents

1

Concepts

Overview

The NetWare® Directory Services™ (NDS) technology is a distributed name service that provides global access to all network resources regardless of where they are physically located.

Users log in to a multiserver network and view the entire network as a single information system. This system is the basis for increased productivity and reduced administrative costs.

This section provides you with conceptual information to assist you in understanding the NDS™ technology and its features.

Contents

This section is divided into four chapters, with the following information discussed in the following sections:

Purpose	Chapter
To learn more about NetWare Directory Services features	Chapter 2, “Understanding NetWare Directory Services”
To learn more about management features in NetWare Directory Services	Chapter 3, “Understanding Management Features”
To learn more about bindery services in NetWare Directory Services	Chapter 4, “Understanding Bindery Services”
To learn more about time synchronization in NetWare Directory Services	Chapter 5, “Understanding Time Synchronization in NDS”

2

Understanding NetWare Directory Services

Overview

This chapter introduces and describes the NetWare® Directory Services™ (NDS) technology and its functionality on your network.

The following topics are discussed on the indicated pages:

To understand the technology and functionality provided by the NDS™ software, you must first understand some of the basic features of directory services technology and its implementation in the Novell® NetWare Directory Services products.

What are Directory Services?

Directory services are databases of information with powerful facilities for storing, accessing, managing, and using diverse kinds of information about users and resources in computing environments.

Standard Directory Services

Directories have traditionally been a component of the computing or network infrastructure that provide services to applications, such as E-mail, human resources, and network management applications. However, no integrated network directory services has been available to applications and users alike.

Users and organizations within computing environments are recognizing the need for a common, distributed directory that provides services to all network applications and users across disparate platforms including hosts, minicomputers, and network systems.

This need is driven by an overall connectivity paradigm, the continuing trend towards downsizing, and the need for directory integration and centralized management.

The NetWare Directory Services technology provided by Novell maintains a single, network-wide directory that is accessible from multiple points by users and applications.

NetWare Directory Services

NetWare Directory Services (NDS) is an object-oriented implementation of directory services that allows you to build sophisticated naming schemes and databases across network-wide resources.

The NDS architecture provides global access to all network resources regardless of where the resources are physically located—forming a single information system.

The following table provides a brief discussion of the features and benefits of NDS.

NOTE: You will encounter several new terms as you work with NDS. These are defined in the following discussion of the basic architecture and design of NDS.

Table 2-1 Features and Benefits Provided by NetWare Directory Services

Feature	Benefit
Simple Administration	The single point of administration provided in the NDS architecture allows for simple and cost-effective management of your entire network and its resources. Each supervisor of your network uses the same management utilities and database of resource objects regardless of each supervisor’s physical location on the network. Network resources, such as users and groups, also maintain a single point of access to the network. This allows you to maintain a single identity for each resource you create throughout the entire network.
Advanced Security	The NDS architecture provides the possibility of improved security. NDS incorporates the advanced RSA (Rivest, Shamir, and Adleman, developers of this particular public key encryption system) security features that make encrypted, single-login authentication to network resources possible. NDS security is based on a top-down architecture. All rights to network resources are established through Access Control Lists (ACLs) that allow for sophisticated, but easily managed, administration.
Usability	The hierarchical database structure of the NDS design reduces network traffic and makes retrieving objects and properties very easy and efficient. You can search the entire Directory tree to locate an object, or a search can be initiated at any level of the Directory tree. Enhanced searching techniques allow objects to be located in a variety of ways, such as using relational expressions and wild cards. Also, objects in the Directory tree do not advertise. Traffic is generated only when an application asks the Directory for information and to allow for synchronization of NDS databases.
Reliability	The replicated nature of NDS creates a fault-tolerant system to ensure that you have no single point of failure in your network system. If implemented correctly, your network maintains operation through routine hardware and software maintenance. Synchronization of Directory replicas is automatic and does not require any administrative intervention.

Table 2-1 Features and Benefits Provided by NetWare Directory Services

Feature	Benefit
Flexibility	The hierarchical design of NDS allows for easy alteration of the network structure. Components of the network can be merged or split as needed. You can move objects from one part of the Directory tree to another.
Scaleability	NDS has a modular design that allows you to customize it for any size and type of network. This means that as your organization changes to incorporate more resources and services, or downsizes to meet more specialized needs, the architecture and management of your network remains the same.
Interoperability	NDS provides compatibility with existing Novell and third-party products. Specifically, NDS is capable of providing bindery services used in the bindery-based NetWare network operating systems. This allows for an easier and more flexible transition of bindery-based NetWare servers, utilities, and client software to NDS. Furthermore, NDS provides centralized management of your bindery-based server and resources in the network.
Future-Looking	The functionality that defines how the Directory tree is constructed can be modified and expanded to suit your present and future needs. If the default definitions do not meet your needs, you can create an entirely new set of definitions or make modifications to parts of the existing definitions.

The Hierarchical Directory Tree

NetWare Directory Services (NDS) was developed as a hierarchical design with multiple levels of organizational units, users, groups, and network resources. This hierarchical structure is referred to as the Directory tree. The Directory tree is formed by organizing objects in a multilevel structure.

NDS and the X.500 Specification

NetWare Directory Services is consistent with the emerging international standard, X.500. The X.500 specification was developed by the CCITT (Consultative Committee for Telegraphy and Telephony) to provide a standard method for organizing information that is accessed transparently on a global basis.

Information such as telephone directories, corporate organizational structures, and directories of available services are all accessible through products compatible with this specification.

Much of the current development for accessing services on the global internetwork is being done according to the X.500 specification.

Directory Schema

The NDS Directory tree is defined by a set of rules called the Directory schema. The schema defines the specific way information is stored in the Directory database.

The following information is defined by the schema:

- **Object classes**. Provide the basis for all entries in an NDS database. The set of defined object classes is referred to as the base schema. For example, servers, users, and print queues are some of the base object classes defined by the base schema.
- **Attribute information**. Describes the additional information an object can or must have associated with it. Attribute types (or properties) are defined within the schema by specific constraints and a specific syntax for their values.

- **Inheritance**. Determines which objects inherit the properties and rights of other objects.
- **Naming**. Determines the structure of the Directory tree, thus identifying and showing an object’s reference name in the Directory tree.
- **Subordination**. Determines the location of objects in the Directory tree.

For a complete list of the base object classes, as well as other Directory information, Appendix B for more information.

Directory Objects

Directory objects consist of categories of information, known as properties, and the data included in those properties. This information is stored in the Directory database.

The Directory database contains three general types of objects.

- [Root] object (Directory tree name)
- Container objects
- Leaf objects

The following figure illustrates the hierarchy of Directory objects in NetWare Directory Services. (The icons represent the objects as they appear in the NetWare Administrator graphical utility.) Please note that the AFP Server is not supported in NetWare Services.

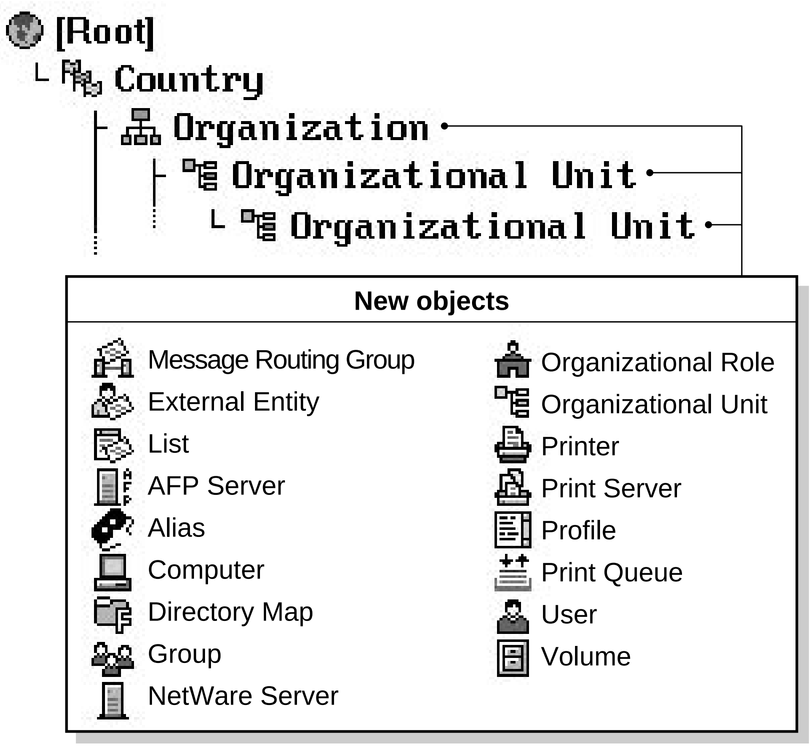


Figure 2-1 **Hierarchy of Directory Objects**

These objects represent both physical and logical resources on the network, such as users and printers or groups and print queues.

Directory objects are structures that store information, not the actual entity represented by the object. For example, a Printer object stores information about a specific printer and helps manage how the printer is used, but it is not the actual printer itself.

This Directory tree structure has the tree growing upside down, starting with the name of the tree or [Root] object at the top of the tree and branching downward. Once the [Root] object is named, you reference that object by its given name.

The following figure illustrates how objects can be laid out to form the Directory tree.

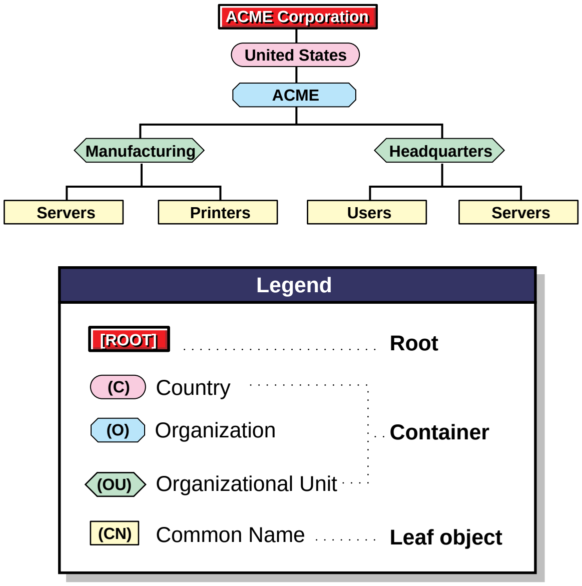


Figure 2-2 **Objects Used in a Directory Tree**

The Directory tree name ([Root] object) is automatically placed at the top of the tree during installation. Branches of the Directory tree consist of container objects and all of the objects they hold. Container objects can hold other container objects. Leaf objects are at the ends of the branches and do not contain any other objects.

The following figure illustrates that the Directory tree is formed by container objects and leaf objects branching down from the tree name or [Root] object.

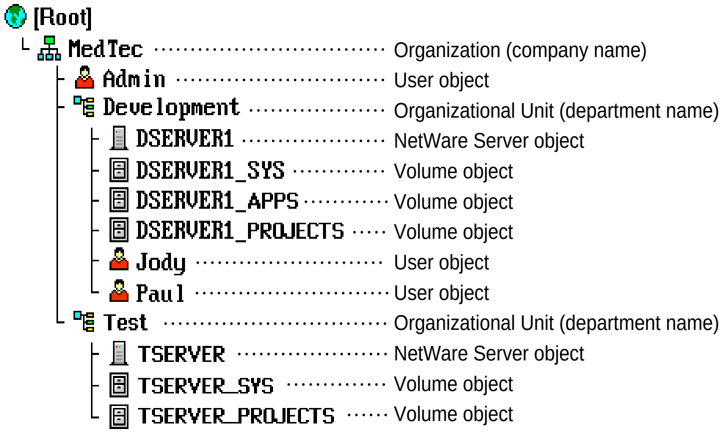


Figure 2-3 **Objects Formed from [Root] in a Directory Tree**

[Root] Object

The [Root] object represents the name of the Directory tree. It resides at the top of the tree and branches downward. Once the [Root] object is named, you reference that object by its given name.

The [Root] object can be created only by the NetWare Directory Services installation program, which automatically places it at the top of the tree. Once the [Root] object is named, it cannot be renamed or deleted.

NOTE: The [Root] object of a Directory tree should not be confused with the root directory in the file system. In the file system, the root directory is the first directory on a volume. It bears no relation to the [Root] object of a Directory tree.

The Directory tree name or [Root] object can have trustees, and rights granted to these trustees flow down the tree. A trustee is an object that is granted rights to work with another NDS object or with components of the file system, such as directories or files. One example is the User object ADMIN, which is created automatically during installation.

By default, ADMIN receives a trustee assignment including the Supervisor right to the [Root] object of the Directory tree. This gives ADMIN all rights to all objects and properties in the tree. Thus, ADMIN is used to initially log in and set up the tree. See ”User Object ADMIN” in chapter 3 for more information.

The [Root] object can also be a trustee. Most likely, however, you will not assign trustee rights to the [Root] object. If you do, every object in the tree has the same rights as the [Root] object by virtue of inheritance. In effect, you assign every user that logs in rights to the [Root] object. See “Security Equal To” in this chapter for more information.

Container Objects

Container objects hold (or contain) other Directory objects. Container objects are a means of logically organizing all other objects in the Directory tree. Just as directories are used to group related files together in a file system, container objects are used to group related objects in the Directory tree.

A container object that contains other Directory objects is known as a parent object.

There are four types of container objects, defined as follows:

- **Country (C)**. The Country object designates the countries where your network resides and organizes other objects within the country. Country objects can be placed only immediately below the [Root] object and their names are limited to two characters.

  Country objects are optional. They are typically used only if your organization spans multiple countries or if you must include a Country object to interact with other X.500 specification compliant directory services. For more information, see”NDS and the X.500 Specification” in this chapter.

  You can use a Country object to designate the country where your organization headquarters reside or, if you have a multinational network, to designate each country that is part of your network.

  Because of the following considerations, you should plan to use a Country object only if your environment requires one:

  Country objects add an extra organizational level to the name of each object in your Directory tree. For more information, see”Context and Names” in this chapter.

  Country objects require you to use typeful names instead of typeless names when referring to contexts within your Directory tree.

- **Locality (L)**. The Locality object designates the location where this portion of your network resides and organizes other objects within the location.

Locality objects are optional. You can use them to designate the region where your organization headquarters reside or, if you have a multinational network, to designate each area that is a part of your network.

Locality objects can reside under Country, Organization, and Organizational Unit objects. They can also hold Organization and Organizational Unit objects.

NOTE: The Locality object is not part of the NetWare default server installation. You can create a Locality object during the server installation.

- **Organization (O)**. An Organization object helps you organize other objects in the Directory tree. It also allows you to set defaults for User objects you create in the Organization object.

  You can use an Organization object to designate a company, a division of a company, a university or college with various departments, a department with several project teams, etc.

  Every Directory tree must contain at least one Organization object.

  Organization objects must be placed directly below the [Root] object, unless a Country or Locality object is used.

- **Organizational Unit (OU)**. An Organizational Unit object helps you organize leaf objects in the Directory tree. It allows you to set defaults in a login script and to create a user template for User objects you create in the Organizational Unit.

  You can use an Organizational Unit object to designate a business unit within a company, a department within a division or university, a project team within a department, etc.

  This object is optional. When used, Organizational Units must be placed directly below an Organization, another Organizational Unit, or a Locality object.

Leaf Objects

Directory leaf objects are objects that do not contain other objects. These represent actual network entities such as users, servers, printers, computers, etc.

You create leaf objects within a container object. The following figure lists the leaf objects you can create. (The icons represent the leaf objects as they appear in the NetWare Administrator graphical utility.) Please note that the AFP Server object is not supported in NetWare Services.

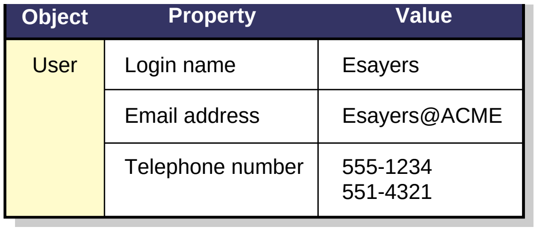
See appendix B for more information.

Object Properties

Each type of object (such as a User object, Organization object, or Profile object) has certain properties that hold information about that object. For example, a User object's properties include a login name, E-mail address, password restrictions, group memberships, etc. A Profile object's properties include profile name, login script, and volume.

Some objects require values for specific properties before setup of that object is complete. Other properties are optional and can be added later as the need arises.

The following figure shows the relationship between object, property, and value.



Object	Property	Value
User	Login name	Esayers
	Email address	Esayers@ACME
	Telephone number	555-1234 551-4321

In many cases, you can enter more than one value for a property. For example, you could enter a home, mobile, and work telephone number for a User object.

NetWare utilities allow you to search for objects that have specific property values. For example, you could search for all users who have a certain area code in their telephone number. The utility returns a list of all the objects with that area code in their telephone number property.

You can also request information on a specific object. The utility searches only for that object, and you receive information on that object's properties, provided you have the appropriate access rights.

To make searching for an object property easier, you can enter information for the optional properties when you create objects. This information can help you track and manage those objects.

Also, if you create objects or assign property values using a consistent format, you can use the NetWare Administrator, NETADMIN, or NLIST utilities to search for objects or property values.

For example, you might want to search for all User objects at a certain location, such as building M1. You cannot easily list all objects located in building M1 if you have entered “Bldg. M1,” “M1 Bldg,” and “M-1” as values in the Location property of various User objects.

Standardizing the value for the Location property for all User objects at the site (such as M1, M2, and M3) makes it easier to search for objects located in each building.

Object and Property Rights

NetWare 4™ software uses four different categories of rights:

- File-system directory rights
- File-system file rights
- NDS object rights
- NDS property rights

Previous versions of NetWare had file-system directory and file rights and some access control to bindery objects in bindery-based NetWare networks. NetWare 4 replaces bindery control with NDS object and NDS property rights. These rights determine what you can do within the Directory tree.

Because the Directory tree is a hierarchical structure, rights assigned in the Directory tree flow down through the tree. This is an important concept to understand and consider when designing your Directory tree.

The concept of rights flowing down through the tree is referred to as inherited rights. This functionality is controlled by the Inherited Rights Filter (IRF). An IRF is a list of rights that can be assigned to objects in containers beneath a parent container within the tree hierarchy. It controls the rights that a trustee can inherit from container objects. See “Inherited Rights Filter, NDS Object” in *Concepts* for more information.

To allow you to better control access to NDS objects and their properties, object and property rights are assigned separately.

Object Rights Rights that control access to an object as an entity are called object rights. Object rights control what trustees of an object can do with that object. Object rights do not allow the trustee to access information stored in that object’s properties unless the trustee has the Supervisor object right, which includes the Supervisor property right.

The following table describes object rights you can assign to a trustee.

NOTE: All object rights of a subordinate object can be blocked by an Inherited Rights Filter (IRF) initiating at the point where the object right is granted.

Table 2-2 **Object Rights**

Right	Description
Supervisor	Grants all rights to the object and to all its properties.
Browse	Grants the right to see the object in the Directory tree. Also allows a user performing a search to see the object if it matches the search value. (This is true only when comparing the base object class or Relative Distinguished Name; otherwise, the Compare right is required.)
Create	Grants the right to create a new object within a container object in the Directory tree. This right applies only to container objects, because leaf objects cannot contain other objects.
Delete	The Write right for all existing object properties is also needed to delete objects. Grants the right to delete an object from the Directory tree. However, a container object cannot be deleted unless all the objects in the container are deleted first.
Rename	Grants the right to change the Relative Distinguished Name of the object, in effect changing the naming property. This changes the object’s Distinguished Name. See “Object Naming Rules” in this chapter..

Property Rights To see the information in an object’s properties, trustees must have the correct property rights. Property rights control access to each property of an object.

Keep in mind the distinction between object rights and property rights. While object rights control access to an object as an entity, property rights control access to the information stored as an object’s property values. The only exception is the Supervisor object right. The Supervisor object right includes the Supervisor property right.

Property rights apply only to NDS object properties (and their values), not to the objects themselves. NDS allows you flexibility in deciding what property information others can access.

For example, if you include a telephone number as a property for a User object, you can prevent others from seeing the value associated with that property–that is, the actual telephone number–by using an Inherited Rights Filter to disable the Read right to that particular property (see “Inherited Rights Filter” in this chapter). At the same time, you can still allow the person to view other properties and their values, such as the user’s address.

The following table describes property rights you can assign to a trustee.

Table 2-3 **Property Rights**

Right	Description
Add or Delete Self	This right is included in the Write right; that is, if the Write right is given, Add or Delete Self operations are also allowed. This right is only used for properties where your User object can be listed as a value, such as group membership lists or mailing lists. Allows you to add or remove yourself as a value of the property, but you cannot change any other values of the property.
Compare	Allows you to compare a value with the existing value of the property. The comparison can return True or False, but cannot give the value of the property.
Read	Allows you to read the values of the property. This right includes the Compare right; that is, if the Read right is given, Compare operations are also allowed.
Supervisor	Gives you all rights to the property. The Supervisor property right can be blocked with an Inherited Rights Filter. See “Inherited Rights Filter” in this chapter for more information.

Table 2-3 **Property Rights**

Right	Description
Write	The Write right to the Access Control List (ACL) property is the same as giving the Supervisor right to the object—it allows you to grant rights. This right includes the Add or Delete Self right; that is, if the Write right is given, Add or Delete Self operations are also allowed. Allows you to add, change, or remove any values of the property.

Access Control List The information about who can access object properties is stored in a property known as the Access Control List (ACL). An object's ACL lists all trustees of the object. The ACL property also stores the object's Inherited Rights Filter.

To modify a trustee's access to an object, you change the trustee's entry in the object's ACL. Only trustees with the Write right for the object's ACL property can change trustee assignments or the Inherited Rights Filter.

Each trustee listed in an ACL can have different rights to that object's properties. For example, if ten users are listed in a Modem object's ACL as trustees, each of those ten users can have different rights to that Modem object and to its properties. One trustee might have the Read right, another might have the Delete right, etc.

See "Access Control List (ACL)" in *Concepts* for more information.

Inherited Rights Filter While trustee assignments grant access to an object, the Inherited Rights Filter (IRF) prevents rights from automatically flowing from a container object to the objects it contains.

In the Directory tree, a child object automatically receives, or inherits, rights granted to its parent objects. The IRF can be used to block any or all of these inherited rights so that no child objects receive them.

Through inheritance, every object and property in the Directory tree can have an Inherited Rights Filter.

See "Inherited Rights Filter, NDS Object" in *Concepts* for more information.

Security Equal To The Security Equal To property lists other objects that you want a given object to have security equivalence to. The object is granted the same rights the objects in its list are granted, both to NDS objects and to files and directories.

Use the Security Equal To property to give a user access to the same information or rights another user has access to.

When a user is added to the membership list of a Group object or to the occupant list of an Organizational Role object, the Group or Organizational Role is listed in that user’s Security Equal To list.

By using the Security Equal To property, you avoid having to review the whole directory structure and determine which rights need to be assigned to which directories, files, and objects.

See “Security Equivalence” in *Concepts* for more information.

Effective Rights The combination of inherited rights, trustee assignments in an ACL, and the Security Equal To property are known as effective rights.

An object’s effective rights control its access to another object and that object’s properties.

See “Effective Rights” in *Concepts* for more information.

Context and Names

In NetWare Directory Services (NDS), context refers to the location of an object in the Directory tree. Context is important because NDS objects are identified by their relative location in the Directory tree.

The complete context, or path, from an object to the [Root] of the Directory tree in addition to the object's common name forms an object's Distinguished Name (also called the complete name). The context, or path, from an object to another object in the Directory tree forms that object's Relative Distinguished Name (RDN).

For example, in Figure 2-5, the following is true:

- The context for the User object ESAYERS is OU=DESIGN.OU=LONDON.OU=MFG.O=ACME.C=US
- The Distinguished Name for User object ESAYERS is CN=ESAYERS.OU=DESIGN.OU=LONDON.OU=MFG.O=ACME.C=US.
- The context for the User object RJONES is OU=HR.OU=HQ.O=ACME.C=US
- The Distinguished Name for the User object RJONES is CN=RJONES.OU=HR.OU=HQ.O=ACME.C=US.
- The Relative Distinguished Name for the User object RJONES in relation to the Organizational Unit SALES is CN=RJONES.OU=HR.OU=HQ.OU=SALES.

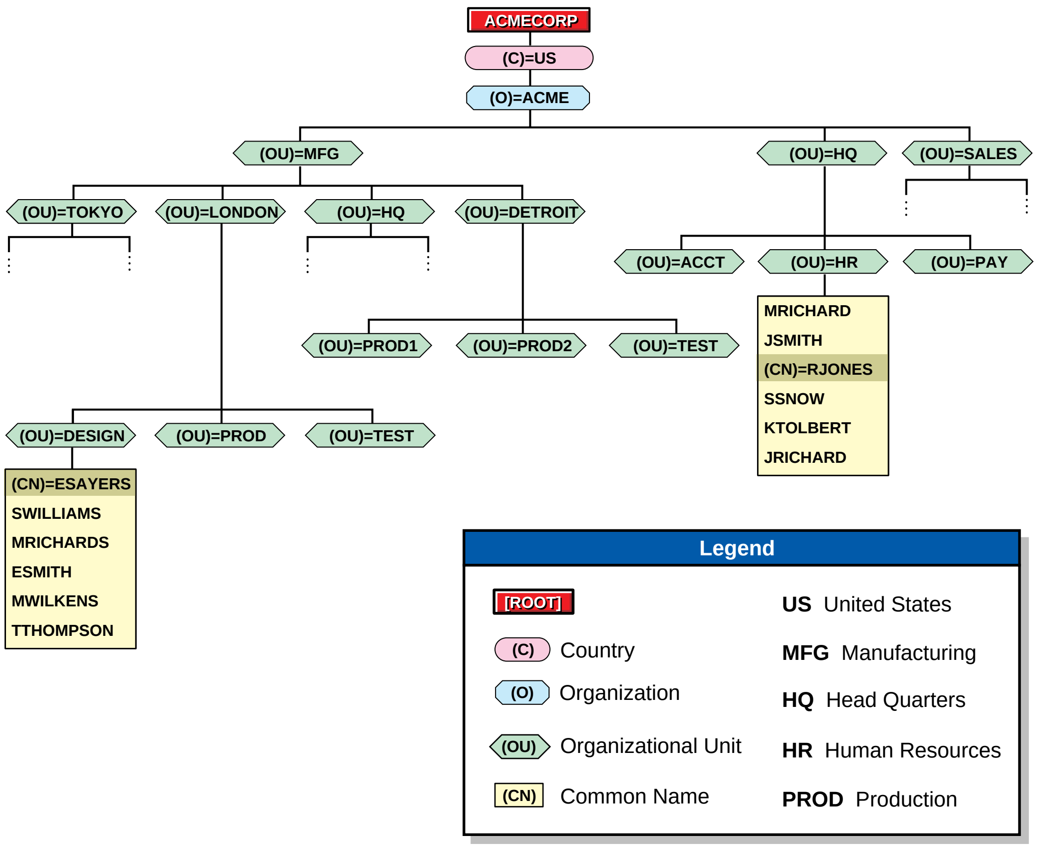


Figure 2-4 **Context in a Directory Tree**

Because names and contexts can be confusing for users, consider the following guidelines:

- Limit the levels of container objects you have in your Directory tree.

  Because it is difficult for some users to remember long Distinguished Names with multiple layers of Organizational Units, you might choose to have no more than two or three levels in your Directory tree.

- Use short object names.

  Because each object is identified by its location within the Directory tree, use a naming scheme that is both practical and functional for your organization.

For example, name servers for their function within a specific organization, and name printers for their type and location.

- Use Alias objects for accessing objects not in current contexts. Alias objects point to objects that exist elsewhere in the tree.

  For example, if RJONES wants to use Accounting's printer, you can create an Alias object for that printer and put it in RJONES' context.

  This way, RJONES can find the printer in his own context, and he doesn't have to remember the longer "real" name of that printer.

- Avoid using spaces in names.

  Spaces in object names appear as underscores in some utilities.

  In other utilities, you might have to enclose the name in quotation marks to avoid having the utilities treat the two-word name as two separate commands or objects.

Common Names

All leaf objects in the Directory tree have a common name. For User objects, the common name is the login name displayed in the Directory tree. For example, the common name for Edwin Sayer's User object is ESAYERS.

Other leaf objects also have common names displayed in the Directory tree.

See "Common Name" in *Concepts* for more information.

Name Types

Names in the Directory tree have two name types: typeful and typeless. A typeful name includes the name type (OU, O, etc.) of each object in the Distinguished Name of an object. A typeless name excludes the name type for each object.

A name type distinguishes the specific object you are referring to, such as a User object or an Organizational Unit container object. For example, the following typeless name

```
ESAYERS.DESIGN.LONDON.MFG.ACME.US
```

is expressed with name types as

```
CN=ESAYERS.OU=DESIGN.OU=LONDON.OU=MFG.O=ACME.C=US
```

where CN is the common name of the leaf object, OU is the Organizational Unit name, O is the Organization name, and C is the Country.

In most cases, you do not need to use name types.

Any time you move from one container object to another, you change context. Whenever you change contexts, you might need to indicate the Distinguished Name of the object you are changing context to.

If you are referring to an object in the same container as your User object, you need only refer to the object by its common name.

NOTE: All Distinguished Names should be unique within a Directory tree. In addition, all object names should be unique within a container. The NDS database recognizes only one occurrence of the same name within each container.

Logging In and Authentication

The location of an object within the Directory tree, or name context, is also important when logging in. When a user logs in to the network, an available server begins a process called authentication.

Based on the current context and the login name provided, authentication identifies the User object to other servers in the tree and verifies that the object has rights to use network resources.

Authentication allows a user who has logged in to the network to access any servers, volumes, printers, etc., in the network that the user has rights to. Conversely, if the users lacks rights, access is denied.

Authentication checks a user’s rights to both NDS and file-system resources. This is one way you, as a network supervisor, can regulate security.

Authentication works in combination with the Access Control List to provide network security. See “Property Rights” in this chapter for more information.

Also see “Name Context” and “Authentication” in *Concepts* for more information.

Object Naming Rules

Apply the following rules when naming NDS objects:

- The name should be unique in the branch (container) of the Directory tree where the object is located.
- The name can be up to 64 characters in length.
- You can use special characters. But, if the object needs to be accessed from a workstation running the NetWare Client shell (NETX), you should avoid using special characters.

  For a list of these special characters, see “Naming Restrictions for Bindery Services” in this chapter.
- Object names are displayed with uppercase and lowercase letters as they are first entered, but they are not case-sensitive. Therefore, “ManagerProfile” and “MANAGERPROFILE” are considered identical names.
- Spaces and underscores can be used and are displayed as spaces. Therefore, “Manager_Profile” and “Manager Profile” are considered identical names.

  If you use a space in a name, you must place quotation marks around that text string whenever you use a command line utility that includes that text string. For this reason, spaces are not recommended.
- Country objects can have only two-character names. For example, the United States is US.

CAUTION: If you anticipate managing objects created from different code pages, you must limit object names and properties to those characters common to all the applicable code tables.

Nondisplayable Unicode* characters for your code page are represented by an ASCII 3 character (a “heart” symbol). For more information, see “Unicode” in *Concepts*.

Naming Restrictions for NetWare Server Objects

The following restrictions apply when naming Server objects:

- When you install NetWare 4, an NDS NetWare Server object is created for the server in the container object you specify.
- If you create a Server object for a server other than a NetWare 4 server, you must use the server name for the object, because NDS searches for the server to verify its existence.

For more information on NetWare Server objects, see “Object” in *Concepts*.

Naming Restrictions for Bindery Services

When you create objects to be accessed from workstations running the NetWare Client shell (NETX), the names of the objects must follow bindery naming rules or these clients cannot recognize them. Object names in bindery services are interpreted as follows:

- Spaces in object names are replaced by underscores
- Object names are cut off after the 47th character

You cannot use the following characters in an object name that must be accessed from a workstation running the NETX client:

/ slash

\ backslash

: colon

, comma

* asterisk

? question mark

NOTE: The object naming rules apply to most objects. Additional rules applying to NetWare Server objects and objects viewed through bindery services are described in a separate chapter. See chapter 4 for more information.

Naming Restrictions for International Support

Unicode is a wide character encoding scheme that provides the basis for internationalization of the information in an NDS database. All character strings exchanged between an NDS server and a client workstation are in Unicode. The NetWare Client software handles the translation of Unicode strings.

Occasionally, however, you might use characters that Unicode cannot translate. When this happens, the character is substituted in your display as a “heart” symbol in DOS and as a box (q) in Windows.

Substituted characters can prevent NDS from recognizing an object. See “Unicode” in *Concepts* for more information.

Where to Go from Here

If you want to	Go to
Use the management features included with NDS	Chapter 3 “Understanding Management Features”
Use bindery services with NDS	Chapter 4 “Understanding Bindery Services”
Use time synchronization with NDS	Chapter 5 “Understanding Time Synchronization in NDS”
Plan, manage, and implement NDS	Chapter 6 “Planning, Implementing, and Managing”

3

Understanding Management Features

Overview

This chapter describes the management features provided by the NetWare® Directory Services™ (NDS) technology on your network.

The following topics are discussed on the indicated pages:

Managing the NDS™ architecture includes creating and managing objects and distributing Directory partitions and replicas.

Management utilities are provided to build and maintain the Directory tree hierarchy and objects and to help you maintain the NDS database on your network.

User Object ADMIN

The first time you log in to a new Directory tree, you log in as the User object ADMIN—the only User object created during the NetWare 4™ installation process. The ADMIN object is created when you first set up a Directory tree but not when you later add other servers to an existing tree.

The ADMIN object is assigned all rights (including the Supervisor right) to every object and property in the Directory tree. This gives ADMIN complete control of the Directory tree.

NOTE: When your first log in to a new Directory tree, you may want to create a User object and assign that object Supervisor rights to ensure that you have more than one object with sufficient rights to completely control the tree. Such an object can be critically important if the ADMIN object is deleted accidentally.

When it is created, ADMIN is assigned the Supervisor object right to the NetWare Server object. This gives ADMIN the Supervisor right to the root directory of all NetWare volumes attached to the server, so ADMIN can be used to manage all directories and files on every volume in the Directory tree.

ADMIN does not have any special significance like that of SUPERVISOR in previous versions of NetWare. ADMIN is granted rights to create and manage all objects simply because it is the first object created.

The following rights are also granted by default to provide basic network functionality:

- The container object where the Volume object SYS resides is granted Read and File Scan rights to the SYS:PUBLIC directory.

  This means that when users are created in that container, they can access all utilities located in the SYS:PUBLIC directory.

  Users outside the container object that holds the SYS volume should be made part of a group with explicit rights to the SYS:PUBLIC directory.

- When created, each User object is granted the Browse object right to the [Root] object.

This means that all users can browse the entire Directory tree.

- When created, User objects are granted the Read right to all properties and the Write right to all login scripts associated with their own User objects.

As User objects are created in the Directory tree, you can grant them the Supervisor object right to selected objects or to entire Directory subtrees.

Other objects that receive the Supervisor object right are allowed to create and manage other container objects and their leaf objects. This allows network control and management to be as centralized or as distributed as you want to make it.

You can rename or delete ADMIN at any time; however, you should assign another User object the Supervisor object right to the [Root] object before you delete ADMIN.

WARNING: **Never delete ADMIN without having assigned the Supervisor right to another User object. Neglecting to do so can be disastrous because you eliminate supervising control of the Directory tree.**

This warning also applies to other sections of the Directory tree where you have an ADMIN object defined. At each level of the tree where you have ADMIN defined, be sure you also have a User object with explicit Supervisor rights.

It is also important to remember that rights can be granted at a container, and they can also be taken away. If all rights are filtered at a container and there is not a user in that container with all rights, then that container is without full administrative rights. This can cause problems.

Directory Partitions

The NDS database can be divided into smaller portions called Directory partitions. Directory partitions are distinct segments of the Directory tree. Directory partitions can be used to decrease possible WAN traffic and to enable more efficient network management.

NOTE: NDS Directory partitions are not related to the logical disk partitions that exist on server hard disks.

Because an NDS database can be separated into partitions located on servers across the network, it is a distributed database.

Partitioning NDS information is completely transparent to network users, making the network look like a single, cohesive collection of resources.

A partition is a subtree or branch of the Directory tree. A partition is named according to the [Root]-most container object within the partition (the one that is closest to the [Root] object).

The [Root] object is always included in the first partition created, which is known as the [Root] partition.

When a partition is subordinate to another in the Directory tree, it is referred to as a child partition. The partition above it is referred to as the parent partition.

The following illustration shows a parent partition in relation to its child partition in a Directory tree.

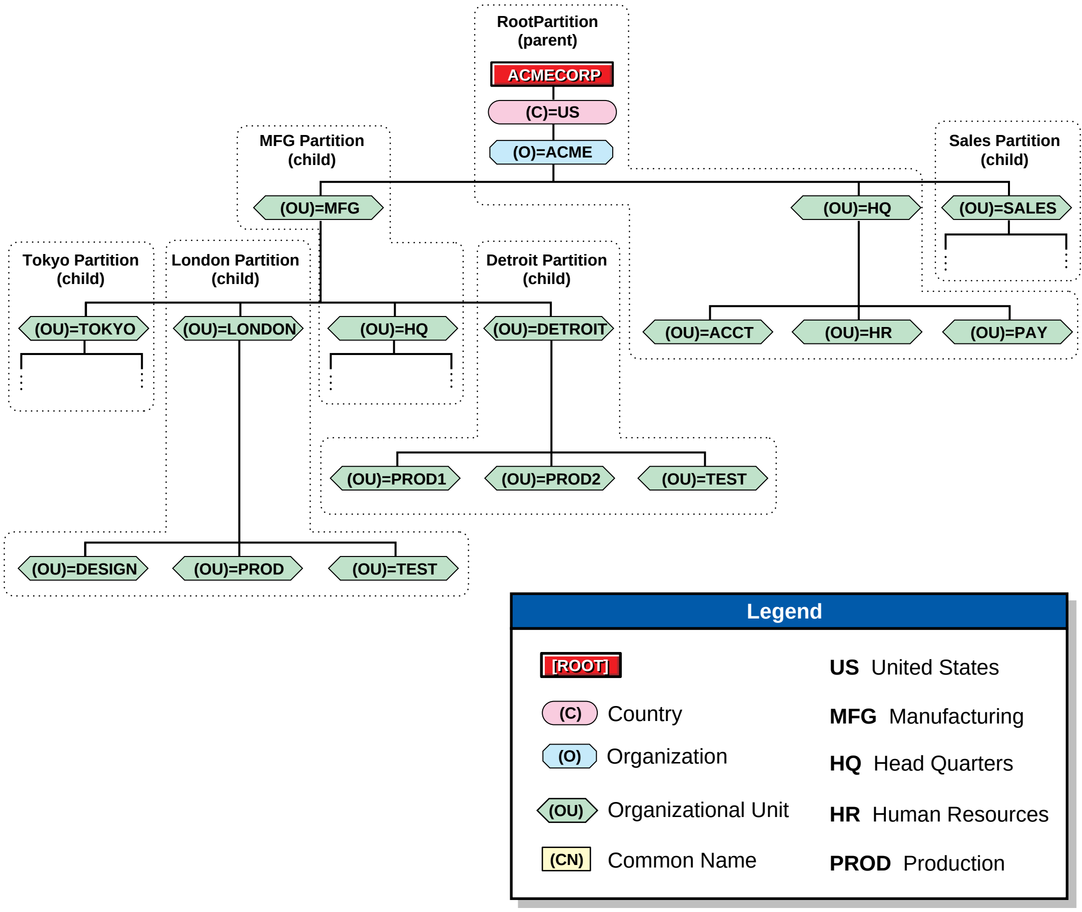


Figure 3-1 **Parent and Child Partitions**

Some characteristics of a Directory partition are as follows:

- A partition contains only NDS objects and related data. It does not include any information about the file system directories and files.
- An NDS object can exist in only one partition, so partitions cannot overlap each other.
- Partitions are stored only on NetWare 4 servers.
- A single NetWare 4 server can store multiple partitions.

Partition Replicas

When you create a partition, you create a master replica of that segment of the Directory tree and database. You can create an unlimited number of additional replicas of the partitions on your network and store them on any NetWare 4 servers on the network.

Purpose

Replicas are created for two reasons:

- Directory Fault Tolerance. If a hard disk crashes or a server goes down, a replica on another server can still authenticate users to the network and provide information about Directory objects.

  With the same information distributed on several servers, you are not dependent on any single server being up to authenticate you to the network or to provide services to you.

  You can store more than one replica on each server.

CAUTION: Directory replication does not provide fault tolerance for the file system. Only information about Directory objects is replicated.

To provide fault tolerance for your files, you must use the host system's fault tolerance system.

- Faster Access Across a WAN Link. If users currently use a WAN link to access some Directory information, you can decrease access time and WAN traffic by placing a replica containing the needed information on a server that users can access locally.

  However, in some cases, WAN traffic could increase due to the NDS synchronization process.

  Distributing replicas among servers on the network allows quick and reliable access because information is retrieved from the nearest available server containing the specified information.

Types

There are three types of replicas.

- Master replica. A writable replica that contains all object information for the partition. All partition operations (create, join, delete, and repair) occur from the master replica of a given partition.

  Only one master replica can be defined for each partition.
- Read/write replica. Contains the same object information as the master replica. Allows modifications (writes) to a partition, which are passed to other replicas of the partition.

  There can be any number of read/write replicas.
- Read-only replica. Contains the same object information as the master replica, but the information can only be read. Used where reading of the partition is required but writes to the partition should not occur.

NOTE: Log in requires a writable replica.

Also, bindery services requires a writable replica. When bindery services is set, use either a master or read/write replica.

See chapter 4, “Understanding Bindery Services” for more information.

Directory Synchronization

When changes are made to objects within a partition, those changes are automatically sent to all other replicas of that partition. This ensures that the global Directory database remains consistent. Only changes are sent to other replicas. For example, if a user changes a phone number, only the new phone number is sent, not the entire User object.

An NDS database is a “loosely consistent” database. As changes occur, all replicas of a partition do not always contain exactly the same information at every instant. In fact, the contents of the replicas most likely vary slightly at any given time. However, these replicas eventually converge to a consistent state once the changes are distributed to all replicas.

Some changes are sent immediately to other replicas, such as changes to a user’s password. Other, less critical changes, such as a user’s last login time, are collected locally for a short period of time before being sent out to the network.

Every partition maintains a record of its replicas. The locations are stored in the partition’s replica property, with one entry for each replica. The collection of replica properties of a partition forms a list of the replicas, sometimes called a replica ring or replica list.

Management Utilities

Management utilities help you build and maintain your Directory tree and objects and help you maintain the Directory database on your network.

See chapter 9, “Managing NetWare Directory Services,” for more information about using the NDS management utilities.

Where to Go from Here

If you want to	Go to
Use NDS on your network	Chapter 2 “Understanding NetWare Directory Services”
Use bindery services with NDS	Chapter 4 “Understanding Bindery Services”
Use time synchronization with NDS	Chapter 5 “Understanding Time Synchronization in NDS”
Plan, manage, and implement NDS	Chapter 6 “Planning, Implementing, and Managing”
Use the management utilities	Chapter 9 “Managing NetWare Directory Services”

4

Understanding Bindery Services

Overview

This chapter describes management procedures for setting up and maintaining bindery services (also called bindery emulation) when you implement the NetWare® Directory Services™ (NDS) technology on your network.

The following topics are discussed on the indicated pages:

Topic	Page
Planning Bindery Services	4-5
Setting a Bindery Context	4-7

Some applications and services which run in the NetWare 4™ environment do not currently take full advantage of NDS™ technology. Novell created bindery services to allow users in these environments access to NetWare 4 services.

With bindery services, NDS imitates a flat structure for leaf objects within an Organization or Organizational Unit object. Thus, when bindery services is enabled, all objects within the specified container can be accessed by NDS objects and by bindery-based servers and client workstations.

CAUTION: Bindery services applies only to leaf objects in the specified container object.

The container object where bindery services is set is called the bindery context. To enable bindery services, you can use the SAM utility or the nwcm command line utility (see “nwcm” in Utilities Reference).

The following figure illustrates bindery services when an Organizational Unit object is specified as the bindery context.

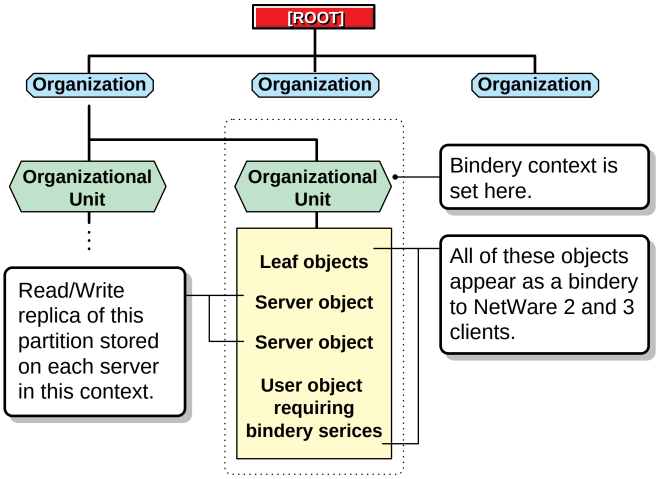

[ROOT]
Organization
Organization
Organization
Organizational Unit
Organizational Unit
Bindery context is set here.
Leaf objects
Server object
Server object
User object requiring bindery serices
All of these objects appear as a bindery to NetWare 2 and 3 clients.
Read/Write replica of this partition stored on each server in this context.

Figure 4-1 **Bindery Services in a Directory Tree**

A writable replica of the partition that includes the container object to be set as the bindery context must be stored on each server you want bindery services enabled on. However, by default, only the first three servers installed on a partition receive a replica of the partition during the installation process and subsequently support bindery services.

You can add replicas to other servers if needed for bindery services. If a read/write or master replica is not present, use the Partition Manager utilities to add one to the server. See chapter 9, “PARTMGR” for information and procedures.

NOTE: If a bindery context is not set, NDS cannot support bindery services.

Bindery services allows NetWare 4 servers to emulate earlier versions of NetWare and is, therefore, server-centric. For instance, if a client workstation requests a bindery login, bindery services directs the default server to use the bindery login script found in the user’s mail directory on the SYS volume instead of using the user’s global NDS login script. Changes to the bindery login script are kept locally and are not distributed to other servers.

You cannot disable bindery services if someone is logged in via bindery services, and bindery objects are always available unless bindery services is disabled.

Planning Bindery Services

When you plan and implement bindery services, you need to consider the following.

Created Objects

Keep these guidelines in mind as you plan bindery services:

- If you require the user GUEST or GROUP EVERYONE or if you use a service that requires GUEST, you must create such a user in the NDS database.
- During installation, a bindery object SUPERVISOR is created but is not used with NDS. The NDS utilities do not display this object. This object is intended to be used with bindery services and to enable access to the server via a bindery login. Once bindery services is enabled, you can use this object to log in to the server, providing you log in as a bindery object.

  You can create an NDS User object SUPERVISOR and assign ADMIN-equivalent rights to it in NDS. However, the bindery object and the NDS object are unique and separate objects even though they are identified by the same name.
- After installing NetWare Services, you can use a migration utility to convert bindery user accounts to NDS User and Group objects. If you do, all users except SUPERVISOR and all groups are updated to NDS objects. The user SUPERVISOR is migrated, but with supervisory rights for that server’s file system and bindery context only. The supervisor does not appear as an NDS object.

Inaccessible Information

Some NDS information is not available to users through bindery services. This information includes, but is not limited to, the following items:

- E-mail name
- Phone number
- Print job configurations
- Aliases
- Profiles
- NDS login scripts

Limited Partitioning

The bindery context for a server can be set to a container that is part of a partition stored on a different server. But, before you can use bindery services, you must place a writable replica of the partition that includes the bindery context on the bindery services-enabled server.

If you set the bindery context for a server to a container object that is not part of a writable replica on that server, users will not be able to log in via bindery services.

Changing Contexts

Avoid changing a server’s bindery context once you set it. Changing a server’s bindery context leaves users in the original context without access to bindery services. Changing the server’s bindery context can also cut off access to print queues.

Setting a Bindery Context

A bindery context is a container object that is specified on each server. You can use either the SAM utility or the nwcm command line utility to create bindery contexts. Only leaf objects in the container that is set as a bindery context are available for bindery services.

Also, any users that will log in via bindery services must have a User object in the container that is specified as the bindery context. However, rather than duplicate User objects in the database, you can place an Alias object in the container that is specified as a bindery context for User objects that aren't ordinarily in that container.

In a Single-Level Directory Tree

If the Directory tree contains only one container level (that is, if the Directory structure is flat), there is only one possible bindery context. For example, the following figure shows a Directory tree with only one level.

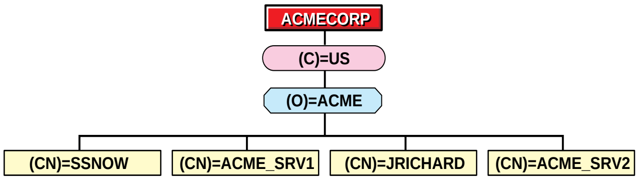


Figure 4-2 **Bindery Context in a Flat Directory Tree**

In effect, this structure is like a bindery and is not fully utilizing NDS. Because there is only one container object, you can set the bindery context on each server to O=ACME. You can do this using either of these methods:

- At the HP-UX prompt, type: **`SAM.`**

  Double click *Networking and Communications* at the SAM main window.

  Double click *NetWare* at the Networking and Communications window.

  Double click *NetWare Directory Services* at the NetWare window.

Type O=ACME in the “Bindery Context” field.

After you have completed the task, exit SAM.

- Use the nwcm command line utility and type the following:

```
nwcm -s ds_bindery_context=”O=ACME”
```

Because the User objects are also located within the O=ACME object, those users can log in to either server under bindery services.

In a Multiple-Level Directory Tree

If the Directory tree contains more than one level, the bindery context has a more noticeable effect on a user’s ability to access bindery services. For example, consider the Directory tree shown in the following figure.

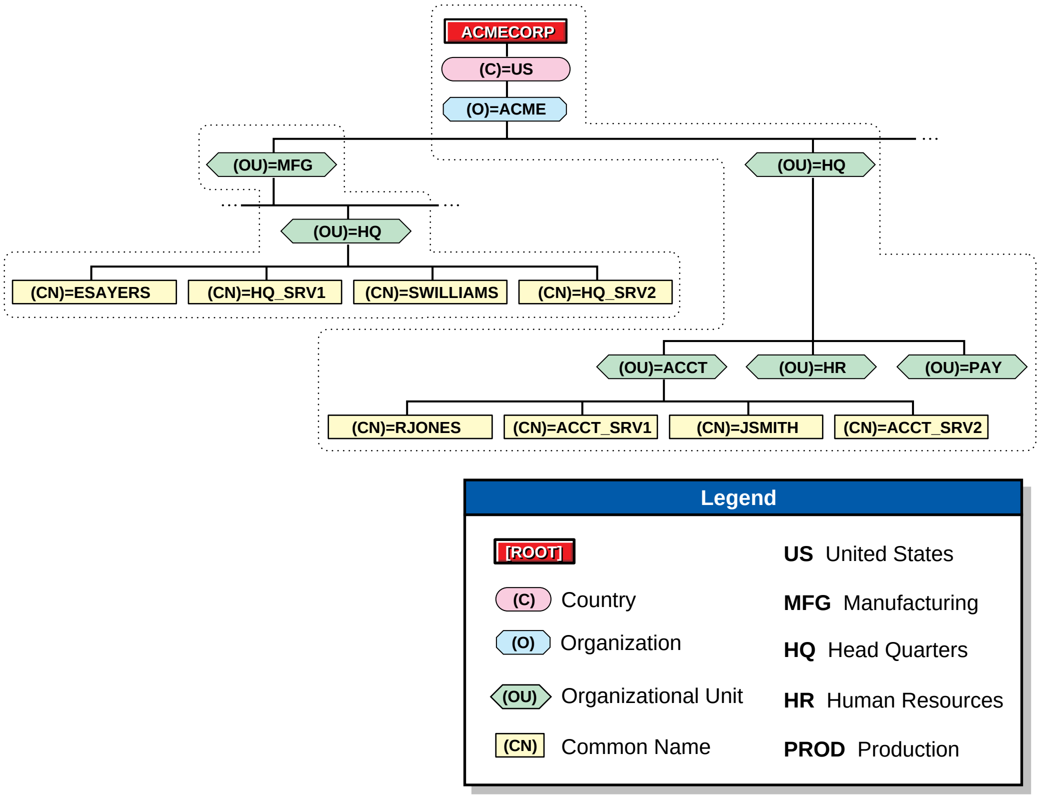


Figure 4-3 Two Different Bindery Contexts in a Directory Tree

This Directory tree has seven container objects, each designated by the name type O (Organization) or OU (Organizational Unit).

NOTE: The following examples use the nwcm command line utility to set bindery contexts. You can also use the System Administration Manager (SAM) utility (see “SAM” in chapter 9).

Suppose the ACME Corporation requires bindery services and sets bindery contexts as follows:

- On the ACCT_SRV1 server, the bindery context is set with the following command:

  ```
  nwcm -s ds_bindery_context=”ACCT.HQ.ACME”
  ```

- On the HQ_SRV1 server, the bindery context is set with the following command:

  ```
  nwcm -s ds_bindery_context=”HQ.MFG.ACME”
  ```

This enables bindery services access to objects in the ACCT.HQ.ACME and HQ.MFG.ACME container objects. Specifically, users in the ACCT.HQ.ACME container can log in as bindery objects and access objects in the ACCT.HQ.ACME container, and users in the HQ.MFG.ACME container can log in as bindery objects and access objects in the HQ.MFG.ACME container.

Now suppose that users in the ACCT.HQ.ACME container no longer need bindery services, but that ESAYERS now requires bindery access to ACCT_SRV1. The bindery context for ACCT_SRV1 can now be set with the following command:

```
nwcm -s ds_bindery_context=”HQ.MFG.ACME”
```

This requires that a writable replica of the MFG partition be stored on the ACCT_SRV1 server. Also, rather than change the bindery context for the ACCT_SRV1 server, you might choose to place an Alias object for the ESAYERS User object into the ACCT.HQ.ACME container.

For a Specific Server

A server’s bindery context can be set to any OU or O that is present in a replica on that server.

For example, given the partitions defined in Figure 4-4, you could set the bindery context of ACCT_SRV1 to any one of the following containers:

- OU=HQ
- O=ACME
- OU=DETROIT

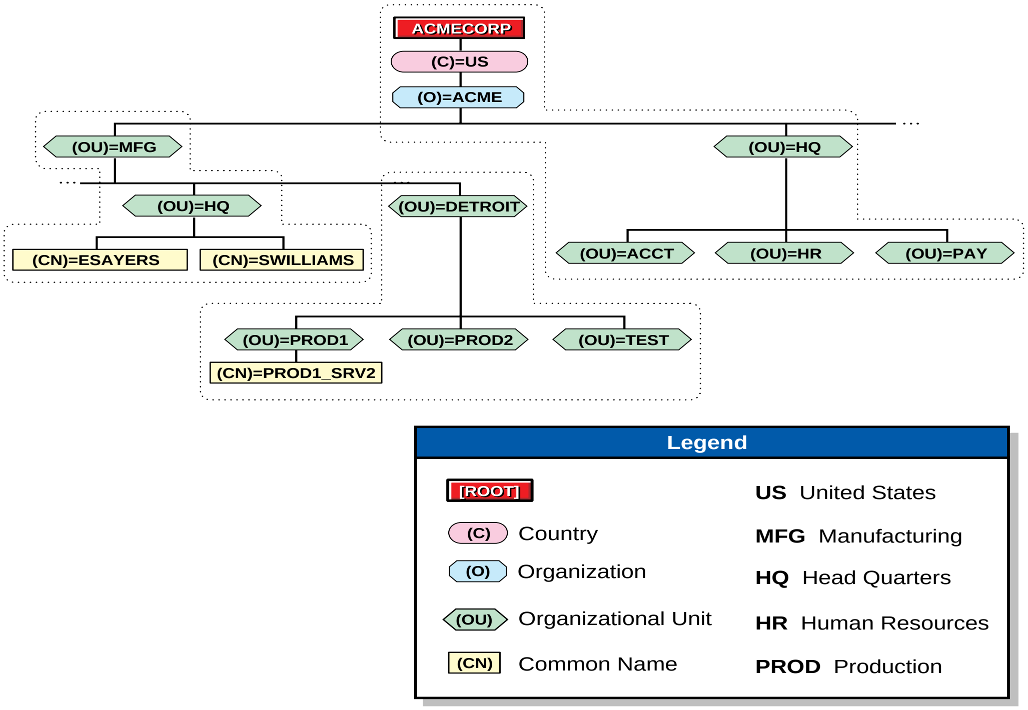


Figure 4-4 **Bindery Contexts for a Specific Server**

This Directory tree represents three partitions of the ACMECORP tree. If there were only one partition, the bindery context could be set to any OU, O, or set of OU and O in the tree. But because multiple partitions exist, any context you set in a different partition must include the path all the way to the [Root] of the tree.

Nevertheless, the bindery context must specify the containers that hold the users that want to log in to that server under bindery services.

For example, suppose you want to set the bindery context for the server PROD1_SRV2 in this tree to OU=HQ so that user ESAYERS can log in to that server with a bindery login. Using nwcm you would type

```
nwcm -s ds_bindery='HQ.MFG.ACME'
```

This command sets the bindery context to the OU=HQ container and provides the path NDS uses to find that container. In this case, the command specifies that the bindery context OU=HQ is contained in OU=MFG.O=ACME.

CAUTION: Be careful when changing a server's bindery context. Removing a container from that server's bindery context prevents all users in that container from using bindery services.

For Multiple Servers in the Same Bindery Context

If a user needs access to several servers, you could use the same container in the bindery context for all of those servers; however, Server objects do not need to be located in their bindery context.

The following figure illustrates how to locate each Server object within the same container as the User object.

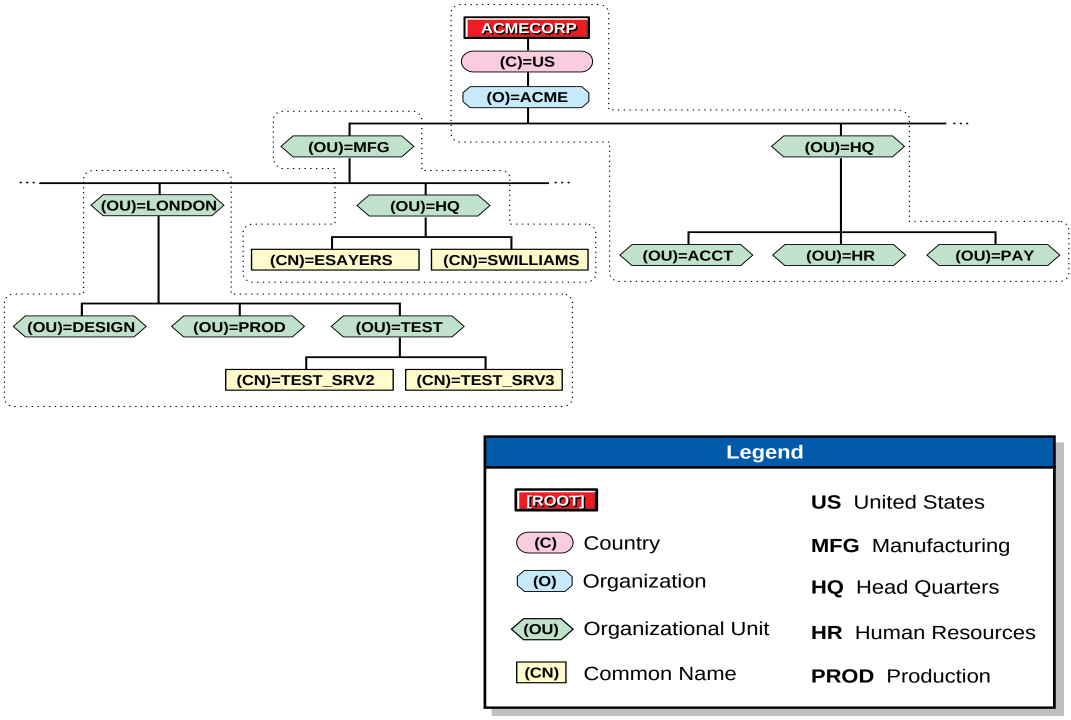


Figure 4-5 **Multiple Servers in the Same Bindery Contexts**

For Objects in Different Bindery Contexts

Ideally, all objects a user wants to access under bindery services should be located in the same bindery context. However, this is not always possible or practical.

You can set multiple bindery contexts for users who need to access objects outside of their own bindery contexts. For example, consider the Directory tree in the following figure.

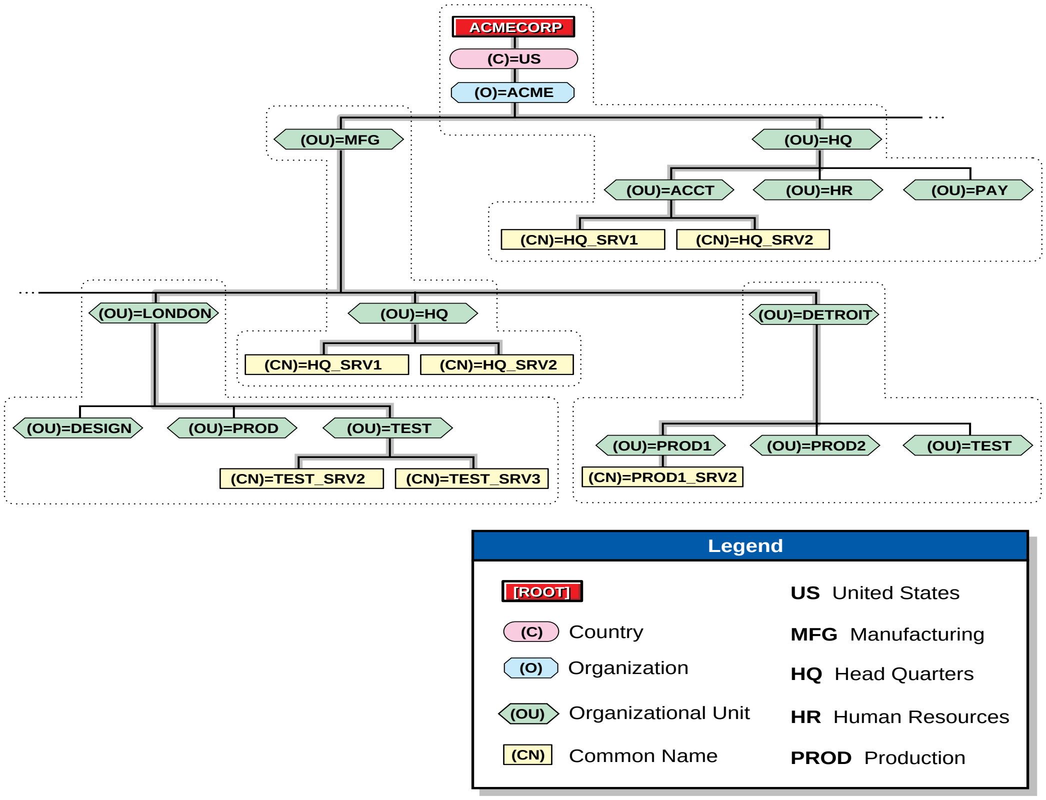


Figure 4-6 **Multiple Bindery Contexts in the Same Directory Tree**

To set bindery contexts on the servers HQ_SRV1 and HQ_SRV2 in this figure, you could use nwcm utility to type

```
nwcm        -s        ds_bindery_context=”ACCT.HQ.ACME;PROD1.
DETROIT.MFG.ACME;TEST.DETROIT.MFG.ACME”;
```

To set multiple bindery contexts, you must set the contexts to include the path all the way to the [Root] of the tree. You can set up to 16 contexts per server.

WARNING: **Do not change a server's bindery context once you set it. Changing a server's bindery context prevents all bindery services users (from the original context) who need to log in to that server from accessing bindery services. Changing the server's bindery context can also disable access to print queues.**

Where to Go from Here

If you want to	Go to
Use NDS on your network	Chapter 2, “Understanding NetWare Directory Services”
Use the management features included with NDS	Chapter 3 “Understanding Management Features”
Use time synchronization with NDS	Chapter 5, “Understanding Time Synchronization in NDS”
Plan, manage, and implement NDS	Chapter 6, “Planning, Implementing, and Managing”

5

Understanding Time Synchronization in NDS

Overview

This chapter describes management procedures for setting up and maintaining time synchronization in an implementation of the NetWare® Directory Services™ (NDS) technology on your network.

The following topics are discussed on the indicated pages.

Time synchronization is important to the operation of NDS™ technology because it establishes the order of events. It is a method of ensuring that all servers in a Directory tree report the same time.

Clocks in computers can deviate slightly, resulting in different times on different servers. Time synchronization corrects these deviations so that all servers in a Directory tree report the same time and provide a time stamp to order NDS events.

For most situations, the default settings for time synchronization are sufficient. However, as you understand the different types of time servers and the process of synchronization, you might decide that your environment will benefit from customization.

Time Stamps

Whenever an event occurs in the Directory database, such as when a password is changed or an object is renamed, NDS requests a time stamp. A time stamp is a unique code that identifies the event and notes the time of its occurrence.

The time stamp is used in the event of collisions (multiple changes to the same object from different servers) on the network to determine the source location and sequence of events.

Time stamps are especially important when Directory partitions are replicated and need to be concurrent with one another.

NDS uses time stamps to

- Establish the order of events (such as object creation and Directory partition replication).
- Record “real world” time values.
- Set expiration dates.

Time Servers

There are four types of NDS time servers: Single Reference, Primary, Reference, and Secondary. During the NDS installation process you are prompted to designate the time server type.

You can also change the time server type after installation by using the System Administration Manager (SAM) utility.

CAUTION: Sometimes the UNIX server is running another, presumably more authoritative, time synchronization protocol (such as NTP).

In this case, the time server type should be set to Reference and the configured sources parameters should be set to” On”). This allows the host time synchronization services to update the UNIX system clock, and allows the NetWare time synchronization services to advertise that clock value to the network.

Each time server type performs a particular time synchronization function, as explained in the following sections.

Single Reference

Single Reference time servers provide time to Secondary time servers and to their own client workstations.

This server determines the time for the entire network. The network supervisor sets the time on the Single Reference time server. (It is possible for the time to be synchronized to an external clock.)

Because the Single Reference time server is the source of time on the network, all other servers must be able to contact it.

The following figure illustrates a Single Reference time server providing time to Secondary time servers and to its own client workstations. The Secondary time servers, in turn, provide time to their own client workstations.

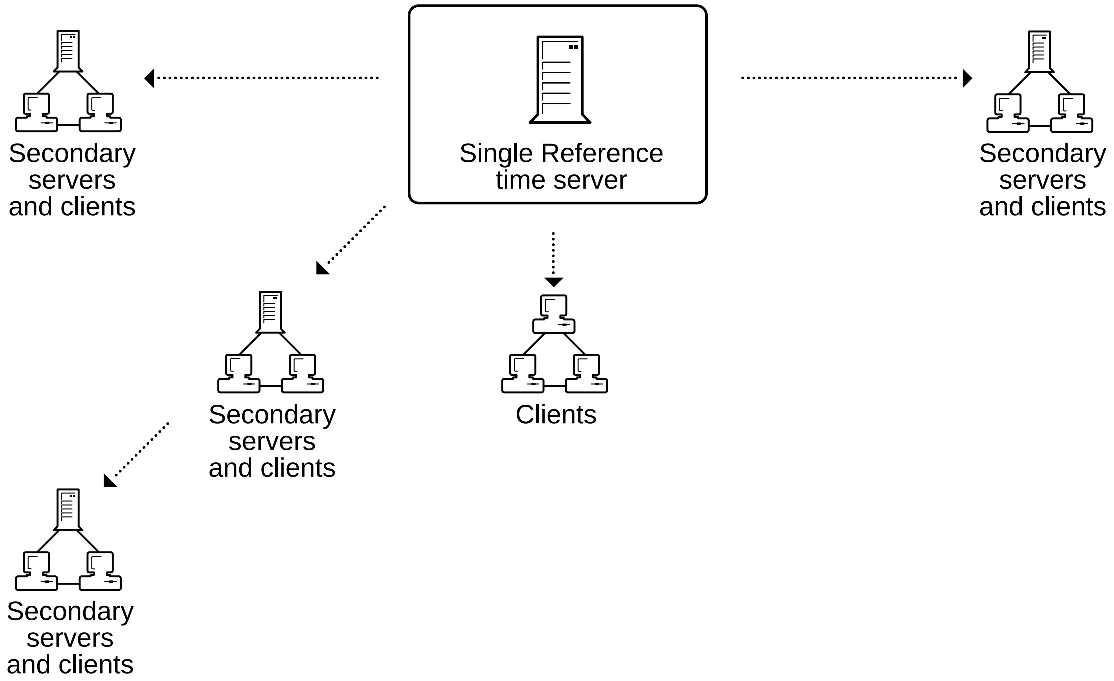

Secondary servers and clients
Single Reference time server
Secondary servers and clients
Secondary servers and clients
Clients
Secondary servers and clients

Figure 5-1 **Single Reference Time Server**

The Single Reference time server works on networks of any size, but the time synchronization configuration shown in Figure 5-1 is used primarily for small networks that don’t include WAN links.

CAUTION: If you use a Single Reference time server, avoid using Primary or Reference time servers in the same tree because the time references might conflict.

Primary

Primary time servers synchronize the time with at least one other Primary time server or with a Reference time server, and they provide the time to Secondary time servers and directly to client workstations.

Primary time servers “poll” other Primary or Reference time servers and “vote” on a common network time. Primary time servers adjust their internal clocks to synchronize with that common network time. Because all Primary servers adjust their clocks, network time might drift slightly.

The following figure shows Primary time servers in various locations providing time to their respective Secondary time servers. Secondary time servers, in turn, provide time to their client workstations.

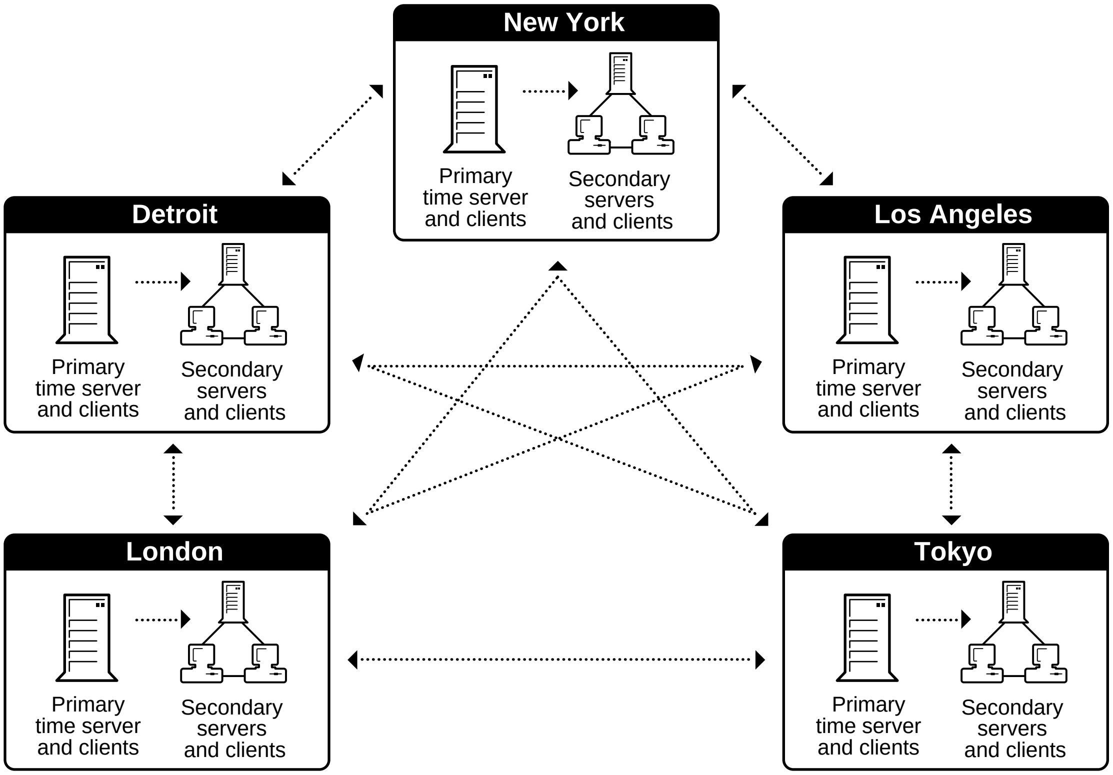

New York
Primary time server and clients
Secondary servers and clients
Detroit
Primary time server and clients
Secondary servers and clients
Los Angeles
Primary time server and clients
Secondary servers and clients
London
Primary time server and clients
Secondary servers and clients
Tokyo
Primary time server and clients
Secondary servers and clients

Figure 5-2 **Primary Time Servers**

You should place a Primary time server in each geographically distinct area so that Secondary servers and client workstations can access them without using WAN links.

Use Primary time servers on larger networks to increase Directory fault tolerance by providing redundant paths for Secondary time servers.

If a Primary time server goes down, the Secondary time server can get the time from an alternate Primary time server.

If you use Primary time servers, each one needs to be able to contact another Primary time server or a Reference time server to determine time on the network.

Reference

Reference time servers provide a time that Primary and Secondary time servers and client workstations can synchronize with.

Reference time servers can be synchronized with an external time source, such as a radio atomic clock.

A Reference time server acts as a central point of control for time on the network. Eventually, all Primary time servers adjust their clocks to agree with a Reference time server.

Reference time servers do not adjust their internal clocks; instead, Primary and Secondary servers' internal clocks are adjusted to synchronize with the Reference time server.

The following figure shows a Reference time server synchronized to an external clock. The Reference time server, in turn, provides time to Secondary servers and client workstations, as well as to a Primary time server at another location.

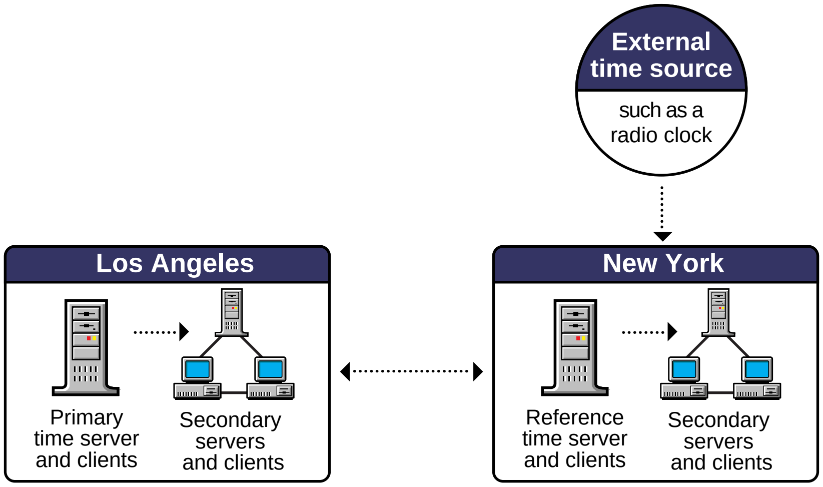

External time source
such as a radio clock
Los Angeles
Primary time server and clients
Secondary servers and clients
New York
Reference time server and clients
Secondary servers and clients

Figure 5-3 **Reference Time Server**

Use a Reference time server when it is important to have a central point of control for time on the network. Usually, only one Reference time server is installed on a network. If you use more than one Reference time server on a network, you must synchronize each Reference time server with the same external time source, such as a radio atomic clock.

You must have at least one Primary time server that the Reference time server can contact in order to synchronize time on the network.

Whenever Primary and Reference time servers are on a network, they must be able to contact each other for polling and synchronization.

Secondary

Secondary time servers obtain the time from a Single Reference, Primary, or Reference time server. They adjust their internal clocks to synchronize with the network time, and they provide the time to client workstations.

A Secondary time server doesn’t participate in determining the correct network time.

Secondary time servers should be close in proximity to Primary or Reference time servers.

For optimal time synchronization, minimize the number of intervening routers and slow LAN segments between Secondary time servers and their Single Reference, Primary, or Reference time server.

Summary

The following table summarizes the types of time servers and their uses.

Type of Server	Function	Cautions
Single Reference time server	Provides time to Secondary time servers and client workstations. Typically used for smaller LANs.	All servers must be able to contact the Single Reference Time Server. No Primary or Reference time servers can be on the network.
Primary time server	Polls and votes with other Primary time servers to determine time, and provides time to Secondary time servers and client workstations. Use with Reference time servers to pass time to Secondary time servers and client workstations.	Must be able to contact at least one other Primary time server or a Reference time server.
Reference time server	Receives time from an external time source and provides time to Primary and Secondary time servers. Use when it is important to have a central point of control for time on the network.	Typically, only one Reference time server is installed on a network. If there is more than one Reference time server, each must be synchronized with the same external time source.
Secondary time server	Receives time from a time source server and provides time to client workstations.	You can have a Secondary time server contact another Secondary time server to obtain the correct time. However, if the intermediate Secondary time server is unavailable, servers that contact it for the correct time might be too many hops away from a time source server to get synchronized.

Time Source Server Functions

The Single Reference, Primary, and Reference time servers are all time source servers. That is, they provide time to the network. Secondary servers do not provide a time to other servers; they only receive a time from a time source server. (They do, however, provide time to client workstations.)

Time source servers use one of two methods to find each other: SAP or custom configuration.

SAP (Service Advertising Protocol)

By default, Primary, Reference, and Single Reference time servers use SAP to announce their presence on the network.

Primary and Reference time servers use the SAP information to determine which other servers to poll in order to determine the network time.

Secondary time servers use the SAP information to choose a time server to follow.

An advantage of the SAP method is that it allows for quick installation without regard to the network layout. It also allows automatic reconfiguration if operating modes are changed or if new servers are added to the network.

CAUTION: The SAP method might generate additional network traffic.

The SAP method might also be disruptive in large network environments where “test” servers come and go, especially if the test server is configured as a time source (Single Reference, Reference, or Primary time server).

Custom Configuration

Custom configuration of your time servers might give you more control over time synchronization, but it requires more planning to synchronize servers efficiently.

An advantage of custom configuration is that you maintain complete control of the time synchronization environment.Also, custom configuration might help eliminate nonessential network SAP traffic, as well as errors associated with accidental reconfiguration.

To customize your time servers, you can use either System Administration Manager (SAM) or the nwcm command line utility to set the following parameters:

- Time Sources. Lists the specific time source servers that a server should contact.
- Configured Sources. Specifies that a server should not listen for SAP information from other time source servers.
- Service Advertising. Disables time source SAP information from being broadcast to the network.
- Directory Tree Mode. Indicates the server should ignore time sources advertising via SAP if the advertising does not originate on the server's Directory tree. This parameter has no effect if the Configured Sources parameter is turned on.

For detailed information on setting these parameters with SAM, see "Managing Network Time Synchronization" in Supervising the Network. For information on nwcm, see "nwcm" in Utilities Reference.

CAUTION: The custom configuration does require additional time for planning and configuration.

It also makes it more difficult to install or remove Primary, Reference, or Single Reference time servers on the network. You must manually change the approved server list maintained on servers that depended on a removed server.

Choosing a Time Synchronization Method

You can use both the SAP and custom configuration methods on the same network. However, the custom configuration information that is stored on the server always takes precedence over the SAP information received by the server.

If a server does not have custom configuration information, SAP information is used for time synchronization.

NOTE: On small networks, where it is unlikely that servers will be added or reconfigured after initial installation, you should use a Single Reference time server using SAP and Secondary time servers. (These are the installation defaults.)

On larger networks, or on networks subject to frequent reconfiguration when servers are added or removed, use a custom configuration.

Where to Go from Here

If you want to	Go to
Use NDS on your network	Chapter 2, “Understanding NetWare Directory Services”
Use the management features included with NDS	Chapter 3, “Understanding Management Features”
Use bindery services with NDS	Chapter 4, “Understanding Bindery Services”
Plan, manage, and implement NDS	Chapter 6 “Planning, Implementing, and Managing”

6

Planning, Implementing, and Managing

Overview

NetWare® Directory Services™ technology requires you to set up a Directory tree on your network. Efficient planning and management can make your implementation simple and easy to do.

Contents

This section is divided into three chapters, with the following information discussed on the indicated pages:

7

Planning NetWare Directory Services Implementation

Overview

This chapter provides instruction for planning an implementation of the NetWare® Directory Services™ (NDS) technology on your network.

The following topics are discussed on the indicated pages:

The size of your network determines the amount of planning necessary for implementing the NDS™ technology—the larger the network, the more planning might be required.

A small network implementation of a Directory tree with only one container object needs minimal, if any, planning of the Directory tree structure.

A large network with thousands of users, hundreds of servers, hundreds of printers, and dozens of network supervisors in various departments benefits greatly from advanced planning of the Directory tree structure.

Regardless, NDS makes all of your network resources available in one information system, with an overall strategy for consistent and logical organization of network resources.

Efficient planning enables your Directory tree to

- Make looking up information easier for users
- Make administering the network easier for network supervisors
- Provide fault tolerance for the Directory database
- Decrease traffic on the network

To plan an implementation of NDS, consider the following issues:

- What organizational structure of the Directory tree makes the most sense for your network resources?
- How do you want the Directory database to be partitioned, and where do you want to store replicas of those partitions?
- How should time be kept and synchronized among the servers on the network?

Although planning is important to a successful implementation, NDS does allow for subsequent changes to the Directory tree structure. NDS is flexible and has been designed to allow restructuring as the structure of your organization changes.

Guidelines for Implementing NDS

You can design a Directory tree several different ways. You might want to develop different prototypes and test them in a lab environment to analyze the advantages and disadvantages of your design.

Nevertheless, the necessary steps for implementing NDS are simple and remain essentially the same for small, medium, and large networks of any design.

Some of the following guidelines are not necessary for smaller implementations of NDS; however, all of these guidelines can assist you in planning for any present and future implementations.

To implement NDS on your network, you need to complete the following tasks:

1 Identify all potential Directory objects and create a NetWare Directory Services standards document that details how to name objects (Users, Printers, Servers, etc.) and how to name object property values, such as telephone numbers.

You can distribute this document to network supervisors who are responsible for adding or moving objects in different parts of the Directory tree.

You should use short names within the hierarchy because each object is identified by its location within the Directory tree. Use a naming scheme that is both practical and functional for your organization. For example, name servers for their function within a specific organization, and name printers for their type and location.

See Appendix C “NDS Object Classes and Properties” for more information.

See also”Creating Container Objects,” “Creating Leaf Objects” and “Searching for Objects” in *Supervising the Network*.

2 Plan your Directory tree from the top, or [Root] level, down to the branches.

See “Planning an Organizational Directory Tree” in this chapter.

3 Organize objects into a logical hierarchy.

The hierarchy of your Directory tree should be as shallow as possible (three to five levels) to facilitate access and manageability. However, NDS supports any degree of subordination you need to best support your organization’s infrastructure.

4 Decide on the model for your Directory tree.

Your Directory tree can model your organization, unit, and workgroup breakdown charts, or it can follow administrative, geographical, and functional divisions present within your organization.

See “Creating Directory Tree Maps” and “Placing Leaf Objects in the Directory Tree” in this chapter.

5 Develop strategies for adequate replication of the partitions to

- Provide fault olerance
- Decrease traffic over WAN links

You should plan to divide the Directory database into partitions based on logical boundaries, and replicate those partitions where you want users to access them without having to go over WAN links.

See “Developing a Replication Strategy” in this chapter.

6 Choose a method for providing time synchronization for the Directory. You must designate which time servers you want to use as time source servers.

See “Developing a Time Synchronization Strategy” in this chapter.

7 Develop a strategy for implementing NDS security.

You can use the design of the tree to implement security for containers and the objects in the containers.

See “Developing a Security Strategy for the Directory Tree” in this chapter.

8 (Conditional) Develop a strategy for supporting bindery services.

See Chapter 4, “Understanding Bindery Services.”

First Steps

To begin planning your Directory tree, look first at your organization’s structure, functions, geography, and needs. NetWare Directory Services is designed to reflect a hierarchical structure.

Generally, this means that your Directory tree will be patterned according to some logical structure of your organization or locale, whether or not that structure is formal. Try to simplify the hierarchy as much as possible.

For example, if your organization is formally divided into departments, you might decide to structure your Directory tree by departments as well.

On the other hand, if people in several departments work together on long-term projects and need access to common resources, it might make more sense to divide your tree by project teams instead of departments.

When planning your Directory tree, also consider who will be running the network. With NetWare 4™, you can centralize network administration so that a single person or small group of people control the entire network.

You can also distribute administration so that many network supervisors throughout the enterprise or organization control their own portion of the Directory tree.

If network administration will be distributed, everyone who will be administering the network must be involved in the planning.

See Appendix C, “NDS Object Classes and Properties.”

Creating Directory Tree Maps

We recommend creating two maps of the tree when planning. The first and most important is a logical map of the tree—in other words, names and placement of Organizational Units and other objects.

The second is a physical map of the placement of replicas—in other words, a view of every server and what replicas are stored on each.

The following illustration shows partial examples of these maps.

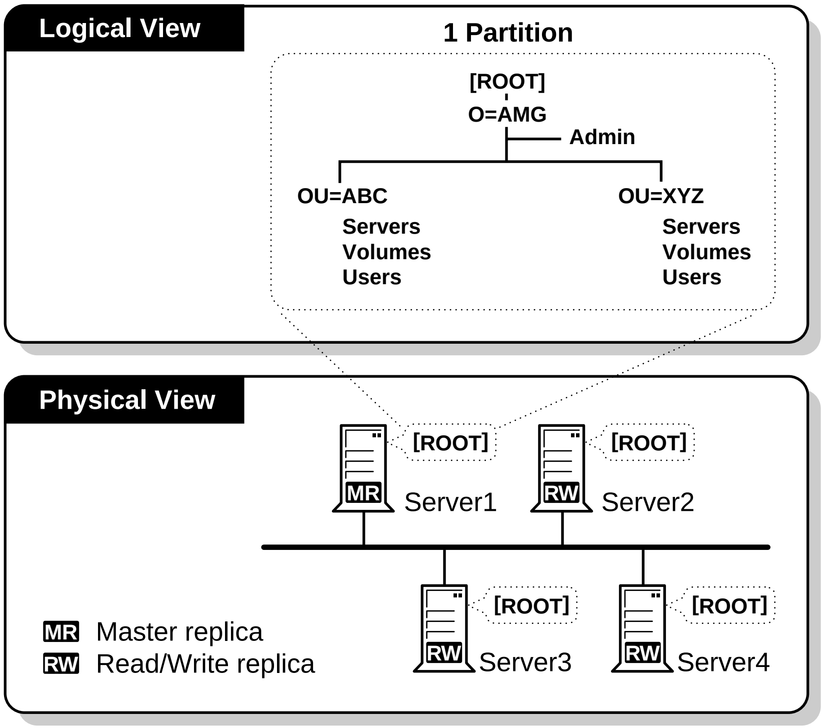


Figure 7-1 **Directory Tree View Maps**

Developing Naming Standards

Part of the process of developing the Directory tree maps is to determine names of objects. If there are standards in place for using NDS, then users can more fully navigate, use, and exploit the Directory tree.

Searching and browsing rely heavily on the ability to do a lookup in the Directory based on criteria from the user. If object names follow a standard, then searching is simpler.

For example, if all laser printers are named “LJuniquename,” where uniquename is more descriptive, then a search for all printers named “LJ*” is feasible.

See Appendix C “NDS Object Classes and Properties” for more information.

NOTE: Use familiar naming conventions, such as users’ E-mail names, to ensure that each user has a unique common name.

Naming standards detail the conventions you will use for naming Directory objects, including users, printers, print queues, and servers. Standards should also specify how you will enter property values (telephone numbers, addresses, etc.) for the objects.

If you will use bindery services, make sure the names are compatible with standards for bindery-based versions of NetWare. (See Appendix C, “NDS Object Classes and Properties.)

Consistency

Consistent naming standards provide a guideline for network supervisors who will be adding file servers, creating users, modifying and moving objects, etc. Consistent standards also make it easy for users to identify the resources available to them in the Directory tree.

NOTE: Although a consistent naming standard for the corporate network is important, you do not need to have it perfected before you implement NDS because leaf objects can be renamed later. See Chapter 9, “Managing NetWare Directory Services. for more information.

Name Length

Make sure naming schemes are short, yet as descriptive as possible. For example, “Software Engineering” could be shortened to “SWEng.”

All Directory object names can contain up to 64 characters in their Name property (the name given when an object is created). The Distinguished Name of an object is limited to 256 characters (including name types, periods, and equal signs).

However, concise (short) Organizational Unit names that are meaningful make it easier to use the Directory tree. Keeping names short reduces the amount of data going across the wire, simplifies logins, and makes names easier to remember.

Planning an Organizational Directory Tree

If your organization is large, you might want to implement an organizational Directory tree. Plan only the top levels, and then allow individual sites to create and administer their parts of the Directory tree.

Consider the following strategies when planning this type of Directory tree:

- After installing the first server into the tree at the organizational level, log in and use the administration utilities to create the next Organizational Unit (OU) levels in the Directory tree. Then, create a User object in each OU with all rights to that container object.
- After creating the User objects indicated above, use a partition utility (PARTMGR or NetWare Administrator) to change each OU into a partition root. However, you should create partitions only if doing so will ease network administration, provide a level of fault tolerance, or decrease WAN link traffic.

  This strategy enables the [Root] level network supervisor to manage the whole network while providing administration duties to supervisors of each partition or local network. Local network supervisors can help build the tree by installing servers in their respective Organizational Units while being administratively restricted to their particular portions of the tree.
- You should use only one Directory tree in your organization so that you can take advantage of the global features of NDS. Rather than create a separate tree for resources you want to deny access to, use NetWare 4 security features to control access to any part of the Directory tree.

Organizing Objects into a Logical Hierarchy

Keeping your Directory tree structure as shallow as possible (three to five levels) benefits both small and large Directory trees. Nevertheless, NDS supports any degree of subordination you need to best support your organization’s infrastructure.

Your Directory tree can model any or all of the following structures:

- Organizational chart structure

  You can begin with your organizational chart, and modify it according to network access requirements and other factors.

- Geographic structure

  You can use geographic locations as Organizational Units. Then, you could use your organizational chart for each location to organize those divisions.

- Functional structure

  If users or groups in your department or organization perform similar functions, consider organizing your Directory tree by function. Such users are likely to share servers and other resources, so it makes sense to group them together.

  This is especially useful for groups of bindery services users.

- Bindery services structure

  The portions of the Directory used by bindery services users should use a combination of all three of the previously mentioned structures.

  Bindery services users should be grouped within bindery contexts defined by workgroups, shared resources, and information usage and exchange.

  Placing similar users in the same container object makes it easier to give bindery services users access to the resources they need.

Planning the Directory Tree Levels

You create container objects to form the top level of the Directory tree for both departmental and organizational strategies. These container objects help you manage and organize the network by relating groups of other container objects and leaf objects.

It is important to remember that the top level is the most important level of the Directory tree. All other levels of the tree branch off the top level. If you organize the top level well, you can organize your entire Directory tree more efficiently.

Consider the following when planning Directory tree levels:

- The name of the Directory tree must be unique on the physical wire or backbone of the actual network hardware connection.
- The depth of the Directory tree should be no longer than 256 characters for the Distinguished Name, which is the full context of the tree.

  Remember that each level you add to the tree can increase the length of a user's context. The shorter you can keep users' contexts, the less problem they will have remembering them.
- Partitions or replicas should be placed close to the end user.

  For example, if there are departments in two cities that access the same resources in the Directory tree, such as printers or servers, then place a replica in both cities to accommodate both departments.
- Rights should be granted by exception. That is, you should grant rights at the container level, then at the group level, and then at an individual object level if necessary.

  For example, if you have a group of users that will generally require the same rights assignment, plan to place them in the same container and assign the rights to the container. Then, if there is a small subset of these users that should not have one of the rights assigned to everyone else, plan to mask the right for those User objects or add those objects to a group that has the right masked.

Placing Container Objects in the Directory Tree

Container objects and their contents should be defined by workgroups, shared resources, and information usage. Use Organization objects and Organizational Unit objects to build the Directory tree structure.

Country and Organization Objects

The Country object, which can be placed only between the [Root] object and your Organization objects, is useful when your network spans more than a single country or when you plan to access information on the global internetwork.

Because the Country object adds another level of complexity to your Directory tree, it is optional and should only be used in the cases previously indicated. See Appendix D, “NDS Object Classes and Properties” for more information.

Organization objects must be placed either directly below the [Root] object or any Country objects you choose to place in your Directory tree. At least one Organization object must be placed in your tree. However, you can create as many sibling Organization objects as you need, and as many Organizational Units underneath the Organization objects as you need, to best structure your Directory tree.

However, because Organization objects must be directory below the [Root] object or a Country object, do not depend on them to fully organize your tree. Use Organizational Unit objects to develop the structure of your tree.

The following figure shows an example tree with a Country object, one Organization object, and multiple Organizational Units.

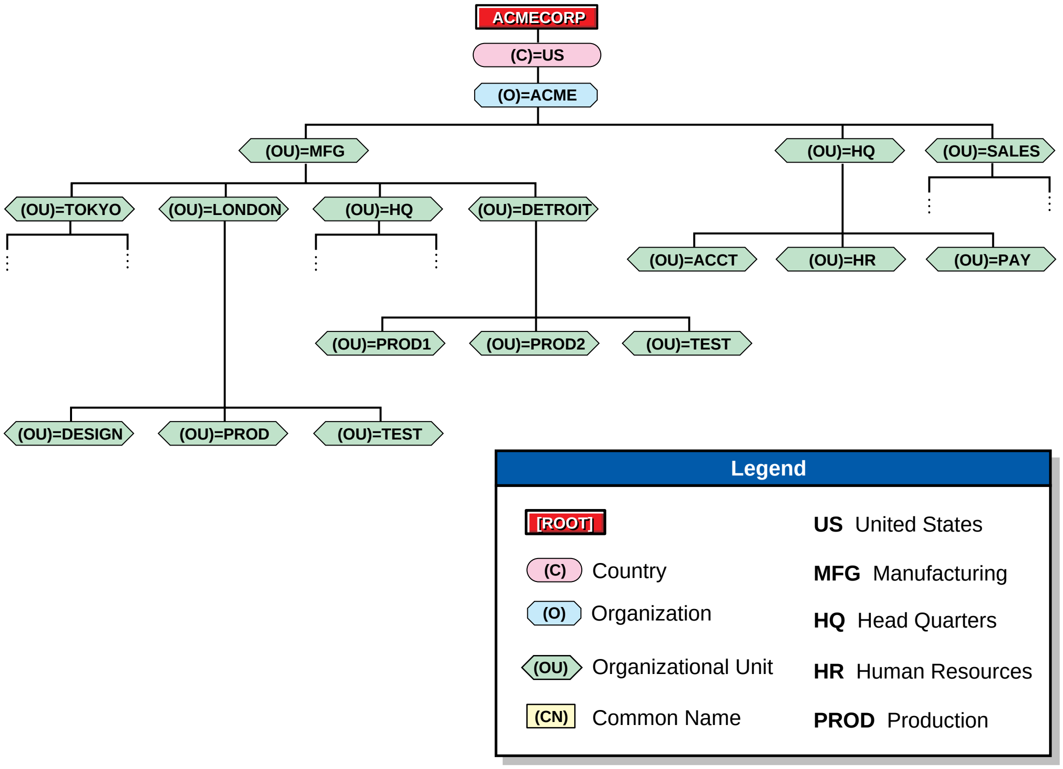


Figure 7-2 **Directory Tree with an Organization Object and Multiple Organizational Unit Objects**

Organizational Unit Objects

You can designate geographic locations, projects, products, etc., as Organizational Units (OU). An advantage to using a geographic structure for your Directory tree is that you can see where objects are physically located. Using geographical locations will assist in the placement of replicas.

Because one goal of having a Directory tree is to provide a static database that is updated infrequently, broad geographic designations in which objects remain static, such as states or cities, might provide a more stable structure for your tree than one that is continually changing.

However, if users or other resources are moved between locations frequently, their contexts can change dramatically even though the organization might not.

The following figure shows an example Directory tree in which the

- Organization object is designated as ACME
- Organizational Units are designated as departments
- Organizational Units at a lower level are designated as geographic locations (Detroit, London, and Tokyo) of those departments

The upper OU level reflects the management organization of the company, and the lower OU level divides the tree into physical locations. This tree is based on data from network administrators at each site, which makes it easy to administer. However, this tree may not facilitate the placement of replicas.

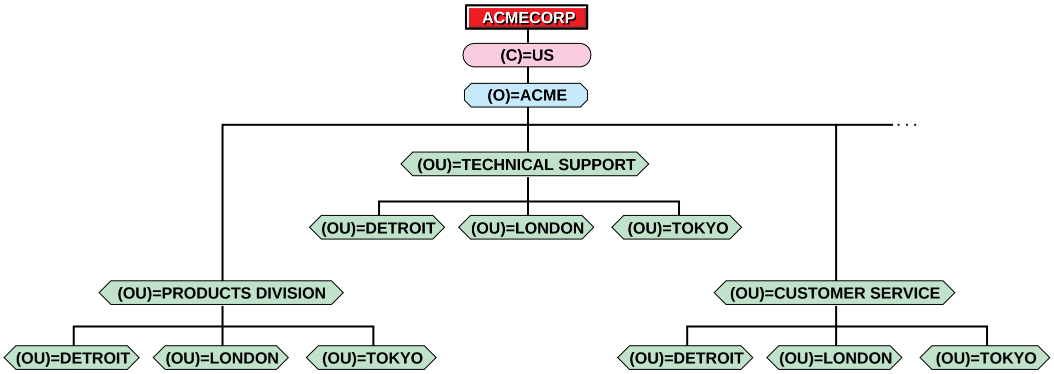


Figure 7-3 **Directory Tree with Organizational Unit Designations at Different Levels**

Organizational Units do not all have to be the same type. That is, you can designate a workgroup as an Organizational Unit and also designate a project as Organizational Unit.

You might want to organize your Directory tree by function if groups of users have the same functionality.

The following figure shows a Directory tree in which MFG (Manufacturing) and HQ (ACME Headquarters) represent departments, and Tokyo and London represent geographical locations, all under the Organization ACME.

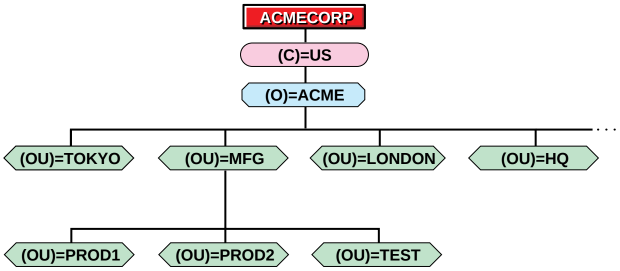


Figure 7-4 **Directory Tree with Mixed Organizational Unit Object Types**

Some areas of your tree might need more than one Organizational Unit. In the current example, the Organizational Unit MFG contains another level of Organizational Units because MFG itself is a self-contained business unit, with its own Quality Assurance department, Engineering and Development departments, etc.

Therefore, another level of Organizational Units resides under the MFG Organizational Unit to allow the department network supervisors more flexibility in designing their portions of the Directory tree.

Having different Organizational Units can help network supervisors customize the Directory tree for their particular needs.

The tree in this example facilitates replication well, but it might be more difficult to manage than the example illustrated in Figure 7-3.

Placing Leaf Objects in the Directory Tree

Container objects and their contents should be defined by workgroups, shared resources, and information usage. Therefore, leaf objects representing resources used by a single group should be placed in the same container.

Keep the following considerations in mind when placing leaf objects in the Directory tree:

- Design your Directory tree so that users have shared access to resources.

For example, if you have a high-speed printer in the organization that everyone needs access to, place the Printer object for that printer in a container where you can assign rights to allow everyone access to that Printer object.

- You can always add, delete, or move leaf objects after you have installed your Directory tree.
- Create User objects only in the container object where they will typically log in. It is undesirable to create duplicate User objects for the same person.

  Plan to use User Templates in specific Organization and Organizational Unit objects. For more information, see "Managing User Templates" in *Supervising the Network*.

  Also, plan to use Alias objects as necessary. Alias objects refer to an object that exists elsewhere in the tree without actually duplicating the object. Updates to an original object are immediately available to all Alias objects that reference it.
- Rights should be granted by exception. That is, you should grant rights at the container level, then at the group level, and then on an individual object level if necessary.

  For more information on container rights, see see "Container Rights" on page 26.

  For more information on inheriting rights, see "Security" in Concepts. For more information on Group objects, see "Managing Group Objects" in Supervising the Network.

Directory Tree Planning Example

The following example represents some of the planning conventions used for implementing NDS in an organization with offices across a continent.

Assume that the ACME Corporation has offices in the following three cities within the United States:

- Sales and accounting offices located in the corporate headquarters in New York, New York
- A development and test facility in Detroit, Michigan
- Manufacturing sites in Los Angeles, California

The following figure shows the physical layout of the offices, facilities, and sites used in the example, illustrating some of the previously discussed planning guidelines.

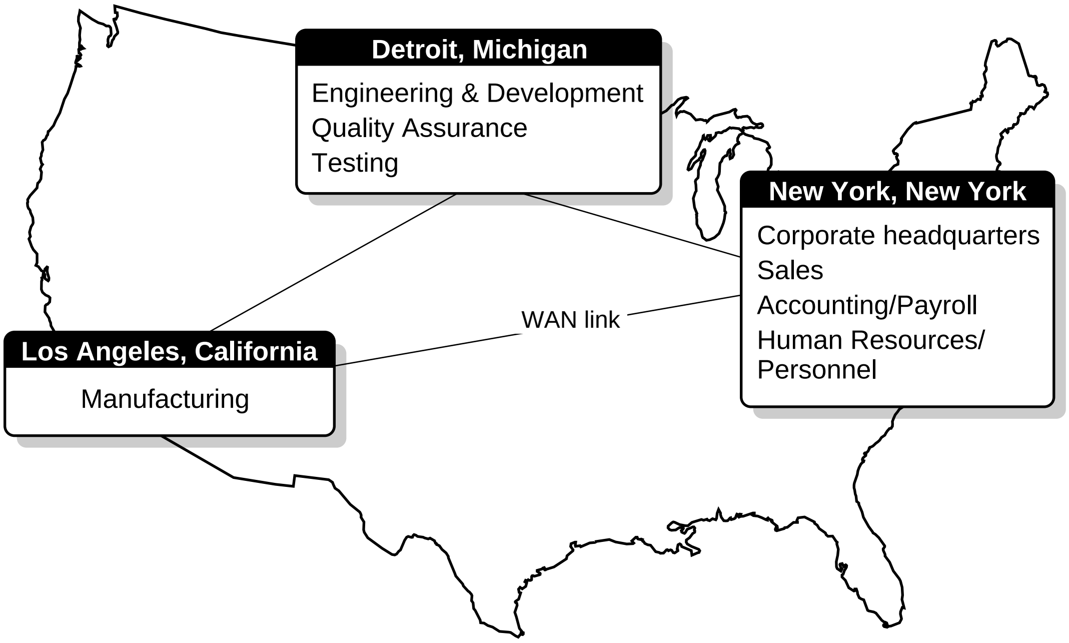


Figure 7-5 **Physical Layout of a Medium-to-Large Directory Tree**

The following figure shows the logical layout for an example Directory tree for ACME Corporation and some example names for leaf objects.

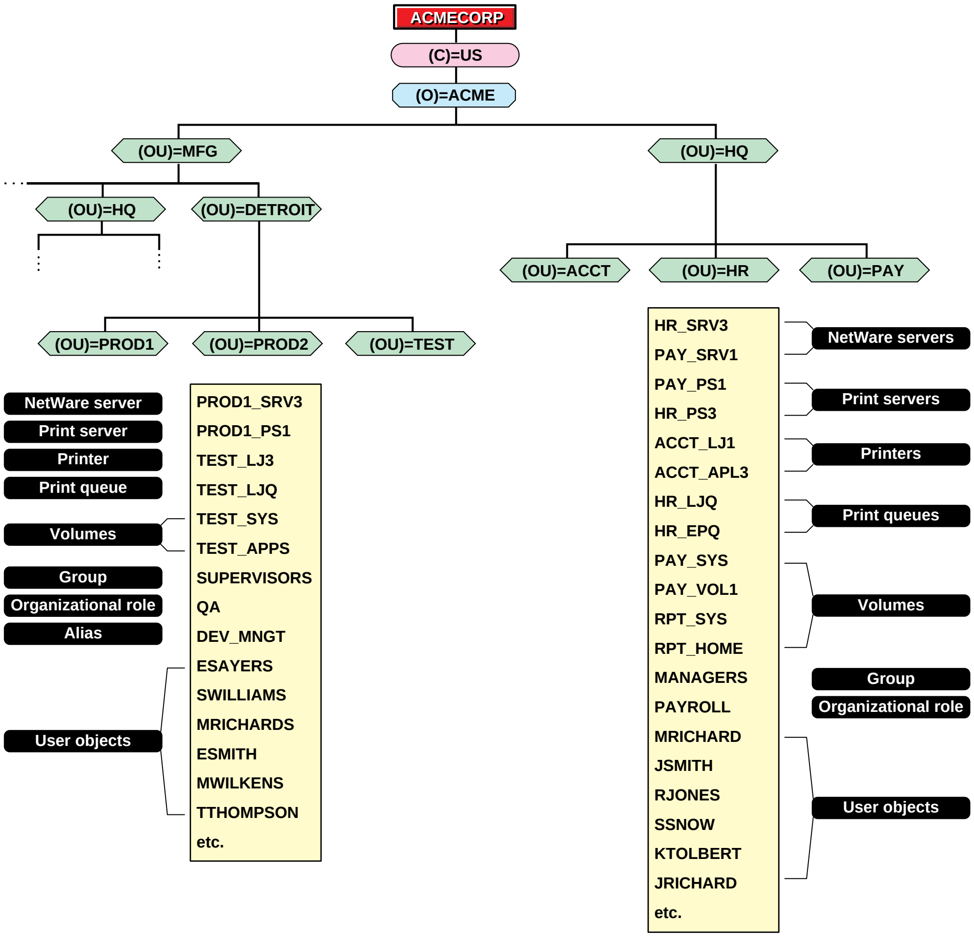


Figure 7-6 **Example Logical Layout and Leaf Object Names**

Notice that in this example all the usernames start with the initial of the first name, followed by the last name. Also notice that the names for LaserJet* printers include an “LJ” and Apple* printers include an “APL”.

These are just examples of how naming standards can be used in the Directory tree. However you decide to name your objects, you should standardize naming throughout the Directory tree in order to exploit the Directory to its fullest.

See Appendix C, “NDS Object Classes and Properties” for ideas on how to standardize the naming of objects and properties in your Directory tree.

Developing a Replication Strategy

Replicas serve two purposes.

- They provide fault tolerance.
- They decrease WAN link traffic at login and authentication.

Providing Fault Tolerance

If your network covers a large geographical distance, you might consider placing partition replicas on a server in another area. This accomplishes two things.

- It allows users in that area to access your partition more rapidly.
- It provides a backup of your partition if a disaster destroys some local servers and replicas.

You should have enough replicas of every Directory partition to provide sufficient database backup. If you lose a partition and do not have a replica of that partition, you could permanently lose access to a part of your Directory tree.

Directory replication does not provide fault tolerance for the file system. Only Directory information about objects is replicated.

To provide fault tolerance for your files, you must take advantage of the host system's fault tolerance features.

Decreasing WAN Link Traffic

If users are accessing the Directory tree through a WAN link, you can place a read-only replica of the necessary partition on a local server so they don't need to cross the WAN link.

Storing a read-only or writable replica on servers that are across a WAN link can be helpful because it cuts down on the traffic that has to cross the link when users try to access that partition's information. Nevertheless, there will be some increase in traffic due to the synchronization of replicas.

Read-only replicas do not support user login. Do not create a read-only replica of a partition that users must attach to before they authenticate to the network.

With a replica of a distant partition stored locally, users have immediate access to the objects they need. The only time Directory information crosses the link is when replicas are being updated.

However, remember that every server that carries a replica must receive all changes to any object within that partition. The more replicas of a given partition you have, the more time needed and the more WAN traffic that exists to fully synchronize the replicas.

Before you begin distributing replicas, think about how much data you want in a partition. Because replicas are stored on servers, unnecessary information in a replica is an inefficient use of disk space and network traffic.

If a partition becomes very large, and you only need to replicate a portion of it, you can use utilities to split the partition and replicate only the necessary portion.

The following figure shows one way to distribute replicas across the WAN on our example tree.

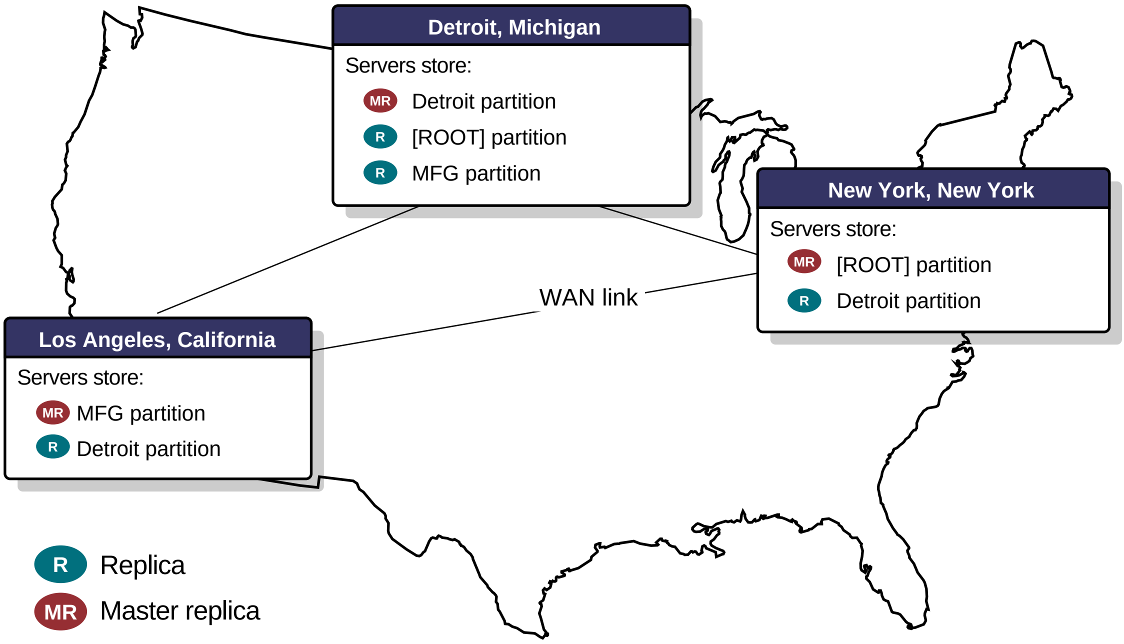


Figure 7-7 **Replica Distribution across a WAN**

This example reflects the following:

- Master replicas are stored at each local site. That is, a server at the New York site stores the master replica of the [Root] partition, a server at the Los Angeles site stores the master replica of the MFG partition, etc.
- Servers at the Detroit location store replicas of the [Root] partition and the MFG partition so that information is locally accessible to users in the Engineering and Quality Assurance departments.
- A server at the Los Angeles office stores a replica of the Detroit partition so that developers in Los Angeles do not have to use a WAN link to access information from their counterparts in Detroit.
- A server in the New York office stores a replica of the Detroit partition to allow local access.
- Objects in the Detroit office often access objects in other parts of the Directory tree across the WAN. (This office operates as a self-contained business unit within the organization.)

This is only one example of how to place replicas. You must decide how to best eliminate single points of failure and provide your users with easy access to information according to your physical network layout..

For more information about	**Refer to**
Partitions and replicas	"Creating and Managing Directory Services Partitions" in *Supervising the Network*
PARTMGR text utility	"PARTMGR" in " in *Utilities Reference*
Partition Manager in NetWare Administrator	"Managing the NetWare Directory Services Tree" in *Supervising the Network*

Developing a Time Synchronization Strategy

Before you install the NetWare Server for HP-UX, decide the following based on your physical network layout and your network time synchronization needs:

- What type of time servers do you need?
- Where should time servers be located on the network so that fault tolerance is provided and network traffic is kept to a minimum?

For many environments, the default values for the time synchronization parameters are very effective. The first NetWare server installed to a Directory tree is set as a Single Reference time server. All later NetWare servers are set as Secondary time servers. The Single Reference time servers broadcast their presence and the time via SAP, and Secondary time servers listen for these broadcasts and adjust their time accordingly.

If you choose not to use the installation defaults, you need to know which time server function to designate on the server you are installing in order to implement a network-wide time synchronization plan.

Provide a time synchronization plan to all the local network supervisors who will install a NetWare Services server on the network so they can designate the correct time synchronization function on each server they install.

For more information about	Refer to
Different ways to set up time synchronization on larger networks	“Choosing a Time Synchronization Method” in chapter 5
Service Advertising Protocol	“SAP (Service Advertising Protocol)” in chapter 5
Time servers	“Time Servers” in chapter 5.

For more information about	Refer to
Time synchronization	“Choosing a Time Synchronization Method” in chapter 5 “Managing Network Time Synchronization” in *Supervising the Network*

Developing a Security Strategy for the Directory Tree

Access control in NDS is very powerful and flexible, and it can also be very easy to implement.

You can use the default security provided during the installation of the Directory tree and then add additional security as needed.

You can further control access to objects within the tree in various ways, as explained in the following sections.

Trustee Assignments

Grant trustee assignments to objects for other objects and their properties.

Container Rights

Rights can be granted at a container level. This allows you to exploit the hierarchal structure of the Directory tree.

By granting rights at the container, those rights are automatically available for every object in that container unless masked by an Inherited Rights Filter. See “Inherited Rights Filter” in chapter 1.

Group Object Rights

Create Group objects to give groups of users limited or unlimited access to particular objects or their properties in the Directory tree.

Inherited Rights Filter

The Inherited Rights Filter is a list of rights that can be assigned for any object. It controls the rights that a trustee can inherit from parent container objects.

Security Equivalency

Use the Security Equal To property to give a user access to the same information or rights that another user has access to.

When a user is added to the membership list of a Group object or the occupant list of an Organizational Role object, the Group or Organizational Role is listed in that user's Security Equal To list.

By using a security equivalency, you avoid having to review the whole Directory tree structure and determine which rights need to be assigned to which directories, files, and objects.

If an object in a User object's Security Equal To list is deleted from the Directory tree, the user no longer has the rights granted through that object.

User objects that manage other User objects should be granted the Write right to the Security Equal To property. This allows User object managers to make users security equivalent to other users that they manage.

User object managers also need the Write right to the ACL property of the objects so that they can add to a User object's Security Equal To property.

Every object inherits rights from the container objects that are part of its Distinguished Name. This means, you can make a container a trustee and objects in or below that container receive the trustee assignment as if you individually granted such an assignment to each of them.

Every object in a container object has the rights that are granted to that container through the Security Equal To property. However, container objects are not listed in a User object's Security Equal To list.

The Security Equal To property is not transitive; that is, if Tom is security equivalent to Jill, and Jill is security equivalent to Bob, Tom is not security equivalent to Bob through Jill. The Security Equal To property grants Tom only those rights that Jill is explicitly granted.

In networks containing confidential data, take care that you don't inadvertently give a user access to restricted information.

For more information about	Refer to
Inherited Rights Filter	"Inherited Rights Filter, NDS OBject" in *Concepts*
Security and security examples	"Security" in *Concepts*

Developing an Integration Strategy for Bindery Services

When planning a hierarchical Directory tree, consider applications and users that still rely on bindery services.

Bindery-based users can access any object in the Directory tree by using multiple accounts. But this can result in significantly more work for the network supervisor (especially if numerous users need several accounts).

Although multiple accounts might still be necessary in your Directory tree, thoughtful planning can reduce the number of accounts you need to create.

Bindery services users should be grouped within a few container objects (bindery contexts) defined by workgroups, shared resources, and information usage and exchange. Placing similar users in the same container object makes it easier to give bindery services users access to the resources they need.

Managing Bindery Services

Once NDS is installed, ADMIN can use the NETADMIN or NetWare Administrator utility to manage the Directory tree from a client workstation.

Using the NetWare DOS Requester™ software, your DOS-based client workstations can take full advantage of the NDS functionality and access the NDS administration tools, such as NETADMIN and NetWare Administrator, for managing bindery services.

See “Managing the NetWare Directory Services Tree,” in Supervising the Network for more information.

Changing Bindery Context

You can use the System Administration Manager (SAM) or the nwcm command line utility to set a server’s bindery context.

For example, suppose you want to change a server’s bindery context to the Organizational Unit MFG under the Organization object ACME. At the server, use the following command:

```
nwcm -s ds_bindery_context=”OU=MFG.O=ACME”
```

Changes made with SAM or nwcm are not effective until the NetWare server is restarted.

Changing Directory Tree Structure

You should always think about bindery services users when making changes to the Directory tree. A change in the structure of the tree could prevent some bindery services users from accessing the network or network objects.

Moving Bindery Contexts

If users are using bindery services within a specific container and that container is moved, you need to reset the bindery context on the servers users are logging in to.

Where to Go from Here

If you want to	Go to
Use the management utilities for NDS	Chapter 9 “Managing NetWare Directory Services”
Implement NDS on your network	Chapter 8 “Implementing NetWare Directory Services”

8

Implementing NetWare Directory Services

Overview

This chapter introduces several models that can be used for implementing the NetWare® Directory Services™ (NDS) technology on your network.

The following topics are discussed on the indicated pages:

Introduction

Implementing NDS™ technology on your network can be as simple or as complex as you want it to be. The flexibility of NDS allows you to install and run it on a single server or on many servers.

With NDS, you can create an enterprise-wide information system that spans multiple sites and countries and maintain multiple partitions and replicas within a multilevel hierarchy of containers and objects. Or, you can create a small workgroup-based network environment.

There are many reasons to implement NDS on any size network. The following are commonly cited reasons to implement NDS:

- Simple, flexible, and cost-efficient network administration, regardless of network size

  NDS allows a single network supervisor to administer an entire network of resources from a single location or to share responsibility with local site supervisors using the same administration tool and database.

- Reliable fault tolerance

  By carefully partitioning and replicating your Directory database, you can decrease WAN link traffic and provide for an accidental loss of Directory information.

- Flexible and usable implementation in mixed environments

  A thoughtful implementation of bindery services can ease the migration process from bindery-based NetWare installations.

- Support for enterprise applications

  NDS supports enterprise applications, such as demographics research tools, database applications, human resources/payroll applications, scheduling systems, statistical services applications, document management, and electronic mail.

- Increased network performance

  NDS allows you to determine how and where network traffic is generated on the network. You can confine network traffic to a local server by implementing partitions and replicas.

- Advanced security features

NDS incorporates the advanced RSA (Rivest, Shamir, and Adleman, developers of this particular public key encryption system) security features that make encrypted, single-login authentication to network resources possible.

NDS security is based on a top-down architecture. All rights to network resources are established through Access Control Lists (ACLs) that allow for sophisticated, but easily managed, administration.

Security features can be set up using NetWare Administrator or NETADMIN.

The following discussion outlines the recommended tasks to be performed for implementing NDS features and functionality on all types of networks.

Completing General Tasks and Guidelines for All Networks

To implement NDS on your network, you need to first complete the following general tasks:

1 Finalize and use any planning documents you have created to make a list of the Directory objects you will install.

This list should include all users, servers, print queues, and other Directory tree objects that will be installed. When listing Directory tree objects, establish a naming standard. By using a standard when creating object names, you can make it easier to recognize objects by type and name.

Use similar guidelines when naming all objects. The conventions you use should be consistent across the entire Directory.

Consult Appendix C ,"NDS Object Classes and Properties" for help in creating this document.

2 Sort Directory objects by location.

You can decrease network traffic by physically locating objects near the users who will access those objects. This keeps data flowing in relatively small segments, rather than travelling across several routers and cable segments where traffic could become congested.

If you plan to use bindery services, centralize the objects that bindery services users will use in a common container. This makes managing the context of bindery services objects easier.

3 Sort objects into a logical hierarchy.

When organizing your Directory tree, consider the following possible organizational structures:

- **Organizational chart structure**. Base the Directory tree on the structure of your organization. When planning the Directory, you can begin with an organizational chart, and then modify that chart according to network access requirements and other factors.

- **Geographic structure**. Use geographic locations as Organizational Units. Then you can use organizational charts for each location to organize workgroups or departments at each location.
- **Functional structure**. Organize your Directory tree by function if users or groups in your organization perform similar functions. Users with similar functions are likely to share servers and other resources, so it makes sense to group them together.
- **Bindery services structure**. Group bindery services users within containers (bindery contexts) defined by workgroups, shared resources, and information usage and exchange. By placing similar users in the same container object, you make it easier to give bindery services users access to the resources they need.

4 Install the first server and set up the Directory tree.

Use the DS_Install utility to install NDS. During this process you are prompted to specify the root context and tree name.

You must also set the server context within the Directory tree. If you want to access information on the global internetwork, add a country code when setting the server context and a Country object will be created directly below the [Root] object.

Keep in mind that your network hardware supports both file services and Directory services. If you add large numbers of leaf objects, such as users or print queues, to a single container object, you might need to increase the amount of shared memory on the server.

5 Use NetWare Administrator utility or NETADMIN and PCONSOLE to complete the setup.

The NetWare Administrator utility is a Windows-based utility, and the NETADMIN and PCONSOLE are DOS-based utilities. To run these utilities, you must first install and set up a DOS or Windows client workstation.

Then, you can set up the remaining Directory tree structure, create objects for all network resources you want available in the Directory database, and create Profile objects for maintenance purposes.

- **Leaf objects**. Place leaf objects in containers that provide the best access for the resources, groups, and users that use them.

For example, a NetWare server object that stores a replica of each partition on the network can be placed in an Organization object for more efficient network management. Other servers and print queues can be placed in Organizational Units with the users or groups that utilize them.

- **Profile objects**. Create Profile objects that provide organization or department login scripts in the appropriate container objects for groups of users who need similar work environments but who are not located in the same container object.

  Implementing objects this way allows for easy, centralized control at the top of the tree and local control of the lower levels. At each container level, a User object with supervisory rights has authority over the objects within that container object.

6 Add new servers to appropriate contexts.

To add a new server, use the DS Install utility to install and set the appropriate context within the Directory tree.

7 Set the appropriate container and property rights.

Security features can be set up in NetWare Administrator or NETADMIN.

8 Configure time synchronization by specifying time synchronization parameters for each NetWare server process.

The number and location of container objects, partitions, and replicas determine the type of time servers you should create for your network.

Time synchronization is set up and managed with SAM or nwcm.

9 Make considerations for, and enable, bindery services by setting bindery contexts.

For security, optimum performance, and reliability, it is a good idea to group servers within container objects, depending on department or site. If, for example, your organization is spread over three cities, specify a site-specific container object as a bindery context for the following reasons:

- To provide local control over bindery services at each site

  This allows the network supervisors to control local administration—updating local servers, adding or deleting users, installing new equipment, and performing other tasks that are often best handled on a local basis.

- To improve security

If, for example, network supervisors in three different cities have supervisor rights over the same container object (bindery context), each of them can assign rights that the other two would disagree with.

- To decrease traffic over WAN links

  If, for example, users in London and Tokyo had their User objects in a bindery context served by a server in New York, every data transmission would take place over WAN links. This would likely result in decreased performance and create the potential for other problems.

  To enable bindery services for objects within a container object, you need to set a bindery context to that container. To specify a bindery context for a server, use the System Administration Manager (SAM) or nwcm utility.

10 Optimize and manage Directory trees.

Use PARTMGR (or the Partition Manager option in NetWare Administrator) to manage the Directory databases on your network.

Partitions. Create most of your partitions at lower levels of the Directory tree. Workgroup boundaries generally determine the number of partitions required in a tree. You should partition your tree in relation to the use and physical locations of network resources. You should create partitions only if they provide better performance or fault tolerance to the network and tree.

Before performing any partition operation, ensure that the state of synchronization for all servers affected by the operation is stable. The following table provides recommendations for determining which partitions will be affected by what operation:

Partition Operation	Partitions Affected
Create, add, delete a partition	Target partitions
Change the replica type	Target partitions
Rebuild any partition replica	Target partitions
Join partitions	Parent and child partitions

Replicas. Do not create too many replicas of the [Root] partition and other parent partitions. Otherwise, you force NDS to keep track of more child partition references than necessary. The appropriate number of replicas for any given partition depends on your environment.

If you create your Directory tree with the network user and resources in mind, you will find that the most efficient use of replicas—reducing WAN traffic while providing fault tolerance—means you should not need many replicas.

Implementing NDS on Various Sizes of Networks

The following discussions outline the recommended implementation of NDS features and functionality specific for small, medium, and large networks. You must decide which method or combination of methods best suits your organization's particular needs and requirements.

If you are implementing NDS in a medium-to-large-sized network, you might benefit from the information provided in chapter 7, "Planning NetWare Directory Services Implementation" for help in developing an implementation plan for NDS.

Small-Sized Network

Implementing NDS on a small sized network is typically based on two possible models:

- The physical location of network resources
- The departmental structure of the organization

The following figure shows a Directory tree in which ACCT (Accounting), HR (Human Resources), and PAY (Payroll) represent departments, all under the Organization HQ (Headquarters).

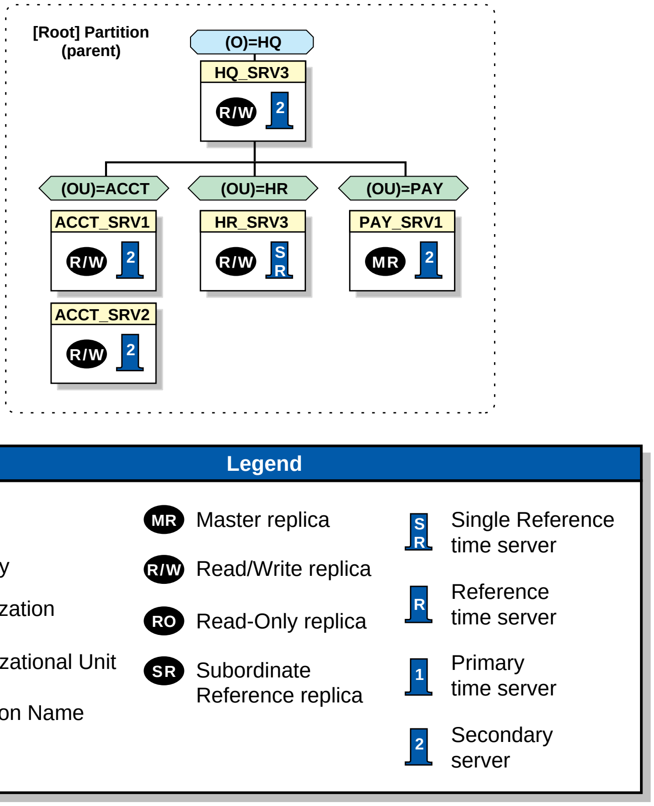


Figure 8-1 **Example of a Small-Sized Directory Tree**

Directory Tree Structure

Small-sized networks are commonly site-, workgroup-, and department-oriented in structure. They are easily managed by a system-wide administrative group with central management at the organizational and departmental levels.

The Directory tree begins with a single Organization object with few or no Organizational Unit objects below. If Organizational Units exist, they are based on functional groups, projects, departments, etc., within a single site.

Resources are usually shared by all network users and groups.

Time Services

Although small-sized business might be restricted to a single- or multiple-segment LAN, time services is still important.

A Single Reference time server is usually adequate for LAN-based networks. The Single Reference time server is monitored and periodically adjusted for time by the network supervisors.

All other servers in the network are designated as Secondary time servers.

Partitions

Workgroup boundaries generally determine the number of partitions required in a tree. You should partition your tree in relation to the use and physical location of network resources. You should create partitions only if they provide better performance or fault tolerance to the network and tree. Small networks may not require partitioning.

If you think it is necessary, create a small number of partitions at the top levels of the tree.

Replicas

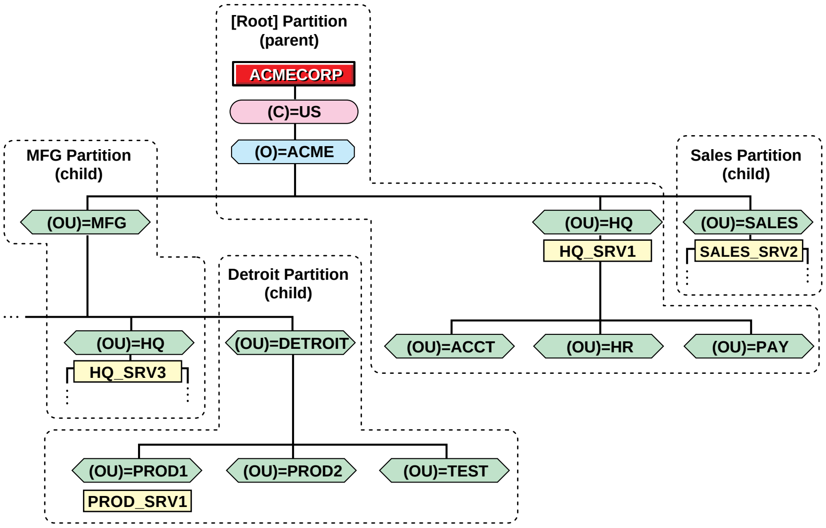
Each server on the network should contain all the resources needed at its location, because small-network users rarely connect to servers at other locations. Replicas, in this case, will most likely not decrease WAN traffic.

However, replicas provide fault tolerance. You should copy two to three replicas of each partition somewhere on the network to provide fault tolerance.

Medium-Sized Network

Implementing NDS on a medium-sized network is typically based on your business’s organizational chart with some geographic considerations for your branch offices.

The following figure shows an example of a common Directory tree structure for a medium-sized network.



Servers		Partitions			
		[Root]	Detroit	Sales	MFG
1	PROD_SRV1	MR	MR	SR	SR
2	SALES_SRV2	R/W	SR	MR	MR
1	HQ_SRV3	R/W	SR	MR	MR
1	HQ_SRV1	MR	R/W	R/W	R/W

Legend

[ROOT]	MR Master replica	SR Single Reference time server
(C) Country	R/W Read/Write replica	R Reference time server
(O) Organization	RO Read-Only replica	1 Primary time server
(OU) Organizational Unit	SR Subordinate Reference replica	2 Secondary server
(CN) Common Name		

Figure 8-2 **Example of Medium-Sized Directory Tree**

Directory Tree Structure

Medium-sized networks are commonly workgroup- and department-oriented in structure. They are typically managed by a central, system-wide administrative group and department network supervisors.

The Directory tree begins with a general Organization object that has multiple Organizational Unit objects below. Organizational Units are based on functional groups, projects, departments, etc.

In the Organization object and high-level Organizational Units are enterprise resources that are managed centrally, including the following:

- Servers that function as SAA* or TCP/IP gateways or as a NACS™ system
- User objects for network supervisors
- Profile objects that create an environment for specific users and groups

Create User objects for centralized supervisors and Organizational Unit (OU) supervisors within their respective container objects. The OU-level supervisors are often department network supervisors.

Centralized supervisors are responsible for general network management and overall support for the Directory tree. OU-level supervisors are responsible for day-to-day tasks, such as User object and resource management and local server backup.

Centralized management helps facilitate the implementation of network-wide standards. You should create and distribute a standards document for the entire network before implementing NDS.

Time Services

Because many medium-sized networks maintain some level of WAN connectivity, time services support is an important consideration.

A Single Reference time server is usually inadequate for networks that have WAN connections. You should use a group of Primary time servers as the basis for network time services.

Determine which servers within your organization provide system-wide services, such as directories or applications that are accessed by multiple departments or the entire organization.

Choose a limited number from the group of servers you identified to be installed as Primary time servers. Limiting the number of Primary time servers to a select few minimizes the network traffic used when the time servers vote on the current time. Typically, you should have one or two Primary time servers at each location on the network.

Set up remaining servers as Secondary time servers.

Partitions

Partitioning medium-sized networks should follow the structure of your Organizational Unit objects. You might want to create a partition for each high-level Organizational Unit in the tree.

This allows each partition to contain all the resource objects that a particular department needs to access. Place the [Root] and Organization objects in the same partition.

Replicas

Create replicas to ensure adequate redundancy of critical partitions. Determine which servers within your organization provide system-wide services, such as applications that are accessed by multiple departments or the entire organization.

Place replicas of the partitions that include these critical servers on other servers in different locations on the network. This allows all users to authenticate to an enterprise resource without increasing network traffic.

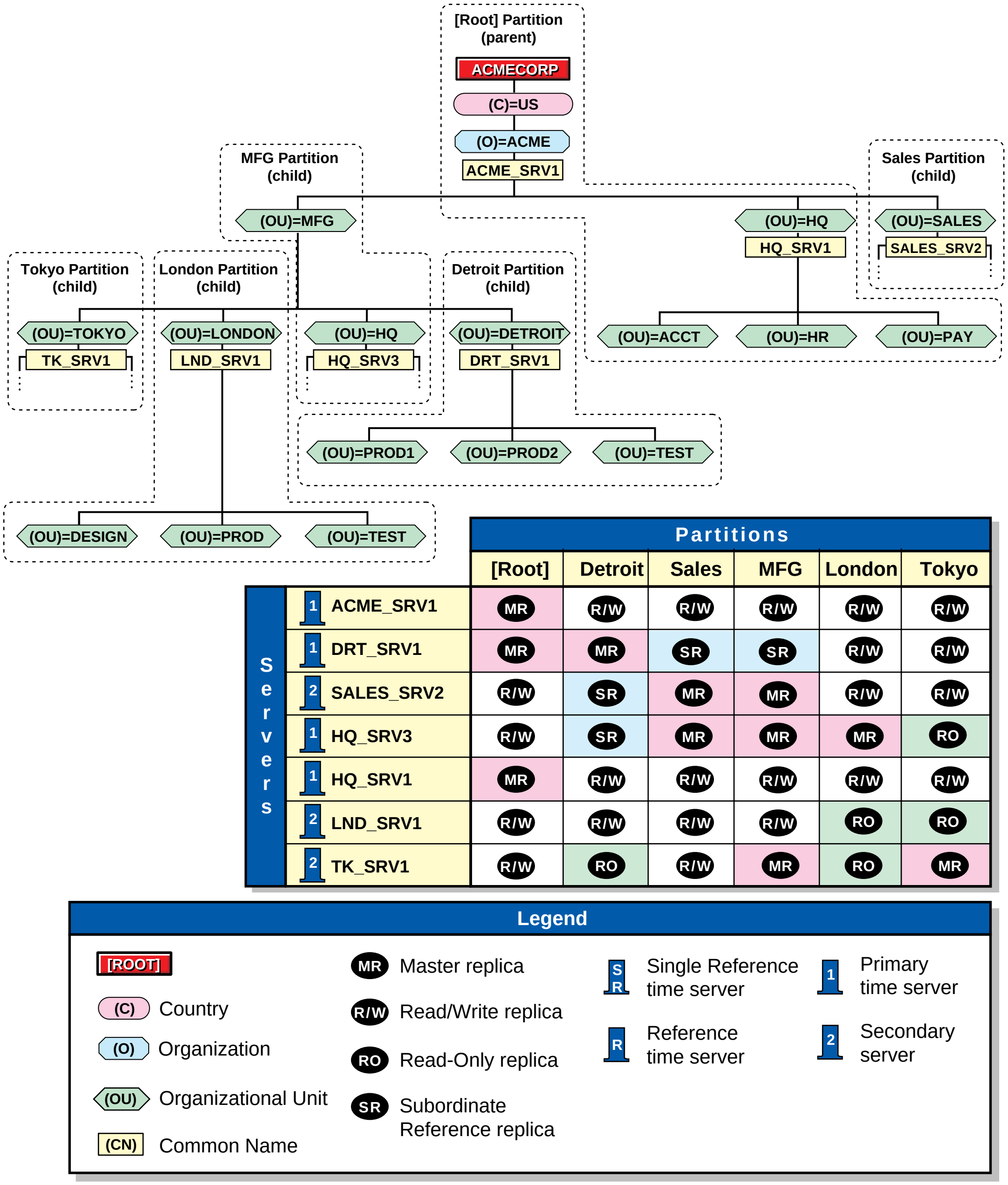
For servers that provide local services, place replicas of the partitions that include them on other local servers.

If only one server exists at a location, place a replica of the partition that includes the server on a server in a different location. Provide additional replicas if possible.

Large-Sized Network

Large-sized networks are enterprise focused, linking large, organizational networks with many other equal- or smaller-sized networks. They require flexibility, advanced security, and centralized management of distant resources as well as local supervision.

The following figure shows an example of a Directory tree for a large-sized network.



Servers	[Root]	Detroit	Sales	MFG	London	Tokyo
1 ACME_SRV1	MR	R/W	R/W	R/W	R/W	R/W
1 DRT_SRV1	MR	MR	SR	SR	R/W	R/W
2 SALES_SRV2	R/W	SR	MR	MR	R/W	R/W
1 HQ_SRV3	R/W	SR	MR	MR	MR	RO
1 HQ_SRV1	MR	R/W	R/W	R/W	R/W	R/W
2 LND_SRV1	R/W	R/W	R/W	R/W	RO	RO
2 TK_SRV1	R/W	RO	R/W	MR	RO	MR

Figure 8-3 **Example of Large-Sized Directory Tree**

Directory Tree Structure

The Directory tree begins with a general Organization object that has multiple Organizational Unit objects below. Organizational Units are based on functional groups, projects, departments, etc., and also on-site locations such as cities or countries.

Large networks typically require both system-wide administrative groups with central management at the organizational and departmental levels and site-based administrative groups that manage local resources and objects.

Large networks typically have a number of high-level divisions within the organization that form the top level of Organizational Units. Most of these divisions are divided into subdepartments which form a second level of Organizational Units. A third level of Organizational Units might consist of locations or functional groups.

Organizational and Departmental Containers Within Organization and Organizational Unit objects are enterprise resources that are managed centrally, including the following:

- Servers that function as SAA* or TCP/IP gateways or as a NACS™ system
- User objects for network supervisors
- Profile objects that create an environment for specific users and groups

Centralized Management As organizations grow, it is necessary to maintain the workgroup and departmental structure of an organization while sufficiently increasing the centralized administration.

You should create User objects for centralized supervisors and Organizational Unit-level supervisors within their respective container objects.

Centralized supervisors are responsible for general network management and overall support for the Directory tree. Organizational Unit-level supervisors are responsible for day-to-day tasks, such as User object and resource management and local file server backup.

Centralized management also helps facilitate the implementation of network-wide standards. You should create and distribute a standards document for the entire network before implementing NDS.

Time Services

Because most large-sized networks maintain high levels of WAN connectivity, which span time zones and international datelines, time services support requires careful planning.

It is critical to have a constant reference of time in order for NDS synchronization to take place. Time is also important to the proper execution of certain events and features, such as network backups and time-based security.

You should use one Reference time server and a group of Primary time servers as the basis for network time services. This ensures that a proper and accurate time reference is available at all times.

Determine which servers within your organization provide system-wide services, such as directories or applications that are accessed by the entire organization. From the servers you identify, select one to function as the Reference time server and set up the others as Primary time servers.

Each geographically distinct site should have at least one Primary time server.

All other NetWare servers in the network should be set up as Secondary time servers.

The Reference time server should be adjusted periodically by an outside time source, possibly the U.S. Naval Observatory Clock in Annapolis, Maryland.

Partitions

Partitioning of large-sized networks should follow a multi-tiered partition plan.

Each division-level Organizational Unit has its own partition representing that container and its objects. Each lower-level Organizational Unit is the root for a partition that includes itself and all the other container and leaf objects beneath it in that branch of the tree.

The [Root] and Organization objects should form one partition. This partitioning structure ensures that all the critical access points in the tree are available and can be replicated for redundancy.

Replicas

Create replicas to ensure adequate redundancy of critical partitions. Determine which servers within your organization provide system-wide services, such as applications that are accessed by multiple departments or the entire organization.

Place replicas of the partitions that include these critical servers on other servers in different locations on the network. This allows all users to authenticate to an enterprise resource without increasing network traffic.

For servers that provide local services, place replicas of the partitions that include them on other local servers.

If only one server exists at a location, place a replica of the partition that includes the server on a server in a different location. Provide additional replicas if possible.

For added security and fault tolerance, place a read/write replica of each partition on a server at the Organization object level of each Directory tree. This enables the central network management staff to maintain a complete Directory database in one location.

Make sure that every partition has a sufficient number of replicas available on the network, including replicas on appropriate distant servers, to ensure fault tolerance and to decrease WAN link traffic.

Most replicas should be located on servers within the main corporate network, except for other locations that have multiple servers. In these cases, replicas of the appropriate partitions are located on all these servers.

Additional Information

Topic	Reference
Container objects	“Container Object” in *Concepts*; “Container Objects” in chapter 2
Context	“Container Object” in *Concepts*; “Context and Names” in chapter 2
Leaf objects	“Leaf Objects” in *Concepts;* “Leaf Objects” in chapter 2
NetWare Administrator utility	“Using NetWare Administrator” or “Using NETADMIN” in *Supervising the Network*
Objects	“Container Object” in *Concepts*; “Directory Objects” in chapter 2
Partitions and replicas	“Container Object” in *Concepts*; “Directory Partitions” and “Partition Replicas” in chapter 2
Rights	“Container Object” in *Concepts;* “Object and Property Rights” in chapter 2

9

Managing NetWare Directory Services

Overview

This chapter briefly describes the management utilities and programs used to set up and maintain your implementation of the NetWare® Directory Services™ (NDS) technology on your network.

The following utilities are discussed on the indicated pages:

Utility	Page
DS Install	9-4
DS Repair	9-5
dsadmin	9-6
NETADMIN	9-5
NetWare Administrator	9-8
SAM	9-9
nwcm	9-10
PARTMGR	9-9
tsadmin	9-13
UIMPORT	9-11

Introduction

The management utilities and programs discussed in this chapter can help you build and maintain your Directory tree hierarchy and objects, as well as help you maintain the Directory database on your network.

DS Install

Use this UNIX utility to install or remove NetWare Directory Services (NDS) and to upgrade volumes into the Directory.

Using DS Install

During the installation process, DS Install scans the network for any existing Directory trees. If it does not find an existing tree, it prompts you to install the first server in the Directory tree. Installation of the first server in a Directory tree is important because it establishes the initial hierarchy of your tree structure.

The DS Install utility prompts you to set up time synchronization and set up the server context or name context. Setting the server context determines the location of the server in the Directory tree.

Additional Information

Topic	Reference
DS Install utility	"DS Install" in *Utilities Reference*
Installing Directory Services	"Installing NetWare Directory Services" in *NetWare 4.1/9000 Installation and Administration Guide*

DS Repair

Use this UNIX utility to check or repair problems in the Directory database concerning records, schema, bindery objects, and external references.

Using DS Repair

DS Repair is described in the *Utilities Reference* manual.

You can do any of the following with DS Repair:

- Repair and synchronize NDS information, such as tree structure, object records, and base schema
- Remove unknown objects, invalid mail directories, and unreferenced streams files
- Repair initial states, network addresses, and replica rings
- Perform local NDS database recovery
- Designate a new master replica for a partition that has lost this replica due to server failure

Additional Information

Topic	Reference
DS Repair utility	“Repairing the NetWare Directory Database” in *Supervising the Network.* “DS Repair” in *Utilities Reference*

dsadmin

Use this UNIX command line utility to display or temporarily set values for configurable NDS parameters, such as dstrace, ds_ttf, and a temporary bindery context.

This utility differs from the SAM and nwcm utilities in several ways. The dsadmin utility does not store any modified values when the server is shut down; to permanently change these parameters, use SAM or nwcm. Also, unlike SAM or nwcm, modifications made with dsadmin take effect immediately.

Using dsadmin

This utility enables you to dynamically set or display the configurable NDS parameters.

You can use dsadmin to do the following:

- Display the name of the NDS tree the server has been installed in
- Print existing dstrace, ds_ttf, or bindery context settings
- Set new values for the dstrace, ds_ttf, or bindery context parameters dynamically

For more information on these parameters, see “dsadmin” in Utilities Reference.

Additional Information

Topic	Reference
dsadmin utility	“dsadmin” in *Utilities Reference*
NDS system messages	Messages numbering –601 through –699 and F966 through F9FE in *System Messages*

NETADMIN

Use this text utility at a client workstation to manage NetWare Directory Services (NDS) objects and their properties.

Users can view, create, move, delete, and assign rights to any NDS object they have access rights to. Use this utility to manage access rights and the objects in your Directory database.

Using the NETADMIN Utility

You can perform the following management tasks with NETADMIN:

- Change object property values
- Create and name container and leaf objects
- Delete objects from the Directory tree
- Manage Organizational Role objects
- Manage trustee assignments to objects
- Move objects in the Directory tree
- Rename leaf objects
- Search for objects
- Manage object and property rights

Additional Information

Topic	Reference
NETADMIN utility	"NETADMIN" in *Utilities Reference*
Using NETADMIN	"Using NETADMIN" in *Supervising the Network*

NetWare Administrator

Use this graphical utility at a Windows client workstation to manage NetWare Directory Services (NDS) objects and their properties.

Users can view, create, move, delete, and assign rights to any NDS object they have access rights to. Use this utility to manage access rights and the objects in your Directory database.

NetWare Administrator is a graphical user interface utility that provides functionality similar to the NETADMIN utility.

This utility is available only if, during the VLM client installation, you selected “Yes” when asked whether to install the Windows utility. If you selected “Yes,” a NWADMIN icon is created for you. Select this icon to start the utility.

Using NetWare Administrator

You can perform the following tasks in Windows in the NETADMIN, PARTMGR, and PCONSOLE utilities:

- Assign rights in the Directory tree and in the file system
- Create users and groups
- Create and delete Directory objects
- Move and rename leaf objects
- Set up printing services
- Set up and manage Directory partitions and replicas

Additional Information

Topic	Reference
NetWare Administrator utility	“Using NetWare Administrator” in *Supervising the Network*

SAM

Use this graphical utility to configure various NetWare services from the server console. The SAM utility is an easy-to-use graphical version of the nwcm command line utility. The following section describes only the NDS parameter that can be configured with NetWare Setup.

Using NetWare Setup to Set NDS Parameters

The SAM utility allows you to configure the Directory Services bindery context using a graphical, multi-column browser. You must restart the server for any changes to take effect.

Additional Information

Topic	**Reference**
SAM	"SAM" in *Utilities Reference*
Configuring NDS with NetWare Setup	"Managing the NetWare Directory Services Tree" in *Supervising the Network*

nwcm

Use this UNIX command line utility to view and configure a variety of NetWare system parameters, including specifying a bindery context for bindery services.

Additional Information

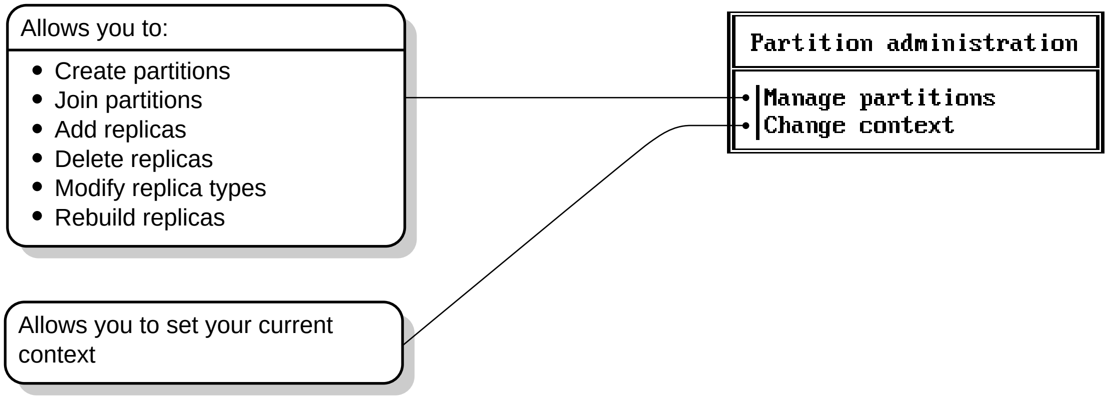
For a complete description of the nwcm command parameters, see “nwcm” in *Utilities Reference*.

PARTMGR

Use this utility at a client workstation to

- Distribute your NDS database.
- Manage partitions and replicas.

The following figure shows the functions available in PARTMGR.



Allows you to:

- Create partitions
- Join partitions
- Add replicas
- Delete replicas
- Modify replica types
- Rebuild replicas

Allows you to set your current context

Figure 9-1 **Functions in PARTMGR**

Using PARTMGR

You can perform the following tasks by choosing “Manage Partitions” from the “Partition Administration” menu:

- Browse up the Directory tree to the parent container
- Browse down the Directory tree to see the Server objects in containers
- View a list of the replicas stored on a server
- View or modify a partition’s replicas
- View or modify the replicas of the current container object (if that container object is a partition)
- Create a new partition with a container object as the root of the partition
- Join a partition with its parent partition

Additional Information

Topic	Reference
PARTMGR utility	"PARTMGR" in *Utilities Reference*

tsadmin

Use this UNIX command line utility to display time synchronization status or restart time synchronization.

Modifications made with tsadmin to time synchronization parameters will take effect immediately when you use this utility to restart time synchronization.

Using tsadmin

This UNIX utility enables you to dynamically set or display the configurable time synchronization parameters.

You can use tsadmin to do the following:

- Display time synchronization information
- Display date and time kept by server’s clock
- Restart time synchronization

UIMPORT

Use this utility at a client workstation to create, delete, and update User objects and their properties by importing user information from an existing database into the Directory database.

Using UIMPORT

This utility is particularly valuable if you have hundreds or thousands of user records that you want to record in NetWare Directory Services without having to manually re-create each user.

Any application capable of converting records to a comma-separated ASCII file can work with UIMPORT.

Use UIMPORT to automate the maintenance of your Directory database when you want to

- Create User objects in the NDS database using records from another database.
- Update User properties in the NDS database when records are changed in your original database program.
- Delete User objects when their accounts on the network are no longer needed.

Additional Information

Topic	Reference
Using UIMPORT	“Importing User Information into the NDS Database” in *Supervising the Network*

A

Appendixes

Overview

The NetWare® Directory Services™ technology supports a large number of object classes and properties.

Creating a consistent naming standards document can make present and future implementation of your Directory tree easier and more efficient.

Naming standards can also help ensure that the Directory objects you create are intuitive and useful to users and groups on your network.

Contents

This section is divided into three appendixes, with the following information discussed on the indicated pages:

Purpose	Chapter
To reference lists and explanations of the object classes and properties available in NetWare Directory Services	Appendix B, “NDS Object Classes and Properties”
To reference lists of available leaf objects in NetWare Directory Services	Appendix C, “Referencing and Using Leaf Objects”
To reference guidelines and samples for creating a standards document for objects in a NetWare Directory Services	Appendix D “Creating A Standards Document for NDS Object Classes and Properties”

B

NDS Object Classes and Properties

Overview

This appendix lists and explains the available object classes and properties available in the NetWare® Directory Services™ (NDS) architecture.

The following topics are discussed on the indicated pages:

NDS Object Classes and Their Functions

This section lists the most common NDS object classes, explains what each is used for, and indicates where that type of object can be contained.

Table B-1 **Object Class, Function, and Possible Container**

Object Class	Function	Possible Container
Alias	Redirects path of Directory tree branch or leaf object to another location for more convenient access	Organization Organizational Unit Root level
Bindery Object	Represents object upgraded from bindery-based server that cannot be mapped to a Directory object	Organization Organizational Unit
Computer	Represents network computers that are not file or print servers (such as gateways, routers, and sometimes workstations)	Organization Organizational Unit
Country	Additional level of organization in Directory tree	Root level
Directory Map	Specifies path on volume that points to frequently used application directory	Organization Organizational Unit
Group	Defines unordered list of users that comprise group for purpose of assigning access rights	Organization Organizational Unit
NetWare Server	Represents server that provides file and other services	Organization Organizational Unit
Organization	Defines organization within network	Country or Root level
Organizational Role	Defines position or role within organization for purpose of assigning access rights	Organization Organizational Unit
Organizational Unit	Defines subdivision within organization to contain objects	Organization Organizational Unit
Print Server	Represents network print server	Organization Organizational Unit

NDS Object Classes and Their Functions

Table B-1 Object Class, Function, and Possible Container

Object Class	Function	Possible Container
Printer	Represents physical printing device on network	Organization Organizational Unit
Profile	Specifies login script used by several users	Organization Organizational Unit
Queue	Represents batch processing queue for printing on network	Organization Organizational Unit
User	Represents user on network	Organization Organizational Unit
Volume	Represents physical volume within NetWare file server	Organization Organizational Unit

NDS Object Classes and Their Properties

This section lists the most common NDS object classes and the properties associated with each.

Table B-2 Object Class and Properties

Object Class	Properties	
Alias	ACL Aliased Object Name Back Link	Bindery Property Object Class
Bindery Object	ACL Back Link Bindery Object Restrictions Bindery Property	Bindery Type CN Object Class
Bindery Queue	ACL Back Link Bindery Property Bindery Type CN Description Device Host Resource Name Host Server L	Network Address O Object Class Operator OU Queue Directory See Also Server User Volume
Computer	ACL Back Link Bindery Property CN Description L Network Address O	Object Class Operator OU Owner See Also Serial Number Server Status
Country	ACL Back Link Bindery Property	C Description Object Class

Table B-2 Object Class and Properties

Object Class	Properties	
Directory Map	ACL Back Link Bindery Property CN Description Host Resource Host Server	L Name O Object Class OU Path
Group	ACL Back Link Bindery Property CN Description E-Mail Address Full Name GID L Login Script	Mailbox ID Mailbox Location Member O Object Class OU Owner Profile Profile Membership
NCP Server	Account Balance ACL Allow Unlimited Credit Back Link Bindery Property CN DS Revision Full Name Host Device L Messaging Server Minimum Account Balance Network Address O	Object Class Operator OU Private Key Public Key Resource Security Equals To Security Flags Status Supported Services User Version

Table B-2 Object Class and Properties

Object Class	Properties	
Organization	ACL Back Link Bindery Property Description Detect Intruder E-Mail Address Facsimile Telephone Number Intruder Attempt Reset Interval Intruder Lockout Reset Interval L Lockout After Detection Login Intruder Limit Login Script Mailbox ID	Mailbox Location NNS Domain O Object Class Physical Delivery Office Name Postal Address Postal Code Postal Office Box Print Job Configuration Printer Control S SA Telephone Number
Organizational Role	ACL Back Link Bindery Properly CN Description E-Mail Address Facsimile Telephone Number L Mailbox ID Mailbox Location Object Class	OU Physical Delivery Office Name Postal Address Postal Code Postal Office Box Role Occupant S SA Telephone Number

Table B-2 Object Class and Properties

Object Class	Properties	
Organizational Unit	ACL Back Link Bindery Property Description Detect Intruder E-Mail Address Facsimile Telephone Number Intruder Attempt Reset Interval Intruder Lockout OU L Lockout After Detection Login Intruder Limit Login Script Mailbox ID	Mailbox Location NNS Domain Object Class Physical Delivery Office Name Postal Address Postal Code Postal Office Box Print Job Configuration Printer Control Reset Interval S SA Telephone Number
Print Server	Account Balance ACL Allow Unlimited Credit Back Link Bindery Property CN Description Full Name Host Device L Minimum Account Balance Network Address O Object Class	Operator OU Print Private Key Public Key Resource SAP Name Security Equals To Security Flags Status User Version

Table B-2 Object Class and Properties

Object Class	Properties	
Printer	ACL Back Link Bindery Property Cartridge CN Default Queue Description Host Device L Memory Network Address Network Address Restrictions Notify	O Object Class Operator OU Owner Page Description Language Print Server Printer Configuration Queue See Also Serial Number Status Supported Typefaces
Profile	ACL Back Link Bindery Property CN Description L	Login Script O Object Class See Also
Queue	ACL Back Link Bindery Property CN Description Device Host Resource Name Host Server L Network Address	O Object Class Operator OU Queue Directory See Also Server User Volume
Unknown	ACL Back Link	Bindery Property Object Class

Table B-2 Object Class and Properties

Object Class	Properties	
User	Account Balance ACL Allow Unlimited Credit Back Link Bindery Property CN Description E-Mail Address Facsimile Telephone Number Full Name Generational Qualifier Given Name Group Membership Higher Privileges Home Directory Initials L Language Last Login Time Locked By Intruder Login Allowed Time Map Login Disabled Login Expiration Time Login Grace Limit Login Grace Remaining Login Intruder Address Login Intruder Attempts Login Intruder Reset Time Login Maximum Simultaneous Login Script Login Time Mailbox ID Mailbox Location Message Server	Minimum Account Balance Network Address Network Address Restrictions Object Class OU Password Allow Change Password Expiration Interval Password Expiration Time Password Minimum Length Password Required Password Unique Required Passwords Used Physical Delivery Office Name Postal Address Postal Code Postal Office Box Print Job Configuration Printer Control Private Key Profile Profile Membership Public Key S SA Security Equals To Security Flags See Also Server Holds Surname Telephone Number Title Type Creator Map UID

Table B-2 Object Class and Properties

Object Class	Properties	
Volume	ACL Back Link Bindery Property CN Description Host Resource Name Host Server	L O Object Class OU See Also Status

NDS Object Classes and Their Properties

C

Referencing and Using Leaf Objects

Overview

This appendix introduces the leaf objects available in the NetWare® Directory Services™ architecture.

The following topics are discussed on the indicated pages:

Directory leaf objects are objects that do not contain any other objects. These represent actual network entities such as users, servers, printers, computers, etc. You create leaf objects within a container object.

User-Related Leaf Objects

This section lists the available leaf objects that are related to network users and groups, explains what each is used for, and indicates when to use each.

Table C-1 User-Related Leaf Object Name, Function, and Usage

Leaf Object	Function	Usage Situation
Group	Assigns a name to a list of User objects that can be located anywhere in the Directory tree.	Many User objects need the same trustee assignments. Rather than making many trustee assignments, make just one trustee assignment to all users who belong to the group by making the trustee assignment to the Group object itself.
Organizational Role	Defines a position or role within an organization.	You want to assign rights to a particular position rather than to the person who occupies that position. The occupant might change frequently, but the responsibilities of the position do not. You can assign any user to be an occupant of an Organizational Role object because every occupant receives the same rights granted to the Organizational Role object.
Profile	Contains a profile login script. When the Profile object is listed as a User object's property, the Profile object's login script is executed when that User object logs in. The profile login script executes after the system login script and before the user login script.	A set of users need to share common login script commands but are not located in the same Directory tree container or are a subset of users in the same container.

Table C-1 User-Related Leaf Object Name, Function, and Usage

Leaf Object	Function	Usage Situation
User	Represents a person who uses the network. In the User object properties, login restrictions, intruder detection limits, password and password restrictions, security equivalences, etc., can be set.	Required for every user who needs to log in to the network. When you create a User object, you can create a home directory for that user who then has default rights to that home directory. When you create User objects, you can also choose to apply a user template to the users that provides default property values. For users who have NetWare 4 workstations, you can create the User objects anywhere in the Directory tree, but the users must know their context in order to log in. Create User objects in the container where the users typically log in. For users who have other workstations, create the User objects in the container where the bindery services context is set for the server that they need to log in to. Bindery-based users do not need to know their context because they log in to the server rather than to the Directory tree.

Server-Related Leaf Objects

This section lists the available leaf objects that are related to NetWare servers and volumes, explains what each is used for, and indicates when to use each.

Table C-2 **Server-Related Leaf Object Name, Function, and Usage**

Leaf Object	Function	Usage Situation
Directory Map	Represents a particular directory in the file system. Directory Map objects can be especially useful in login scripts by pointing to directories that contain applications or other frequently used files.	You want to avoid making changes to many login scripts when the location of applications changes. Instead, you change only the Directory Map object. For example, you have a directory that contains DOS 5.0. You could map a search drive to that directory in any login scripts you create. But if you later upgrade to DOS 6.0 and rename the directory, you would have to change the mapping in every login script where that search mapping appears. By using a Directory Map object instead, you would need to change the information in only that one object.
NCP Server	Represents a server running NetWare on your network. In the NetWare Server object's properties, you can store information about the server—such as its physical location and what services it provides. In addition, the NetWare Server object affects the network in that it is referred to by several other objects.	Automatically created during server installation. It must exist for a server's file systems and volumes to be accessible. If you have a bindery-based server, create this object to be able to access that server's volumes. When you create this object for a bindery-based server, that server must be running.

Table C-2 Server-Related Leaf Object Name, Function, and Usage

Leaf Object	Function	Usage Situation
Volume	Represents a physical volume on the network. In the Volume object’s properties, you can store identification information—such as the Host server, volume location, etc. You can also set restrictions for use of the volume, such as space limits for users.	Optional for every physical volume on the network. Automatically created for every physical volume during server installation. Can be used to display information about the directories and files on that volume.

Printer-Related Leaf Objects

This section lists the available leaf objects that are related to NetWare print services, explains what each is used for, and indicates when to use each.

These objects are created and controlled using the NetWare print utilities.

Table C-3 Printer-Related Leaf Object Name, Function, and Usage

Leaf Object	Function	Usage Situation
Print Queue	Represents a print queue on the network.	Required for every print queue on the network. See *Print Services* for more information.
Print Server	Represents a network print server.	Required for every print server on the network. See *Print Services* for more information.
Printer	Represents a physical printing device on the network.	Required for every printer on the network. See *Print Services* for more information.

Informational Leaf Objects

This section lists the available leaf objects that exist only to store information about network resources, explains what each is used for, and indicates when to use each.

Table C-4 Informational Leaf Object Name, Function, and Usage

Leaf Object	Function	Usage Situation
Computer	Represents a nonserver network computer, such as a workstation or a router.	Use this object to store information about a nonserver computer, such as its network address, its serial number, or the person it’s assigned to. This object has no effect on network operations; it only stores information about the computer.

Miscellaneous Leaf Objects

This section lists the remaining available leaf objects, explains what each is used for, and indicates when to use each.

Table C-5 Miscellaneous Leaf Object Name, Function, and Usage

Leaf Object	Function	Usage Situation
Alias	Points to another object in the Directory tree and makes it appear as if that object actually exists in the Directory tree where the Alias object is. Although an object appears both where it was actually created and where an Alias referring to it was created, only one copy of the object really exists. If you delete or rename an Alias, the Alias itself (not the object it is pointing to) is deleted or renamed.	You want to allow access to an object that is in another context. For example, you can use an Alias to represent a resource, such as a special printer, that most users in the tree need to access.
Bindery Object	Represents an object placed in the Directory tree by an upgrade or migration utility.	It is used by NDS only to provide backward compatibility with bindery-based utilities.
Bindery Queue	Represents a queue placed in the Directory tree by an upgrade or migration utility.	It is used by NDS only to provide backward compatibility with bindery-based utilities.
Unknown	Represents an NDS object that has been invalidated and cannot be identified as belonging to any of the other object classes.	Directory Services utilities rename objects that they do not recognize. Delete or re-create the correct object for the resource.

D

Creating a Standards Document for NDS Object Classes and Properties

Overview

This appendix provides you with guidelines and samples for creating a standards document for objects in a NetWare® Directory Services™ (NDS) database.

The following topics are discussed on the indicated pages:

Using a consistent naming standard makes current and future implementation of NDS™ easier and more efficient. A naming standard helps ensure that the Directory objects you create are intuitive and useful to users and groups on your network.

An effective naming standard document includes a list of objects to be implemented, the format of each property value, and the intended use of each property.

There is no preset naming standard. Different organizations can adopt different naming standards based on both their requirements and their existing configurations.

The naming standard in this appendix is offered as an example that works well for any organization, regardless of size, but that can be modified and adjusted to better meet the requirements of individual organizations.

Sample Object Naming Standards

In our examples, we have tried to create relatively short names. This helps keep the context short and reduces data traffic as NDS searches for specific objects.

If you have already chosen a different format to name users or servers in an existing NetWare 3™ network, you might want to use those names as a starting point as you implement your NetWare 4™ network.

Table D-1 Object Name and Suggested Standards

Object	Suggested Standards
Directory Map	Name Directory Map objects after the application or process being mapped. For example, WordPerfect® application files would be mapped with a Directory Map named DM-WP.
Group	Base group names on the function performed by the group. For example, a word processing group could be named GP-WP.
Organization and Organizational Unit	Choose your Organization and Organizational Unit names based on your organization’s abbreviations of unit names. For example, an organization with the name WIDGET, with a business unit called ASG, and a division called NCS would set a context in the Directory tree as OU=NCS.OU=ASG.O=WIDGET.
Organizational Role	For security purposes, always use the Organizational Role object to grant administrative rights. The Organizational Role object can be used in any situation where changes in personnel might be frequent or when an error in controlling rights would pose a grave security risk to the organization. For example, a container administrative organizational role could be named OR-NCSADMIN.
Printer	Use a three-character location code (such as the airline city code) followed by the mail stop and the printer type for a printer name. For example, a LaserJet 4 Si would be named PRV-E232-LJ4SI. A duplex LaserJet 4 Si in the same location would be named PRV-E232-LJ4SID.

Table D-1 Object Name and Suggested Standards

Object	Suggested Standards
Print Queue and Print Server	A print queue and print server name should start with the characters PS and PQ. The rest of the name should contain the department server name and a number for each print server or print queue. For example, a print server could be named PS-NCS001-1 and service print queues named PQ-NCS001-1 and PQ-NCS001-2.
Profile	Base profile names on the function supplied by the profile. For example, the profile in a container providing all the required mappings for the departmental users could be named PF-NCSMAP.
Server	Use a set of three-character codes (for the location, division, and server) for a server name. Use the airline city code as the three-character location code. This code is widely recognized because all commercial airports in the world have one. Therefore, a server located in Provo and used by NCS would be named PRV-NCS-001. Server names must be unique. So if you have one server named ACCTG, you cannot have another server with that same name anywhere on the network. (This is a restriction of SAP rather than NDS.)
User	Restrict usernames to eight characters to match the E-mail name length. The E-mail name consists of the first letter of the first name, followed by the last name. For example, Bill Smith becomes BSMITH. If more than one person with the same first initial and last name exists, add the middle initial as the second character of the name. For example, Bill Smith could be BSMITH and Beverly Lynn Smith could be BLSMITH.

Sample Object Property Standards

Following is a sample format that you might use to allow all network supervisors in your organization to enter object names and property information in a consistent manner.

The following examples describe possible standards used for User objects and Organization objects. You should ultimately define standards for all objects.

User Object Property Standards

Use the following information standards in the User object properties.

Account Restrictions Properties

Property	Suggested Standards
Login Restrictions	Determined by the security policy of your organization.
Login Time Restrictions	No standard necessary.
Password Restrictions	Determined by the security policy of your organization.

Environment Properties

Property	Suggested Standards
Default Server	Use the server that the user gets SEND messages from.
Home Directory	Enter both the Volume object name and the pathname.
Language	Use the user’s language.

Identification Page Properties

Property	Suggested Standards
Department	Use the department codes found in the company's telephone directory.
Description	No standard necessary.
E-Mail Address	Use the E-mail format found in the company's telephone directory.
FAX Number	Use the full FAX format found in the company's telephone directory.
Last Name	Enter the last name, followed by a comma, followed by the first name and middle initial. Capitalize only the first letter or initial in each name. Example: Smith, John A
Location	Use the building codes and other geographic information found in the company's telephone directory.
Login Name	Use the first initial and last name of the user. In some cases, the user might have a name where the first initial and last name conflict with another user's. In this case, add a middle initial to distinguish the user. The company's E-mail format uses the appropriate names.
Other Name	No standard necessary.

Property	Suggested Standards
Telephone Number	Use the full telephone format found in the company's telephone directory. A telephone number can consist of the following information, separated by hyphens: **1** Outside code (for outside line) **2** Access code (for long distance) **3** Country code **4** City code **5** Area code **6** Prefix number **7** Main number **8** Extension Note: If an extension is required, insert a space, rather than a hyphen, between the main number and the extension number. For example: 1-801-555-1234 7698
Title	Enter the user's current job title. If this person is an administrator for part of the tree, add ADMIN to the title.

Postal Address Properties

Property	Suggested Standards
Postal Address	Enter the user’s default corporate postal address. Place the mail stop information in the “Post Office Box” field. Use the “Copy to label” to set the mailing label. Use the full name on the mailing label.

Organization Object Property Standards

Use the following information standards in the Organization object properties.

Identification Page Properties

Property	Suggested Standards
Description	Enter a brief description of the Organization’s job functionality.
E-Mail Address	Use the E-mail address of the company or organization (such as an Internet address).
FAX Number	Use the full FAX format found in the company’s telephone directory.
Location	Use the current location names found in the company’s telephone directory. If the organization is located in different geographical areas, put all locations in the field.
Name	Enter the full name of the Organization that is associated with this container. For example, instead of SED, enter “Systems Engineering Division”.

Property	Suggested Standards
Other Name	No standard necessary.
Telephone Number	Use the full telephone format found in the company's telephone directory. A telephone number can consist of the following information, separated by hyphens: **1** Outside code (for outside line) **2** Access code (for long distance) **3** Country code **4** City code **5** Area code **6** Prefix number **7** Main number **8** Extension Note: If an extension is required, insert a space, rather than a hyphen, between the main number and the extension number. For example: 1-801-555-1234 7698

Glossary

Access Control List (ACL) A list that contains information about an object describing which other objects can access it. It is a property of every object in the NetWare® Directory Services™ database. Trustees and the Inherited Rights Filter are contained in the ACL.

Add Self property right (A) Grants a trustee the right to add or remove itself as a value of the property. This right is used only for properties that contain object names as values, such as lists of group members or mailing lists.

ADMIN User object A User object that is created at installation. It has the Supervisor object right to all objects so that it can be used to create the Directory tree.

Alias object An object that points to another object at a different location in the Directory tree. Use it to see an object that you need to use regularly but that is not located in the context that you normally work in.

All Properties option An option you can choose in order to give a trustee specific property rights to all properties at once instead of assigning rights individually to each property. While property rights assigned individually to a property cannot be inherited, rights granted with the All Properties option flow down the Directory tree to objects below.

authentication A means of verifying that a user is authorized to use the network. Authentication works in combination with Access Control to provide network security.

base schema A set of defined object classes.

See also “object classes”.

bindery A network database in NetWare versions earlier than NetWare 4™. The bindery contains definitions for entities such as users, groups, and workgroups.

bindery context The container objects where bindery services is set.

See also “context”.

bindery services A feature of NetWare 4 that allows bindery-based utilities and clients to coexist with NetWare Directory Services on the network, using a subset of the Directory tree as if it were a bindery.

Bindery object An object that was upgraded from a bindery-based server, but that cannot be identified. Bindery-based clients must use older NetWare utilities to access these objects through bindery emulation.

branch A container object and all the objects it holds, which can include other container objects.

Browse object right (B) Grants the right to see the object in the Directory tree. The name of the object is returned when a search is made that matches the object.

child partition A partition that has a Directory tree boundary immediately below another partition.

common name (CN) The name of a leaf object, as displayed in the Directory tree.

Compare property right (C) Allows a trustee to compare the value of a property with another value to see if they are equal. With the Compare right, an operation can return True or False, but you cannot see the value of the property.

complete name See “Distinguished Name.”

Computer object An object that represents a computer on the network.

container object An object that holds, or contains, other objects. Container objects are used to logically organize all other objects in the Directory tree. The three types of container objects are Country, Organization, and Organizational Unit.

context The location of an object in the Directory tree.

Country object (C) An object that designates a country where your network resides and organizes other objects within the country.

Create object right (C) Grants the right to create a new object below the designated object in the Directory tree. This right is available only for container objects.

current context Your current location in the Directory tree.

CX A text workstation utility that allows you to view or change your current context in the Directory tree.

Delete object right (D) Grants the right to delete the object from the Directory tree. To delete a container object, all subordinate objects must first be deleted.

Directory database A database that maintains, stores, and manages Directory objects that consist of categories of information, known as properties, and the data included in those properties.

Directory schema The rules that define how the Directory tree is constructed. The schema define specific types of information that dictate the way information is stored in the Directory database.

directory services Databases of information with powerful facilities for storing, accessing, managing, and using diverse kinds of information about users and resources in computing environments.

See also “NetWare Directory Services (NDS)”.

Directory tree A hierarchical structure of objects in the NetWare Directory Services database. The Directory tree includes container objects that are used to organize the network and leaf objects that represent resources.

Directory tree name A name of 1 to 32 characters assigned during installation to each Directory tree. It can contain upper- and lowercase letters, numbers, hyphens, and underscores, but no spaces or trailing underscores.

Distinguished Name The complete name, or path, from an object to the [Root] of the Directory tree.

See also “Relative Distinguished Name (RDN)”.

distributed database Databases that provide services to all network applications and users across disparate platforms including hosts, minicomputers, and network systems.

dsadmin A utility that enables you to set dynamic, configurable NDS variables.

dsrepair A utility that corrects problems in the NetWare Directory Services database.

effective rights The rights that an object can actually exercise to see or modify a particular directory, file, or object. An object’s effective rights to a directory, file, or object are calculated by NetWare each time that object attempts an action.

fault tolerance A means of protecting data by providing safeguards against hazardous events such as power outages or hard disk crashes.

global login Allows users to log in to the network rather than to individual servers, and to gain access to all network resources.

graphical utilities Allow network supervisors to manage the network through MS Windows 3.x Presentation Manager*.

Group object A leaf object listing several User objects, used to allow collective (rather than individual) network administration.

inheritance The rights granted to a trustee by a trustee assignment. These rights apply to everything below the point where the trustee assignment is made, unless another explicit trustee assignment is made or the rights are blocked by an Inherited Rights Filter.

Inherited Rights Filter (IRF) A filter that is part of every directory, file, and object, controlling which rights a trustee can inherit from parent directories and container objects.

internationalization Allows adaptation of a network for use with multiple languages.

IRF See “Inherited Rights Filter (IRF).”

LAN See Local Area Network (LAN).

LAN driver An NLM program that understands and controls the network board. A LAN driver serves as a link between a station’s operating system and the physical network infrastructure.

leaf object An object that doesn't contain any other objects. Leaf objects are located at the end of a branch in the Directory tree.

local area network (LAN) A network located within a small area or common environment, such as in a building or a building complex.

See also "wide area network (WAN)".

login script A list of commands that are executed when a user logs in to the network. These commands establish a user's network environment.

Three different login scripts can be executed when a user logs in: one from the user's immediate container object, one from a Profile object (if specified for the user), and one from the User object itself.

master replica A writable replica that contains all object information for the partition. All partition operations (create, join, delete, and repair) occur from the master replica of a given partition.

Only one master replica can be defined for each partition.

name type Distinguishes the type of object name of an object (such as O, OU, or CN).

NDS See "NetWare Directory Services (NDS)."

NETADMIN A text utility that allows you to create objects and assign rights and properties.

NetWare Administrator A graphical utility that provides much of the same functionality as the text menu and command line utilities. With NetWare Administrator, you can perform most of the tasks in one utility.

NetWare Directory Services (NDS) An object-oriented implementation of directory services that allows you to build sophisticated naming schemes and databases across network-wide resources.

See also "directory services".

NetWare Services server A computer running the NetWare 4.1/9000 Services operating system software.

network A group of computers that can communicate with each other, share peripherals (such as hard disks and printers), and access remote hosts or other networks.

nwcm A utility that enables you to view and configure NDS parameters, to monitor the internal time on a server, and to ensure that the time reported by all servers across the network is consistent.

object Logical representations of network resources including users, groups, printers, volumes, computers, etc., that make up the Directory tree.

Some objects represent physical entities while others represent logical entities such as groups and print queues.

It is important to note that an object is a structure where information is stored. It is not the entity that it represents.

See also “property”.

object classes A defined list of objects such as servers, users, and print queues used by NDS.

object rights Rights that control access to an object as an entity are called object rights. Object rights control what trustees of an object can do with that object. Object rights do not allow the trustee to access information stored in that object’s properties unless the trustee has the Supervisor object right, which includes the Supervisor property right.

Organization object (O) A container object that helps organize other objects in the Directory tree.

Organizational Role object A leaf object that defines a position or role within an organization. It is used to specify a position that can be filled by different people, such as a Team Leader or Vice President.

Organizational Unit object (OU) A container object, a level below the Organization object, that helps to further organize other objects in the Directory tree.

parent partition A partition that is organizationally above another partition in the Directory tree.

partial name See “Relative Distinguished Name (RDN).”

partition A logical division of the NetWare Directory Services database. A partition forms a distinct unit of data in the Directory tree that is used to store and replicate Directory information.

Each partition consists of a container object, all objects contained in it, and data about those objects. Partitions do not include any information about the file system or the directories and files contained there.

PARTMGR The text workstation utility that can create, modify, and delete partitions and replicas.

Primary time server A time source server that synchronizes the time with at least one other Primary or Reference time server and provides the time to Secondary time servers and to clients.

See also “time synchronization”.

Print Queue object A leaf object that represents the print queue and contains its properties.

Print Server object A leaf object that represents a network print server.

Printer object A leaf object that represents a physical printing device on the network.

Profile object A leaf object that represents a login script that is used by a special group of users who need to share common login script commands.

It can be used for users who are not located under the same container in the Directory tree or who are a subset of users in the same container.

property A characteristic of a NetWare Directory Services object such as name, volume, login name, password restrictions, group membership, etc.

Some properties can contain multiple values, such as multiple telephone numbers.

See also “object”.

property rights Rights that apply to the properties of a NetWare Directory Services object.

protocol Convention or rule used by a program or operating system to communicate between multiple endpoints.

PUBLIC directory A directory on SYS where NetWare utilities and their related files are copied to during installation.

[PUBLIC] trustee A special trustee that can be added to any object, directory, or file. Rights granted to [PUBLIC] are effective for any object in NDS that does not have other effective rights.

Read property right (R) The right to read the values of an object’s properties, assigned on a per property basis.

read-only replica A type of replica that can be read but not written to by any user.

read/write replica A type of replica that can be read and written to by any user. However, it cannot be used for partition operations such as create, join, delete, and rebuild.

Reference time server A time source server that provides a time to which all other time servers and clients synchronize.

See also “time synchronization”.

Relative Distinguished Name (RDN) The context, or path, from an object to another object of the Directory tree.

See also “Distinguished Name”.

Rename object right (R) Allows you to change the name of the object. This changes the value of the naming property. Only the last part of the complete name can be changed with this right.

For example, if you have the Rename object right on a Printer object, you can rename that Printer object so that its complete name changes from CN=HR_Printer.OU=Personnel.O=Novell to CN=Personnel_Printer.OU=Personnel.O=Novell.

replica A copy of a NetWare Directory Services database partition’s information. An unlimited number of replicas can be created for each partition, and they can be stored on any server in the network.

There are three types of replicas: master, read/write, and read-only.

replica list The collection of replica properties of a partition.

replica ring See “replica list.”

root directory The highest directory level in the NetWare file system hierarchical directory structure. With NetWare, the root directory is at the volume level and all other directories are subdirectories of the volume.

[Root] object An object in the Directory tree whose purpose is to provide a highest point to access different Country and Organization objects, and to allow trustee assignments granting rights to the entire Directory tree.

root partition The first partition that is created (at the top of the tree) which includes the [Root] object.

SAP See “Service Advertising Protocol (SAP).”

schema See “Directory schema.”

Secondary time server A time server that obtains the time from a Single Reference, Primary, or Reference time server and provides the time to clients.

See also “time synchronization”.

Server object A leaf object that represents a server. Information about its location can be stored in its properties.

Service Advertising Protocol (SAP) A protocol that provides a way for services to advertise on a NetWare internetwork.

Single Reference time server A time source server that provides time to Secondary time servers and to clients. It is the sole source of time on the network.

See also “time synchronization”.

subordinate reference replica A type of replica that is automatically placed on a server if the parent Directory partition has a master, read/write, or read-only replica and the child Directory partition does not. Subordinate replicas cannot be modified.

subtree A branch of a Directory tree partition.

Supervisor object right (S) Grants all access privileges to an object. A trustee who has the Supervisor right automatically has access to all properties of an object.

The Supervisor right can be blocked by the Inherited Rights Filter, both for objects below the object where Supervisor is assigned and for individual properties of an object.

Supervisor property right (S) Grants all access privileges. A trustee who has the Supervisor right automatically has all other rights to the property.

The Supervisor right can be blocked by the Inherited Rights Filter, both for objects below the object where Supervisor is assigned and for individual properties of an object.

synchronization A means of ensuring that replicas of a Directory partition contain the same information as other replicas of that partition. Replica synchronization updates the replicas and runs periodically at a cycle controlled by the network supervisor.

See also “replica”.

text utilities One of the two main types of utilities available, the other being graphical utilities. There are two categories of text utilities: command line utilities and menu utilities.

time server A server which provides time to the system. There are four types of time server: Primary, Reference, Secondary, and Single Reference.

See also “time synchronization”.

time source server The server that provides time to the network. These are three types of time source servers: Single Reference, Primary, and Reference.

time stamp A unique code that identifies an event and includes the time it occurred. It is reported by the Directory tree at the time of an event such as a password change. NetWare Directory Services uses this to establish event order, record real-world times, and set expiration dates.

time synchronization A method of ensuring that all servers in a Directory tree report the same time.

In systems with a Single Reference time server or a Reference time server, all other servers synchronize to them.

Primary and Secondary time servers synchronize with other Primary or Reference time servers and provide time to Secondary time servers.

tree See “Directory tree.”

tree name See “Directory tree name.”

trustee A user or group that has been granted rights to work with a directory, file, or object.

See also “trustee assignments”.

trustee assignments Rights granted to an object to perform actions on another object or its properties, on a file, or on a directory.

In the NetWare Administrator utility, trustee assignments granting rights to an object can be viewed by selecting the object and choosing “Trustees” from the “Object menu.” Trustee assignments are stored in the Access Control List (ACL) property of every object.

tsadmin A utility that enables you to force an immediate time synchronization with the network.

typeful name The object name that includes the name type (OU, O, etc.) of each object when identifying the Distinguished Name of that object.

typeless name The object name that excludes the name type (OU, O, etc.) of each object when identifying the Distinguished Name of that object.

UIMPORT A text utility that allows the network supervisor to import User objects from an existing database.

User object A leaf object that represents a person who uses the network. Its properties can store information such as a telephone number, address, group membership, etc.

value The contents of an object property. Many properties can have multiple values, such as a telephone number property containing three different telephone numbers. Each telephone number is a value of the property.

Access rights control access to a property, but not to individual values of a property.

Volume object A leaf object that represents a physical volume on the network. Its properties can store information about its location, owner, space use restrictions, etc.

WAN See “wide area network (WAN).”

wide area network (WAN) A network that communicates over a long distance, such as across a city or around the world. It can be comprised of or incorporate one or more local area networks.

See also “local area network (LAN)”.

Write property right (W) Allows a trustee to add, change, or remove any value of a property. If the Write right is given, Add Self is disabled because Write includes its functionality.

Index

Index

Index

Index

Index

Printed in USA 12/96

Manufacturing Part No.
J2768-90005
Mfg. number is for HP internal use only

**Customer Order No.
J2768-90005**