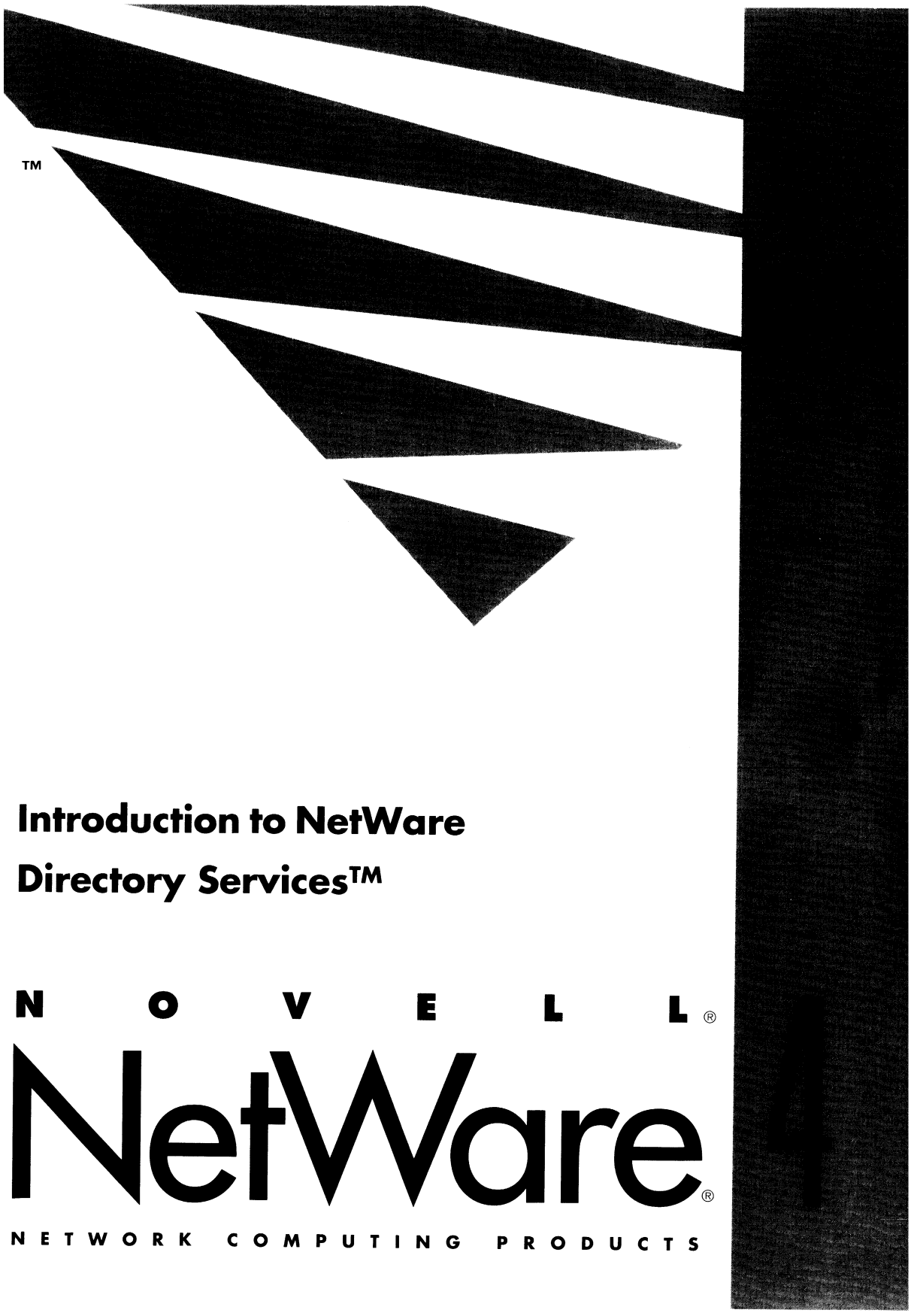


Introduction to NetWare Directory Services™

NOVELL®

NetWare®

NETWORK COMPUTING PRODUCTS

disclaimer

Novell, Inc. makes no representations or warranties with respect to the contents or use of this manual, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any NetWare software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of NetWare software, at any time, without any obligation to notify any person or entity of such changes.

trademarks

Novell, NetWare, and the N-Design are registered trademarks and the NetWare Logotype is a trademark of Novell, Inc.

A complete list of trademarks and their respective owners appears in "Trademarks" on page 199.

Novell, Inc.
122 East 1700 South
Provo, UT 84606
U.S.A.

Introduction to NetWare Directory Services
December 1994
Part Number 100-002069-001

WCP.....95
8 9 2 0 8

Contents

How to Use This Manual

Introduction vii
Documentation Conventions vii
- Asterisk (*) vii
- Commands viii
- Delimiter Bar (|) viii
- DOS Commands viii
- DOS Filenames, Directory Names, and Pathnames viii
- Ellipses ix
- Emphasis ix
- Icons ix
- Key Names x
- Options x
- Square Brackets x
- System Response x
- Variables x

Supplemental Documentation xi
Online Help xii
Additional Help Resources xiii
User Comments xvi

Concepts

Overview 1
Contents 1

1 Understanding NetWare Directory Services

Overview 3
What are Directory Services? 3
- Standard Directory Services 3
- NetWare Directory Services 4

The Directory Tree 6
Hierarchical Tree Structure 6
Directory Schema 7
Directory Objects 8
Context and Names 22
Name Context 25
Common Names 25
Name Types 26
Object Naming Rules 27
Where to Go from Here 29

2 Understanding Management Features

Overview 31
User Object ADMIN 32
Directory Partitions 34
Partition Replicas 36
Purpose 36
Types 37
Directory Synchronization 38
Management Utilities 39
Where to Go from Here 39

3 Understanding Bindery Services

Overview 41
Setting Up Bindery Services 43
Created Objects 43
Inaccessible Information 44
Limited Partitioning 44
Setting Bindery Context 45
In a Single-Level Directory Tree 45
In a Multiple-Level Directory Tree 46
For a Specific Server 47
For Multiple Servers in the Same Bindery Context 49
For Objects in Different Bindery Contexts 51
Where to Go from Here 52

4 Understanding Time Synchronization in NDS

Overview 53
Time Stamps 54
Time Servers 55
Single Reference 55
Primary 57
Reference 58
Secondary 60
Time Source Server Functions 60
SAP (Service Advertising Protocol) 61
Custom Configuration 61
Choosing a Time Synchronization Method 62
Where to Go from Here 67

Planning, Implementing, and Managing

Overview 69
Contents 69

5 Planning NetWare Directory Services Implementation

Overview 71
Guidelines for Implementing NDS 73
Creating Naming Standards 75
Creating Directory Tree Maps 76
Developing an Implementation Strategy 79
Planning a Departmental Directory Tree (Merge Tree Method) 79
Planning an Organizational Directory Tree 82
Organizing Objects into a Logical Hierarchy 83
Planning the Directory Tree Levels 84
Placing Container Objects in the Directory Tree 85
Placing Leaf Objects in the Directory Tree 90
Directory Tree Planning Examples 91
Developing a Replication Strategy 95
Providing Fault Tolerance 95
Decreasing WAN Link Traffic 96
Developing a Time Synchronization Strategy 99
Developing a Security Strategy for the Directory Tree 100
Trustee Assignments 100
Container Rights 101
Group Object Rights 101
Inherited Rights Filter 101
Security Equivalency 102

Developing an Integration Strategy for Bindery Services 103
Managing Bindery Services 104
Where to Go from Here 105

6 Implementing NetWare Directory Services

Overview 107
Introduction 107
Completing General Tasks and Guidelines for All Networks 108
Implementing NDS on Various Sizes of Networks 114
Small-Sized Network 114
Medium-Sized Network 117
Large-Size Networks 122
Additional Information 129

7 Managing NetWare Directory Services

Overview 131
Introduction 132
DSMERGE 132
Using the DSMERGE Utility 133
Completing the Tree Merge 134
Additional Information 135
DSREPAIR 135
Using the DSREPAIR Utility 137
Additional Information 138
DSTRACE 138
Using the DSTRACE Utility 138
Additional Information 139
INSTALL 139
Using the INSTALL Program 139
Removing NDS with the INSTALL Program 140
Additional Information 141
NETADMIN 141
Using the NETADMIN Utility 142
Additional Information 143
NetWare Administrator 144
Using the NetWare Administrator Utility 144
Additional Information 145
PARTMGR 146
Using the PARTMGR Utility 146
Additional Information 147

SET (NDS Parameters) 147
Using the SET Utility (NDS Parameters) 148
Additional Information 149
TIMESYNC 149
Using the TYMESYNC Utility 149
Additional Information 150
UIMPORT 151
Using the UIMPORT Utility 151
Additional Information 151

Appendixes

Overview 153
Contents 153

A NDS Object Classes and Properties

Overview 155
NDS Object Classes and Their Functions 156
NDS Object Classes and Their Properties 158

B Referencing and Using Leaf Objects

Overview 167
User-Related Leaf Objects 168
Server-Related Leaf Objects 170
Printer-Related Leaf Objects 171
Messaging-Related Leaf Objects 172
Informational Leaf Objects 173
Miscellaneous Leaf Objects 174

C Creating a Standards Document for NDS Object Classes and Properties

Overview 175
Sample Object Naming Standards 176
Sample Object Property Standards 178
User Object Property Standards 178
Organization Object Property Standards 181

Glossary

Glossary 183

Trademarks

Novell Trademarks 199
Third-Party Trademarks 200

Index

Index 201

How to Use This Manual

Introduction

Introduction to NetWare Directory Services has two purposes:

- To help you understand the features provided by NetWare® Directory Services™ (NDS) technology
- To help you plan the implementation of the NDS™ technology on your network

This manual is directed to network supervisors responsible for planning and implementing the NetWare 4.1 software and NDS technology on the network.

You should read this manual before you upgrade to or install NetWare 4.1 software.

Documentation Conventions

This manual uses the following Novell® conventions.

Asterisk (*)

An asterisk denotes a trademarked name belonging to a third-party company. Novell trademarks are denoted with specific trademark symbols (®, ™, etc.).

An ownership listing of all (Novell and third-party) trademarks cited in a manual can be found either on the disclaimer page in the front or in a "Trademarks" section at the back of printed manuals. A trademarks list is also available in the DynaText* online documentation.

Commands

Boldface characters indicate items that you type, such as commands and options. You can use any combination of uppercase and lowercase letters.

For example:

```
C:\A INSTALL
```

Delimiter Bar (|)

In syntax examples, a delimiter bar separating two command options indicates that you can choose one of the options.

For example:

```
-S | -R
```

Do *not* type the bar.

DOS Commands

DOS commands and command option letters are shown in uppercase letters. For example: FTPD.

Because DOS is not case-sensitive, you can type DOS commands in uppercase or lowercase letters.

DOS Filenames, Directory Names, and Pathnames

DOS filenames, directory names, and pathnames are shown in uppercase letters. For example, AUTOEXEC.BAT.

Because DOS is not case-sensitive, you can type these names in uppercase or lowercase letters.

Ellipses

Ellipses in syntax examples indicate that parameters, options, or settings can be repeated.

For example, in the command

LOGIN SERVER1/SUPERVISOR /*option*...

you could replace *option* with any number of available options.

Emphasis

Italic type indicates emphasized text. For example:

Remember to load the driver *before* you install the application.

Icons

Checklists, which often contain prerequisites, are marked with the "Checklist" icon to the left of this text.

Procedures to follow in order to accomplish a specified task are marked with the "Procedure" icon to the left of this text.

Additional or "nonessential" but noteworthy information is marked with the "Note" icon to the left of this text.

Vital information about system or software requirements, etc., that deserves particular attention is marked with the "Important" icon to the left of this text.

Guidelines or tips about fine-tuning, optimizing, etc., which might be applicable to your site or situation but maybe not to all, are emphasized with the "Suggestion" icon to the left of this text.

Warnings about potential danger to data, hardware, or person are emphasized with the "Warning" icon to the left of this text.

Key Names

Angle brackets surround the name of a key. For example, <Enter> corresponds to the Enter key on your keyboard. <Ctrl>+<c> means hold down the Ctrl key and simultaneously type the letter c (in lowercase, in this case).

Options

In syntax examples, braces indicate that you are required to choose one of the enclosed options. For example, the following notation means that you must include a 0 or a 1 in the command:

```
{0, 1}
```

Square Brackets

In syntax examples, boldface type enclosed in square brackets indicates command options that you can type as needed. For example:

```
FTP [ -D ] [ -F ]
```

System Response

Monospace type shows system-generated responses that appear on your workstation screen. For example:

```
TNVT220>
```

Variables

Italic type indicates variables—descriptive item names, such as command parameters—that you replace with appropriate values.

For example, in the command:

FTP -F *remote_host*

you type the name of a computer on your network in place of *remote_host*.

Supplemental Documentation

The following publications provide supplemental information specifically related to NetWare Directory Services technology and software.

- "An Introduction to NetWare Directory Services," *Novell Application Notes*, Apr 93 (Novell part no. 164-000032-004).
- "Implementing Naming Standards for NetWare Directory Services," *Novell Application Notes*, Feb 94 (Novell part no. 164-000036-002).
- "Migrating to NetWare 4.0: An Example," *Novell Application Notes*, Apr 93 (Novell part no. 164-000032-004).
- "NetWare 4.0 Bindery Emulation: An Overview," *Novell Application Notes*, Jun 93 (Novell part no. 164-000032-006).
- "NetWare 4.0 Performance Tuning and Optimization: Part 1," *Novell Application Notes*, May 93 (Novell part no. 164-000032-005).
- "NetWare 4.0 Performance Tuning and Optimization: Part 2," *Novell Application Notes*, Jun 93 (Novell part no. 164-000032-006).
- "Novell's Corporate-Wide Upgrade to NetWare 4," *Novell Application Notes*, Jan 94 (Novell part no. 164-000036-001).
- "Planning a NetWare 4.0 Directory Tree," *Novell Application Notes*, Apr 93 (Novell part no. 164-000032-004).
- "Planning for NetWare 4.0 Installation, Server Migration, and Coexistence," *Novell Application Notes*, Apr 93 (Novell part no. 164-000032-004).
- "Time in the NetWare Environment," *Novell Application Notes*, Jan 94 (Novell part no. 164-000036-001).
- "Time Synchronization in NetWare 4.x," *Novell Application Notes*, Nov 93 (Novell part no. 164-000032-011).
- "Understanding NetWare Directory Services Rights," *Novell Application Notes*, Apr 93 (Novell part no. 164-000032-004).

- "Upgrading to NetWare 4: The Chase Manhattan Bank's CC and FMI Groups," *Novell Application Notes*, Jan 94 (Novell part no. 164-000036-001).

- "Using the DOS Requester with NetWare 4.0," *Novell Application Notes*, Apr 93 (Novell part no. 164-000032-004).

Online Help

- **Context-sensitive help.** If you are using a NetWare menu utility and want more information about how to complete a task, press <F1>.

  If you are unsure how to use a command, type the command name and add the /? option for help. For example, for help with the RIGHTS command, type "RIGHTS /?".

- **Online MS Windows help.** Microsoft* (MS) Windows help viewer allows you to read NetWare help developed for the MS Windows environment. To access the NetWare help screens within MS Windows, press <F1> or the "?" button.

- **DynaText online documentation.** The DynaText viewer allows you to read NetWare documentation from your DOS, MS Windows, Macintosh*, UNIX®, or OS/2* workstation.

  All NetWare 4™ and 3.12 documentation except the *Quick Access Guide* are available on the *NetWare Online Documentation* CD-ROM.

Additional Help Resources

- **Customer service.** You can contact your Novell Authorized Reseller[CLM] representative for technical assistance.

  Most Novell Authorized Resellers have Certified NetWare Engineer[SM] representatives on their staffs ready to assist users with their networking problems.

- **Novell Authorized Service Center[SM] (NASC) locations.** NASC[SM] facilities are local support providers authorized and supported by Novell. They provide both telephone and on-site assistance, and should be your first source for technical support.

  For the Novell Authorized Service Center nearest you, in the U.S. and Canada call 1-800-338-NASC.

- **Hardware documentation.** Many network problems occur because of malfunctioning hardware.

  If you can isolate a problem to a certain computer component or cable segment, check the manuals that came with the hardware involved.

- **NetWare Management System™ (NMS) services.** NMS™ services help you manage the cabling system, computers, software, and other components of the network.

  For more information about using NMS on your network, contact your Novell Authorized Reseller.

- **Other Novell publications.** *Novell Application Notes* and the Novell Research Reports™ publications cover technical aspects of NetWare-based system design, implementation, and management.

  Application Notes is a collection of technical articles published monthly. Research Reports is published as the research becomes available.

  To purchase subscriptions and back issues of these publications from within the United States or Canada, call the Novell Research Order Desk at 1-800-UPDATE1. From other locations, call 801-429-5380.

- **Third-party books and periodicals.** Books on NetWare, including books published by Novell Press™ publishing, are available at most bookstores.

  In addition, numerous networking periodicals give advice on configuring, managing, and troubleshooting your network.

- **NetWire® forum on the CompuServe* bulletin board.** A fairly inexpensive way to get up-to-date advice and patches is through NetWire on the CompuServe bulletin board.

  To open a CompuServe account, call one of the following numbers and ask for "Representative 200":

  - In the United States or Canada: 1-800-524-3388
  - In the United Kingdom: 0800-289-378
  - In Germany: 0130-37-32
  - In other European countries: 44-272-255-111
  - In locations other than the United States, Canada, or Europe, use the appropriate country code for the U.S. and call 614-457-0802. Ask for "Representative 200." This phrase identifies you as a Novell customer.

- **Technical Support Database and NetWire forum on the Internet.** The Novell FTP sites support access through FTP, Gopher, and World Wide Web (WWW) systems. Over 9,000 documents exist on the WWW system for providing technical hints and information.

  To access the Novell Internet sites, log in as ANONYMOUS and use your E-mail address as your password.

  Contact one of the following site addresses:

  In the United States: ftp.novell.com
  In Germany: ftp.novell.de
  In the United Kingdom: ftp.salford.ac.uk
  In Canada: novell.nrc.ca

  See public areas in these sites for possible listings of other sites' addresses.

- **FaxBack Service.** Novell provides a FaxBack Service for obtaining additional product information to help with support needs.

  To access the Novell FaxBack Service, complete the following steps.

  - Within the continental United States

    1. Dial 1-800-NETWARE (1-800-638-9273).
    2. Press #1 (the "Presale Product Information and Upgrade Information" option).
    3. Again press #1 (the "Receive Product Information via Fax" option).

  - Outside the continental United States

    Dial 1-801-429-2772. You are connected directly to the FaxBack Service.

  Follow the directions provided on the phone. You are prompted to enter a document number and then a fax number to send the document to.

- **Network Support Encyclopedia Professional Volume℠ (NSE Pro) package.** This encyclopedia gives customers access to regularly updated information on products and services—plus patches, fixes, and more—from Novell and other vendors.

  The NSE Pro℠ package is distributed on CD-ROM on a subscription basis. Updates are sent out several times each year. More information is available on NetWire or from your Novell Authorized Reseller.

- **Troubleshooting hardware and software.** Specialized hardware and software packages, such as the Novell LANalyzer® software, are available to help you isolate network problems.

User Comments

We are continually looking for ways to make our products and our documentation as easy to use as possible.

You can help us by sharing your comments and suggestions about how our documentation could be made more useful to you and about inaccuracies or information gaps it might contain.

Submit your comments either by filling out the "User Comments" form at the end of this document or by writing to us directly at the following address:

Novell, Inc.
Technical Publications MS C-23-1
122 East 1700 South
Provo, UT 84606 USA

We appreciate your comments.

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 are discussed on the indicated pages:

Purpose	Chapter	Page
To learn more about NetWare Directory Services features in NetWare 4™	Chapter 1, "Understanding NetWare Directory Services"	3
To learn more about management features in NetWare Directory Services	Chapter 2, "Understanding Management Features"	31
To learn more about bindery services in NetWare Directory Services	Chapter 3, "Understanding Bindery Services"	41
To learn more about time synchronization in NetWare Directory Services	Chapter 4, "Understanding Time Synchronization in NDS"	53

chapter 1

Understanding NetWare Directory Services

Overview

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

The following topics are discussed on the indicated pages:

Topic	Page
What are Directory Services?	3
The Directory Tree	6
Context and Names	22

To understand the technology and functionality provided in the NDS™ software, you must first understand some of the basic features of directory service 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 resource of directory services has been available to applications and users alike.

Users and organizations within computing environments have begun to recognize 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.

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

Table 1-1
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	NDS provides the architecture that allows you to build a full range of 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, and 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. However, there is some traffic maintained to allow for synchronization of NDS.
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. The synchronization of the Directory replicas is automatic and does not require any administrative intervention.
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.

Table 1-1 *continued*

Features and Benefits Provided by NetWare Directory Services

Feature	Benefit
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 NetWare 2 and NetWare 3™ 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 (NetWare 2 and NetWare 3) server and resources in the network. NDS also provides an external synchronization process for exchanging information with other naming and directory services, which allows for a seamless sharing of information between NDS and other service providers, such as those providing services on the information superhighway.
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 Directory Tree

NetWare Directory Services was developed as a hierarchical design with multilevels 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.

Hierarchical Tree Structure

NetWare Directory Services (NDS) is consistent with the emerging international standard, X.500. The X.500 specification was developed by the IEEE (Institute of Electrical and Electronic Engineers) 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 available on the information superhighway is being done according to the X.500 specification.

Much of the current development for accessing services available on the information superhighway is being done according to the X.500 specification. NetWare Directory Services (NDS) is consistent with this emerging international standard.

The X.500 specification was developed by the IEEE 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.

Directory Schema

The *Directory schema* is 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.

The following information is defined by the schema:

- **Attribute Information**. Describes what type of additional information an object can or must have associated with the object. Attribute types are defined within the schema by specific constraints and a specific syntax for their values.

- **Inheritance**. Determines which objects will 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, thus identifying and showing an object's location in the Directory tree.

The basis for all entries in an NDS database is a set of defined object classes referred to as the *base schema*. Object classes such as servers, users, and print queues are some of the base object classes defined by the base schema.

For a complete list of the base object classes, as well as other Directory information, see Appendix A, "NDS Object Classes and Properties," on page 155 for more information.

The NDS schema can be modified and expanded to suit the specific needs of your organization. Object class definitions can be added to and modified for the existing base schema.

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 types of objects:

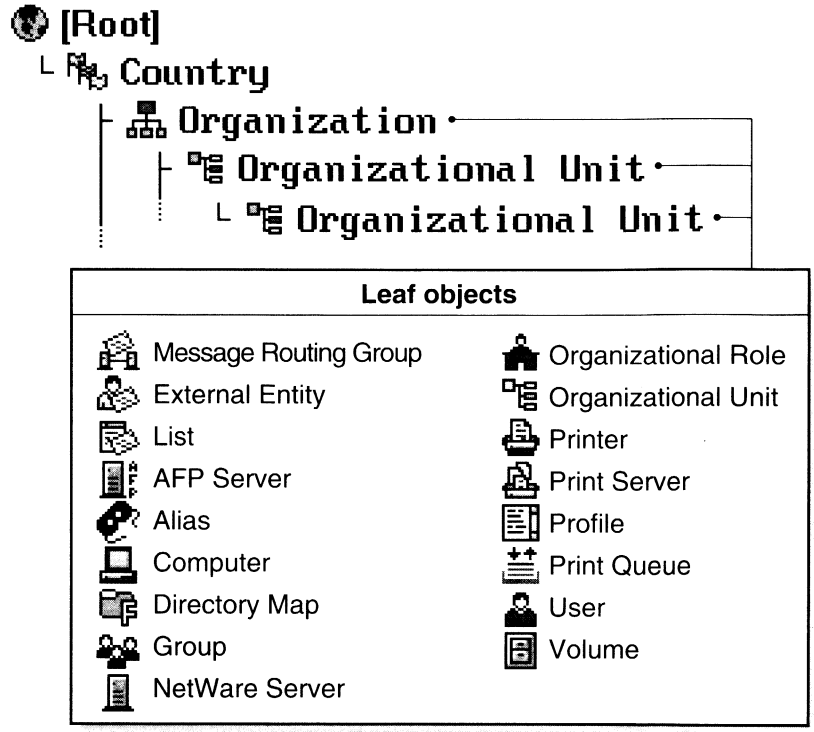
- [Root] object (Directory tree name)

- 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.)

Figure 1-1
Hierarchy of Directory Objects



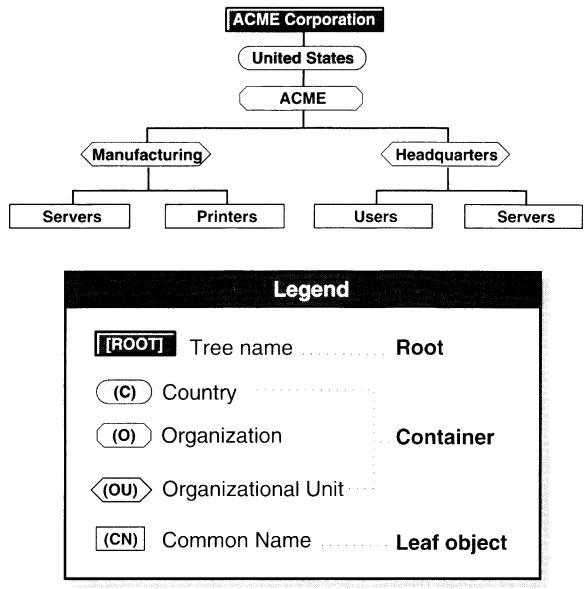
These objects represent both actual and logical resources in the network, such as users and printers, or groups and print queues.

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.

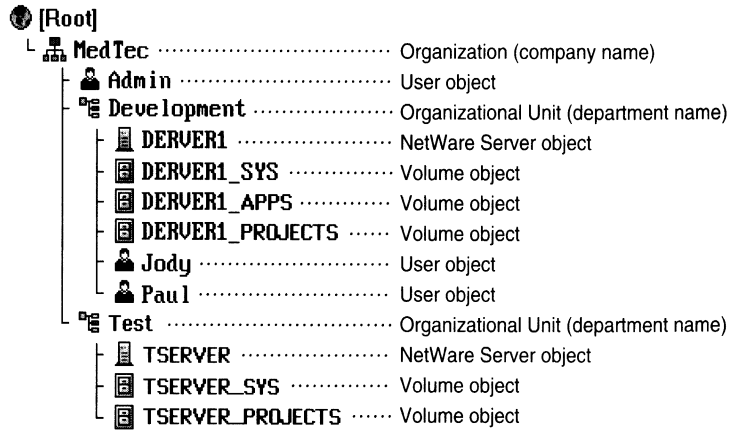
Figure 1-2
Objects Used in a Directory Tree



The Directory tree name ([Root] object) is automatically placed at the top of the tree by the NetWare 4 installation program. Branches of the Directory tree consist of container objects and all of the objects they hold. These container objects can also contain 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 1-3
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 4 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 the rights granted to these trustees flow down the tree. 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, so that it can be used to initially log in and set up the tree. See "User Object ADMIN" on page 32 for more information.

The [Root] object can also be a trustee. However, you should give careful consideration before making [Root] a trustee of another 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" on page 21 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 items in the Directory tree.

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

There are four kinds of container objects:

- Country
- Locality
- Organization
- Organizational Unit

NDS support of Country and Locality as container objects provides useful class definitions for organizing and naming objects within a Directory tree that are represented by countries or regions of your organization.

However, Directory tree structures based on centrally located organizations might not benefit from the added level of complexity.

NDS container objects are defined as follows:

- **Country (C)**. A level below the [Root] object, the Country object designates the countries where your network resides and organizes other objects within the country.

  This object is optional.

  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 a part of your network.

  Normally, you need to create a Country (C) object if you have a global network that spans multiple countries or you plan to participate in the information superhighway.

The Country object is not part of the NetWare 4 default server installation; that is, you are not prompted for a Country object when you install NetWare 4 software.

Nevertheless, you can create a Country object during the server installation. See "Install Server Software," in Chapter 2, *Installation* for more information on installing a server.

- **Locality (L)**. A level below the [Root] object, Organizational (O) object, or Organizational Unit (OU) object, the Locality (L) object designates the location where this portion of your network resides and organizes other objects within the location.

  This object is optional.

  You can use a Locality object 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 (C), Organization (O), and Organizational Unit (OU) objects. They can also hold Organization (O) and Organizational Unit (OU) objects.

The Locality object is not part of the NetWare 4 default server installation; that is, you are not prompted for a Locality object when you install NetWare 4 software.

Nevertheless, you can create a Locality object during the server installation. See "Install the first server and set up the Directory tree." on page 109 for more information.

- **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 container.

  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 to organize leaf objects in the Directory tree. It also allows you to set defaults in a login script and to create a user template for User objects you create in the Organizational Unit container.

  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.

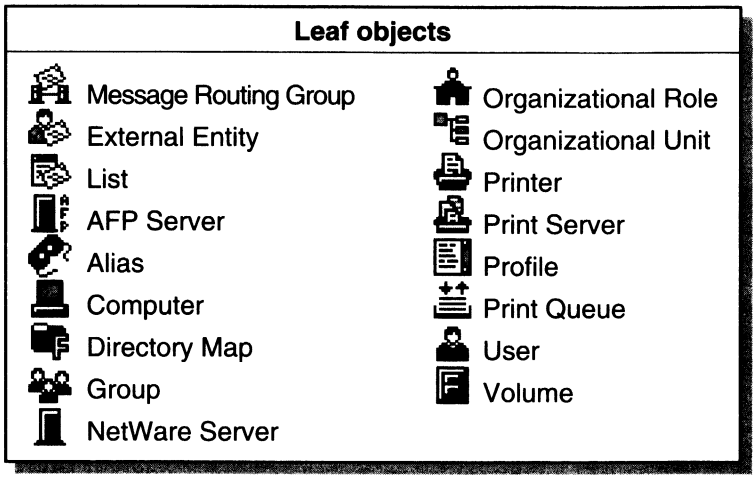
  This object is optional. When used, Organizational Units must be placed directly below an Organization, another Organizational Unit, or a Locality object.

Leaf Objects

Directory leaf objects are objects that do not contain any 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.)

Figure 1-4
Leaf Objects You Can Create



See Appendix B, "Referencing and Using Leaf Objects," on page 167 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 the 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 properties are required for a specific object 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@Novell
	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.

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. When the area codes are found, the utility returns a list of all the objects with that area code in their properties.

Or, you can request information on a specific object. The utility searches only for that object, and you receive information on that object's properties that you have access to.

To make searching for an object property easier, enter information for the optional properties when creating container and leaf objects. Entering information in objects' properties can help you track and manage those objects.

Also, if you define properties using a consistent format, after you have created the objects, you can use the NetWare Administrator, NETADMIN, or NLIST utility to search for and list these objects. You can also search for their various properties.

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," "BLDG M1," and "M1" as values in the Location property of multiple User objects.

Standardizing the value for the Location property for all User objects at the site (such as M1, M2, and M3) makes it possible 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 very limited access levels to bindery objects existent in NetWare 2 and NetWare 3 networks. NetWare 4 adds NDS object and NDS property rights, which determine what you can do within the Directory tree.

Because the Directory tree is a hierarchical tree 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 provided by the Inherited Rights Filter (IRF). An IRF is a list of rights that can be assigned for any object in a lower container than a parent container within the tree hierarchy. It controls the rights that a trustee can inherit from container objects. See "Inherited Rights Filter" on page 21 for more information.

To provide better access control to the pieces of information (properties) contained in NDS objects, object and property rights are assigned separately.

Object Rights

Object rights control what trustees of an object can do with that object. Object rights control the object as a single entity in the Directory tree, but do not allow the trustee to access information stored in that object's properties (unless the trustee has the Supervisor object right, which also includes the Supervisor property right).

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

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 1-2
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 for property objects 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	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. The Write right for all existing object properties is also needed to delete objects.
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 "Name Types" on page 26.

Property Rights

While object rights allow you to see an object, delete an object, create a new object, etc., only the Supervisor property right allows you to see the information stored in an object's properties.

To see the information in an object's properties, you must have the correct property rights. Property rights control access to each property in an object.

Property rights apply only to NDS object properties, 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 anyone else from seeing the specified telephone number by using an Inherited Rights Filter (IRF) to disable the Read right to that particular property (see "Inherited Rights Filter" on page 21). At the same time, you can still allow the person to view other properties, such as the user's address.

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

Table 1-3
Property Rights

Right	Description
Add or Delete Self	Allows you to add or remove yourself as a value of the property, but you cannot change any other values of the property. 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. This right is included in the Write right; that is, if the Write right is given, Add or Delete Self operations are also allowed.
Compare	Allows you to compare any value to an existing value of the property. The comparison can return True or False, but cannot give the value of the property.

Table 1-3 *continued*
Property Rights

Right	Description
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" on page 21 for more information.
Write	Allows you to add, change, or remove any values of the property. 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. The Write right to the Access Control List (ACL) property is the same as giving the Supervisor right to the object—the right to grant rights.

Access Control List

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

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

Each object 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 object 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 spreading from one object to another.

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

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

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

Effective Rights

The combination of inherited rights, trustee assignments in an ACL, and Security Equal To (lists the rights other objects have within a container that the User object has equal security to) are known as effective rights.

An object's effective rights are what control its access to another object and that object's properties.

See "Effective Rights" in *Concepts* for more information.

Security Equal To

A property of every User object that lists the rights other objects have within a container compared to the rights of that User object. The user is granted all rights that any object (such as a User, Group, or Printer object) in that list is granted, both to objects and to files and directories.

Use the Security Equal To property to give a user *temporary* 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 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 Equal To" 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. This context is important for NDS to locate specified network resources.

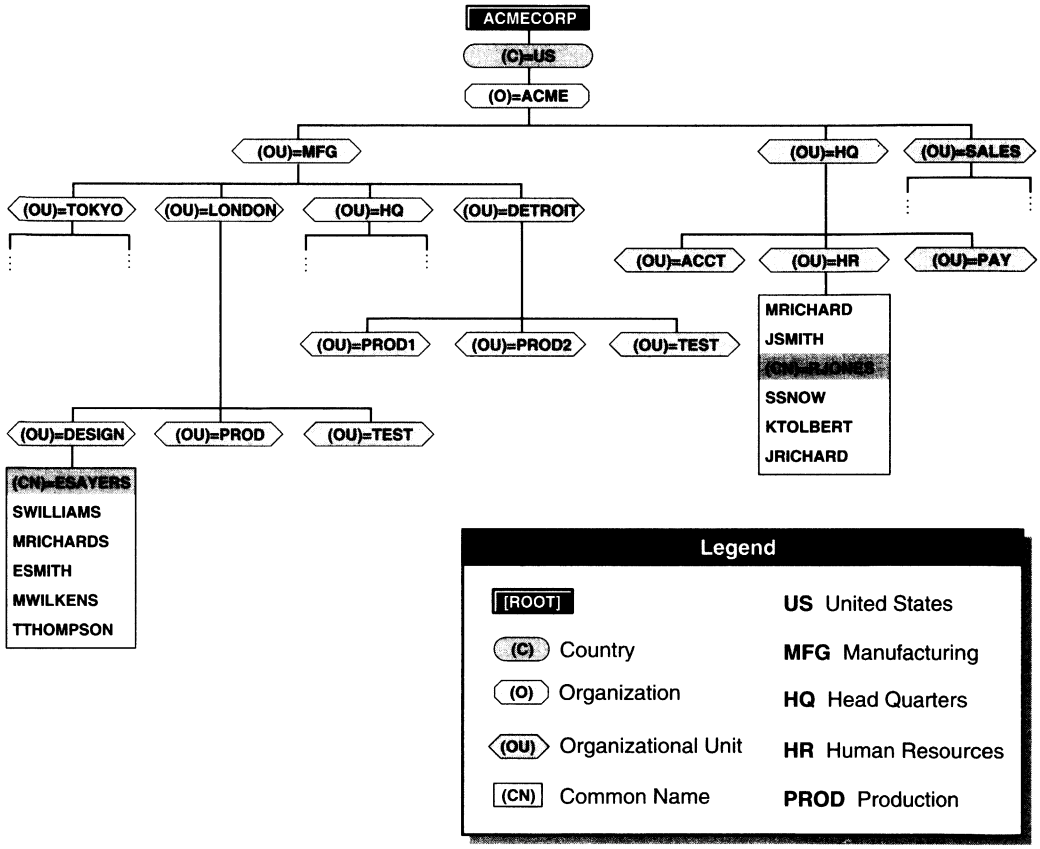
The complete context, or path, from an object to the [Root] of the Directory tree identifies and forms the object's *Distinguished Name*. The context, or path, from an object to another object of the Directory tree identifies and forms the object's *Relative Distinguished Name (RDN)*.

For example, in the following figure, the context for the User object ESAYERS is OU=DESIGN.OU=LONDON.OU=MFG.O=ACME.C=US and the Distinguished Name would be ESAYERS.DESIGN.LONDON .MFG.ACME.US. The context for the User object RJONES is OU=HR.OU=HQ.O=ACME.C=US and its Distinguished Name would be RJONES.HR.HQ.ACME.US.

The Relative Distinguished Name for the User object RJONES in relation to the Organizational Unit (OU) SALES is RJONES.HR .HQ.SALES.

Figure 1-5

Context in a Directory Tree



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

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

  Since it is difficult for some users to remember long Distinguished Names with multiple layers of Organizational Unit (OU) objects, you might choose to have no more than two or three levels of OU objects.

- Maintain short names within the hierarchy.

  Because each object is identified by its relative 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.

  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.

  When naming objects, you can use spaces in the name. But spaces appear as underscores in some utilities.

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

Name Context

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 mutual *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 certain 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 Directory 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" on page 19 for more information.

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

Common Names

All leaf objects in the Directory have a *common name* (CN). 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.

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 when identifying the Distinguished Name of that object. A typeless name excludes the name type for each object in a name.

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

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

is expressed with the 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, and O is the Organization name.

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.

All Distinguished Names must be unique within a Directory tree. In addition, all container and object names must be unique within that container. The NDS database recognizes only one common name of the same name within each container.

Object Naming Rules

- The name must 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 any special characters. But if the object needs to be accessed from a client workstation running a version of NetWare earlier than NetWare 4, you should avoid using special characters.

  For a list of these special characters, see "Naming Restrictions for Bindery Services" on page 28.
- Object names are displayed with uppercase and lowercase letters as they were first entered, but they are not case-sensitive. Therefore, "ManagerProfile" and "MANAGERPROFILE" are considered to be identical names.
- Spaces and underscores can be used, and they are displayed as spaces. Therefore, "Manager_Profile" and "Manager Profile" are considered to be identical names.

  If you use a space in a name, you must place quote 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.

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 first NetWare Server object for a NetWare 4.1 server must be created with INSTALL. The object is given the same name as the physical server. Rules for naming physical servers are in the <F1> Help of INSTALL.

- If you create a NetWare Server object for a server other than a NetWare 4.1 server, you must use the physical server name as well, because NetWare Directory Services must search for the server on the network to verify its existence.

Because of these restrictions, we recommend renaming Server objects by changing their names in the AUTOEXEC.NCF file.

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

Naming Restrictions for Bindery Services

When you create objects to be accessed from a client workstation running the NetWare Client shell software, such as NETX, the names of the objects must follow bindery naming rules or the NetWare Client shell software cannot recognize them. Object names in bindery services are interpreted 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 client running a version of NetWare earlier than NetWare 4:

/ 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 3, "Understanding Bindery Services," on page 41 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 (❑) in MS Windows.

Substituted characters can prevent NDS from recognizing an object. See "Code Page" and "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 2, "Understanding Management Features," on page 31
Use bindery services with NDS	Chapter 3, "Understanding Bindery Services," on page 41
Use time synchronization with NDS	Chapter 4, "Understanding Time Synchronization in NDS," on page 53
Plan, manage, and implement NDS	"Planning, Implementing, and Managing," on page 69

chapter 2

Understanding Management Features

Overview

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

The following topics are discussed on the indicated pages:

Topic	Page
User Object ADMIN	32
Directory Partitions	34
Partition Replicas	36
Directory Synchronization	38
Management Utilities	39

Management of the NDS™ architecture includes the creation and management of objects and the distribution of the 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 databases within your network.

User Object ADMIN

The very first time the network supervisor logs in, it is as the User object ADMIN—the single User object created by default during the NetWare 4™ software installation. However, NetWare 4 does not require you to limit yourself to one network supervisor.

When the ADMIN object is created on the first NetWare 4 server you install, it is granted all rights (including the Supervisor right) to every object and property in the Directory tree. This gives ADMIN complete control of the tree until you assign other User objects explicit rights and delete or mask ADMIN.

As part of this assignment, ADMIN receives the Supervisor object right to the NetWare Server object. This in turn gives ADMIN the Supervisor right to the root directory of all volumes attached to the server, so ADMIN can be used to manage all directories and files on every volume in the Directory tree.

However, user 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 at installation to provide basic network functionality:

- [Public] has the Browse object right to the [Root] object.

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

- 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:PUBLIC directory should be made part of a group with explicit rights to the SYS:PUBLIC directory.

- Users are granted the Read right to all attributes and the Write right to the login scripts attributed on their own User objects.

As other User objects are created in the Directory, you can grant 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.

After you have assigned another User object the Supervisor object right to the [Root] object, you can rename or delete ADMIN.

Never delete user ADMIN without having assigned the Supervisor right to the [Root] object or 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 an Supervisor right.

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 if there is not a user in that container with all rights, then there is a container without full administrative rights. This can cause problems. If you experience these problems, contact a Novell Authorized Reseller^{CLM} or Novell Technical Support^{SM} representative.

Directory Partitions

Directory partitions are logical divisions (or portions) of the Directory database that can be replicated. They form a distinct unit of data in the Directory tree used to store and replicate Directory information.

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

The fact that an NDS database can be separated into partitions located on servers across the network enables it to be referred to as a *distributed database*.

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

Partitions are created to increase performance and provide ease of management and scalability.

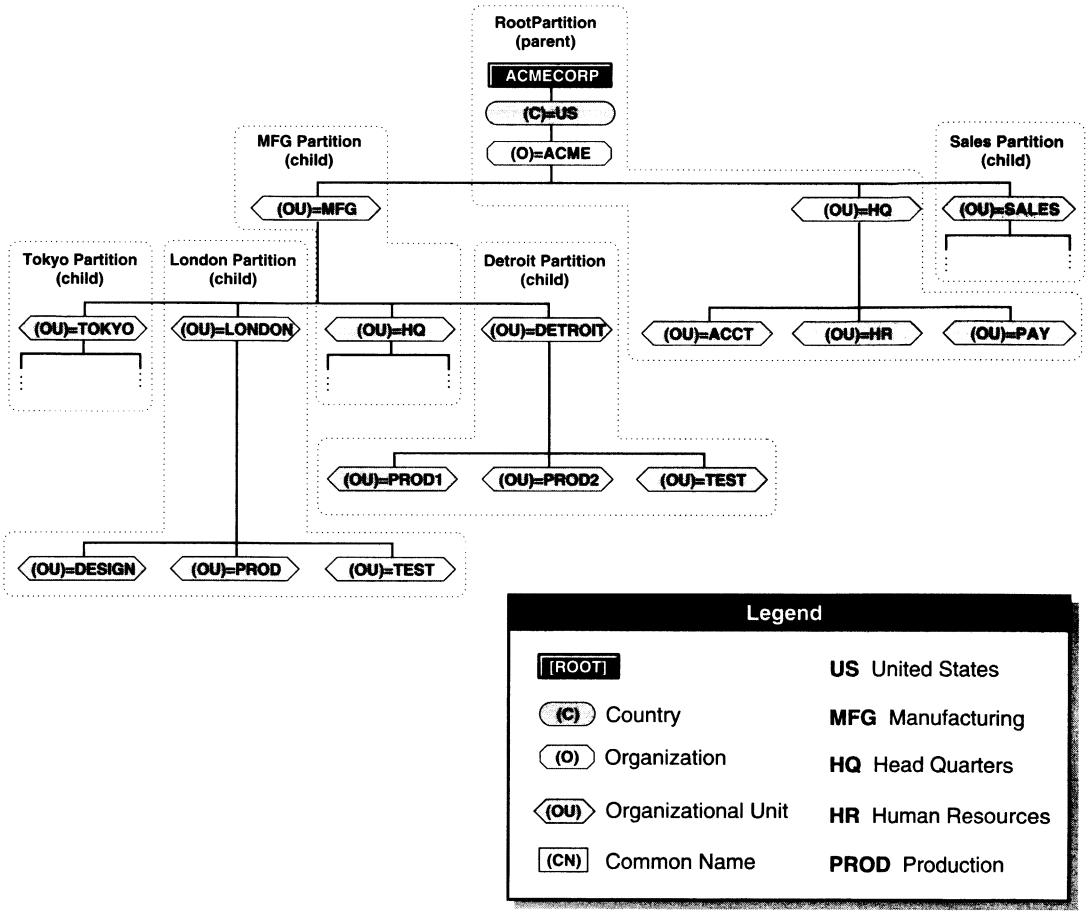
A partition is a *subtree* or *branch* of the Directory tree. Each partition is named according to the [Root]-most container object within the partition (the one that is closest to the [Root], also called the *Partition Root Entry*).

The [Root] object (also at the top of the tree) 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 2-1
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.
- Partitions are stored only on NetWare 4 servers.
- A single NetWare 4 server can contain multiple partitions.
- Partitions cannot overlap each other.

Partition Replicas

A *replica* is an instance of a partition. You can create an unlimited number of replicas for each partition and store them on any NetWare 4 servers on the network.

Purpose

Replicas are created for two reasons:

- **Directory Fault Tolerance.** If a disk crashes or a server goes down, a replica on a server in another location can still authenticate users to the network and provide information on objects in the disabled server's replica.

  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 a replica of one partition with a replica of another partition on the same server.

Important 

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 mirror or duplex your hard disks and enable the Transaction Tracking System™ (TTS) feature.

The TTS™ feature must be enabled for NDS to work. See "Transaction Tracking System" in *Concepts* and Protecting Database Integrity with TTS" in Chapter 7 of *Supervising the Network* for more information.

- **Faster Access Across a WAN Link**. If users currently use a WAN link to access particular 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 amount of synchronization required.

  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 four types of replicas you can create:

- **Master replica**. A writable replica that contains all object information for the partition. All partition operations (create, merge, move, creating a replica, deleting a replica, repair) occur from the master replica of the 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) when a master replica of the given partition is already defined somewhere else.

  There can be any number of read/write replicas.

  Client workstations can update master and read/write replicas only.

- **Read-only replica**. Contains the same object information as the partition, but the information can only be read. Used where reading of the partition is required, but writes to the partition should not occur.

A read-only replica cannot be used on a server where bindery services is required because bindery services requires a writable replica. When bindery services is set, use either a master or read/write replica.

The default setting in the NetWare 4 installation program copies a read/write replica of the partition that a bindery server is being upgraded to.

See Chapter 3, "Understanding Bindery Services," on page 41 for more information.

- **Subordinate reference replica**. Cannot be modified by any user. Automatically placed on a server by NDS if the parent partition has a master, read/write, or read-only replica on the server and the child partition does not exist on the server.

  If you add a read/write or read-only replica of the child partition to the server, the subordinate reference replica is removed.

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.

The master replica participates in the partition synchronization process by exchanging updates with other replicas, but it is not a controlling entity in this process. Similarly, each read/write replica synchronizes with the other replicas of the partition. Read-only replicas also synchronize with other replicas, but they receive updates only from other servers.

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.

For this synchronization to occur, each replica must be contacted. Each replica maintains a record of the location of each of its replicas. The locations are stored in the partition's replica property, with one property entry for each replica of the partition. 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 hierarchy and objects, as well as helping you maintain the Directory databases within your network.

See Chapter 7, "Managing NetWare Directory Services," on page 131 for more information about the use of the management utilities.

Where to Go from Here

If you want to	Go to
Use NDS in your network	Chapter 1, "Understanding NetWare Directory Services," on page 3
Use bindery services with NDS	Chapter 3, "Understanding Bindery Services," on page 41
Use time synchronization with NDS	Chapter 4, "Understanding Time Synchronization in NDS," on page 53
Plan, manage, and implement NDS	"Planning, Implementing, and Managing," on page 69
Use the management utilities	Chapter 7, "Managing NetWare Directory Services," on page 131

chapter 3

Understanding Bindery Services

Overview

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

The following topics are discussed on the indicated pages:

Topic	Page
Setting Up Bindery Services	43
Setting Bindery Context	45

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

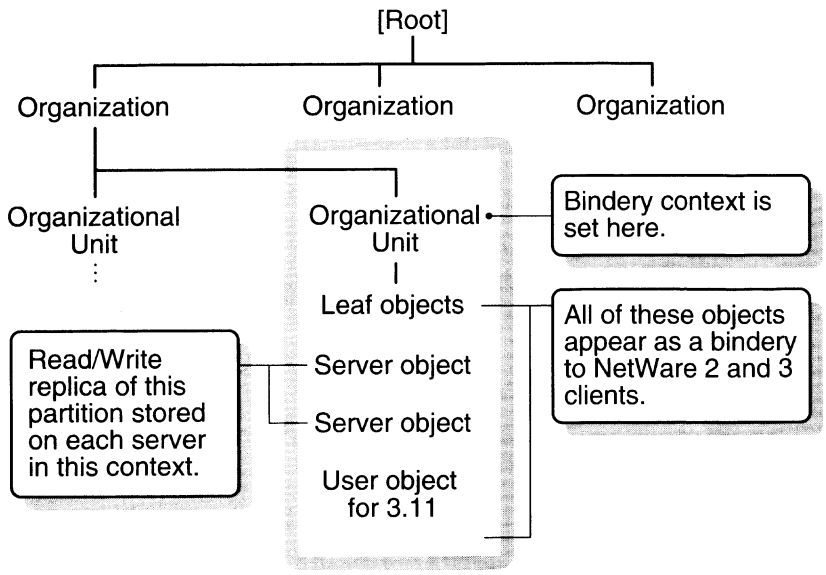
With bindery services, NDS imitates a flat structure for leaf objects within a set of Organization (O) and Organizational Unit (OU) objects. Thus, when bindery services is enabled, all objects within the specified container's bindery context can be accessed both by NDS objects and by bindery-based servers and client workstations.

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

The container object where bindery services is set is called the *bindery context*. You can change the bindery context by using the appropriate parameters with the SET or SERVMAN server utility. See "SET (NDS Parameters)" on page 147 or "SET" in *Utilities Reference*.

The following figure illustrates bindery services when a bindery context is set at an Organizational Unit object.

Figure 3-1
Bindery Services in a Directory Tree



A read/write replica of the Directory partition that provides the bindery context is stored on the server that you want bindery services enabled on. However, by default, only the first three servers installed on a partition or upgraded from a bindery to NDS receive a replica of the partition during the installation procedure and are able to subsequently support bindery services. You can still add replicas to other servers if needed for bindery services.

If a read/write or master replica is not present, you must use the Partition Manager utilities to add one to the server. See Chapter 7, "Managing NetWare Directory Services," on page 131 for information and procedures.

The installation program also sets the bindery context in the AUTOEXEC.NCF file for upgraded bindery-based servers only.

The bindery context exists, but no static bindery services exists. The SET command allows you to set up to 16 semicolon-delimited contexts. These can be both valid and invalid contexts.

Every ten minutes the validity of these contexts is verified, and static bindery services exists only in those contexts which prove to be valid. The SET command then reflects a set of bindery contexts, but some choices might be invalid and are thus ignored.

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

Bindery services is server-centric. If a client workstation does a bindery login, the login script is coming from the MAIL directory on the server that the client is logging in to. Any changes to the user's bindery login script are made on a single server and are not be distributed to any other servers.

You cannot disable bindery services if the bindery is being accessed. The bindery objects are always available unless the bindery is closed through bindery services.

Setting Up Bindery Services

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

Created Objects

- If bindery files exist for the User object GUEST and the Group object EVERYONE, the objects are upgraded. However, if these files do not exist, the objects are not automatically created.

  If you require the user GUEST or group EVERYONE, or if you use a service that requires either, you must create such a user or group.

  For example, if you are using NetWare NFS*, which requires the group EVERYONE, you must create the group and add users to the group manually.

- The User object SUPERVISOR is automatically created but is not used under NDS. (You can create a User object SUPERVISOR and assign ADMIN-equivalent rights to this user in order to use the SUPERVISOR account in NDS.)

  However, the user SUPERVISOR does exist as a bindery services User object. Although it is not visible under NDS, it can be accessed through a bindery services login. These two user objects are unique and separate objects.

- During the upgrade process from NetWare 3™ to NetWare 4™ software, all users except SUPERVISOR and all groups are upgraded to NDS objects. The user SUPERVISOR is upgraded, 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

Bindery services users cannot access NDS information that is not in the bindery. This information includes, but 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 stored on a different server, but this will not be valid until a master or read/write replica is placed on that server.

It is also possible to set the bindery context for a server to a container that is not part of a writable replica, but the context will not be accessible through bindery services.

Setting Bindery Context

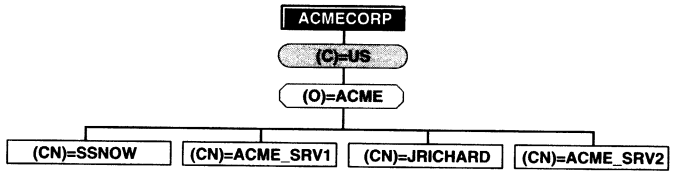
Only objects at the bindery context are available for bindery services. The bindery context should be set in the AUTOEXEC.NCF file on each server, although it can also be set at the server with the SET command.

Each server can support up to 16 containers that comprise the bindery context. Each bindery context is the name of the container where bindery services is available on the server.

In a Single-Level Directory Tree

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

Figure 3-2
Bindery Context in a Flat Directory Tree



In effect, this structure is like a bindery and is not scalable like NDS. Because there is only one container object, the bindery context for the servers can be set to O=ACME, using the following command in the AUTOEXEC.NCF file:

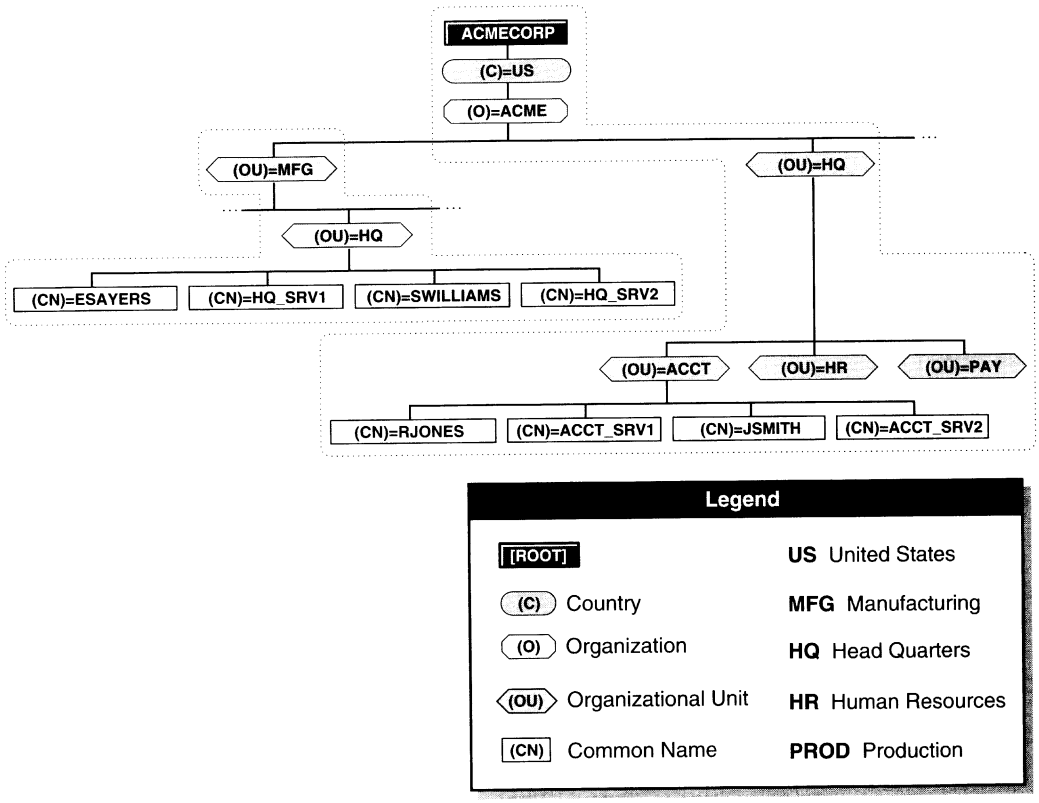
```
SET BINDERY CONTEXT=ACME.US
```

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

In a Multiple-Level Directory Tree

If the Directory tree hierarchy contains more levels, 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 3-3
Two Different Bindery Contexts in a Directory Tree



This Directory tree has five container objects, each designated by the naming type O or OU:

```
O=ACME
OU=HQ.O=ACME
OU=HR.OU=HQ.O=ACME
OU=ACCT.OU=HQ.O=ACME
OU=PAY.OU=HQ.O=ACME
```

This Directory tree also contains two servers, each with its bindery context set as follows:

```
SET BINDERY CONTEXT=HQ.ACME

SET BINDERY CONTEXT=ACCT.HQ.ACME
```

For a Specific Server

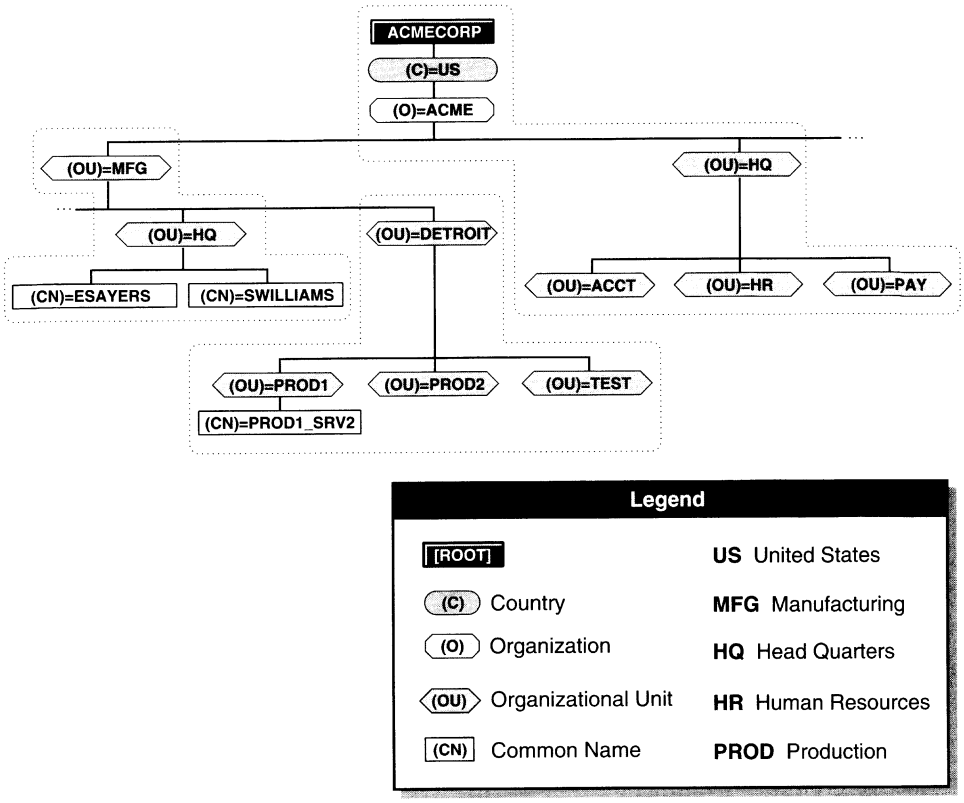
A server's bindery context can be set to any OU or O that is present in a replica on that server.

For example, given the partitions defined in Figure 3-4, you could set the bindery context of ACCT_SRV1 to any one of the following containers:

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

Figure 3-4

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 be set 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. You would enter the following command in that server's AUTOEXEC.NCF file:

```
SET BINDERY CONTEXT=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.

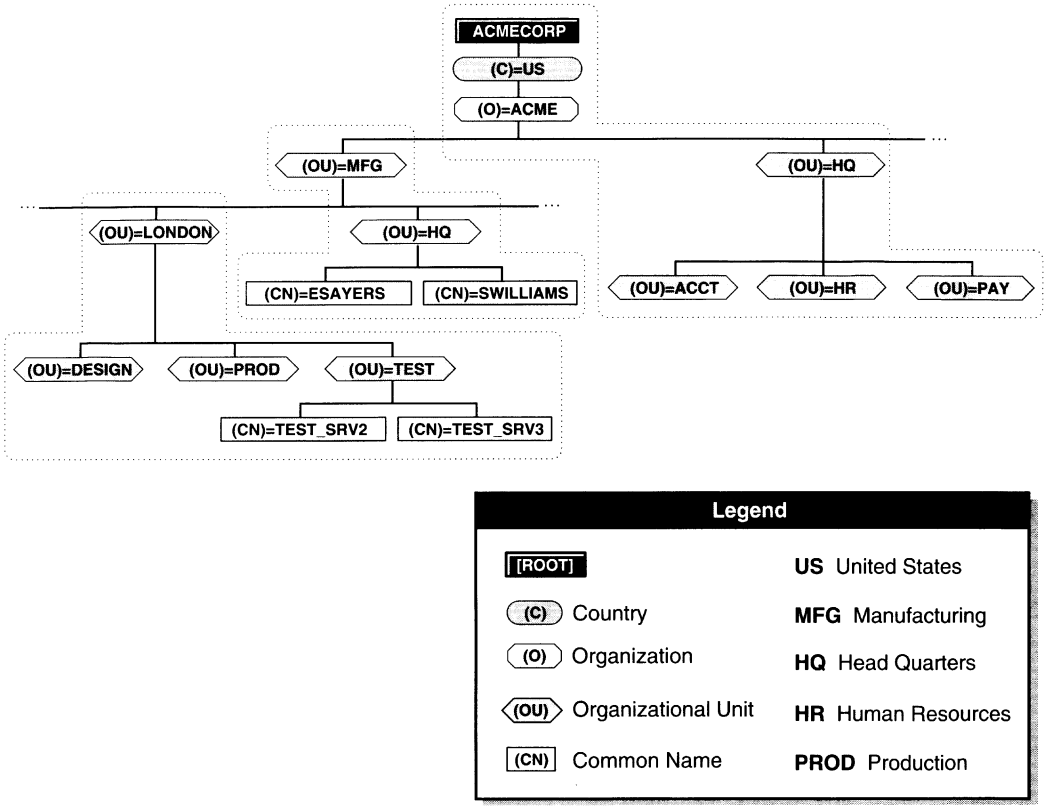
Be careful when changing a server's bindery context. Removing a container from that server's bindery context prevents all users in that container from using bindery services.

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 3-5
Multiple Servers in the Same Bindery Contexts



In this example, ESAYERS can access both servers, TEST_SRV2 and TEST_SRV3, within the OU=TEST bindery context. However, if you remove the container holding ESAYERS from the bindery context setting of either server, ESAYERS can't log in to the changed server through bindery services.

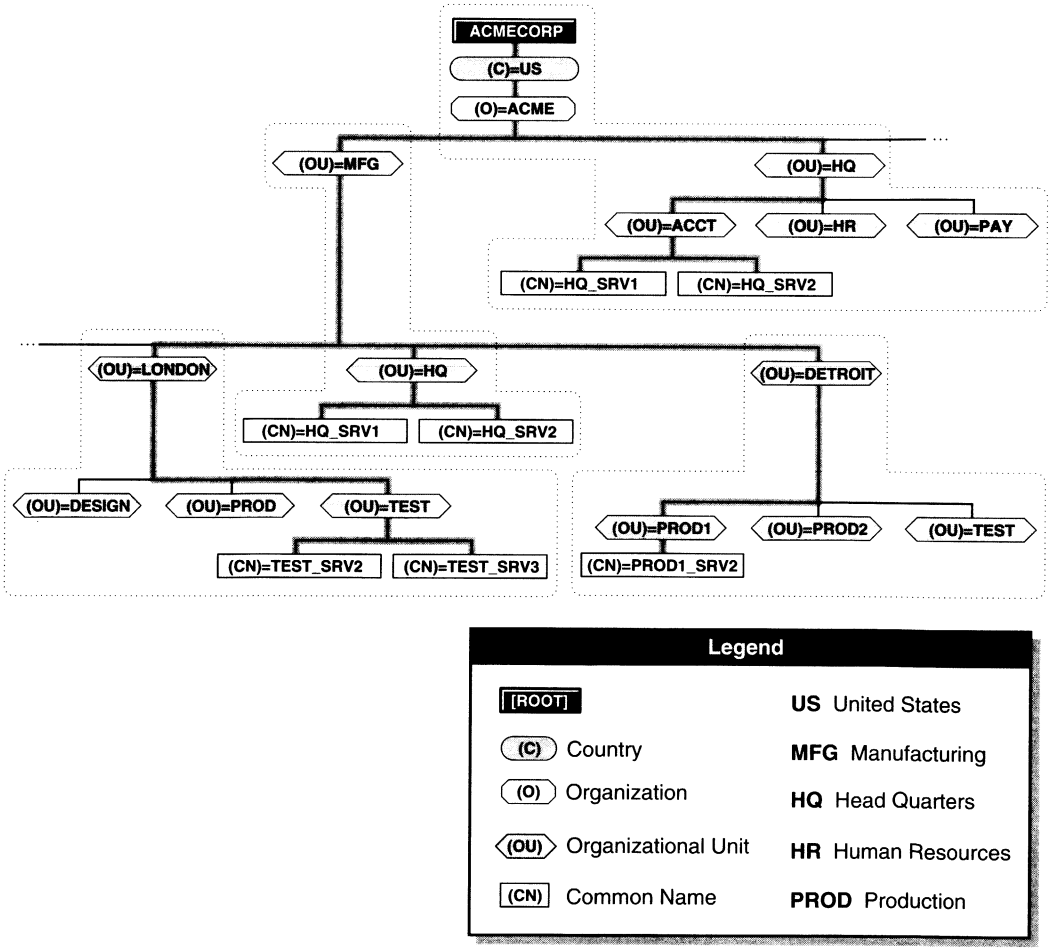
Make sure that each server you want to set in the bindery context has a writable (master or read/write) replica of the partition with the Organizational Unit (OU) you want.

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 3-6
Multiple Bindery Contexts in the Same Directory Tree



To set the bindery contexts in this figure to the container OU=HQ where ESAYES and SWILLIAMS can access the objects in the other bindery contexts, you would enter the following command in the server's AUTOEXEC.NCF file where the users are located:

```
SET BINDERY CONTEXT=ACCT.HQ.ACME;PROD1.DETROIT.MFG
  .ACME;TEST.DETROIT.MFG.ACME;SALES.ACME;
```

To set multiple bindery contexts, you must set the contexts to the [Root] of the tree. You can also set multiple bindery contexts with the SET utility. You can set up to 16 contexts per server.

Use a semicolon to separate contexts. For more information, see "SET" in *Utilities Reference*.

Important

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 in your network	Chapter 1, "Understanding NetWare Directory Services," on page 3
Use the management features included with NDS	Chapter 2, "Understanding Management Features," on page 31
Use time synchronization with NDS	Chapter 4, "Understanding Time Synchronization in NDS," on page 53
Plan, manage, and implement NDS	"Planning, Implementing, and Managing," on page 69

chapter 4

Understanding Time Synchronization in NDS

Overview

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

The following topics are discussed on the indicated pages.

Topic	Page
Time Stamps	54
Time Servers	55
Time Source Server Functions	60
Choosing a Time Synchronization Method	62

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.

Time Stamps

Whenever an event occurs in the Directory, 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.

Every NDS event is assigned a time stamp so that the order in which replicas are updated is correct by means of the time set in the time stamp. This 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.

This is important because some events, such as deleting or creating an object, need a method of recording the order of these events. If no method existed, NDS might try to create an object and then delete it.

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

When you install NetWare 4.1 on a server, you are prompted to designate it as a Single Reference, Primary, Reference, or Secondary time server.

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

Single Reference

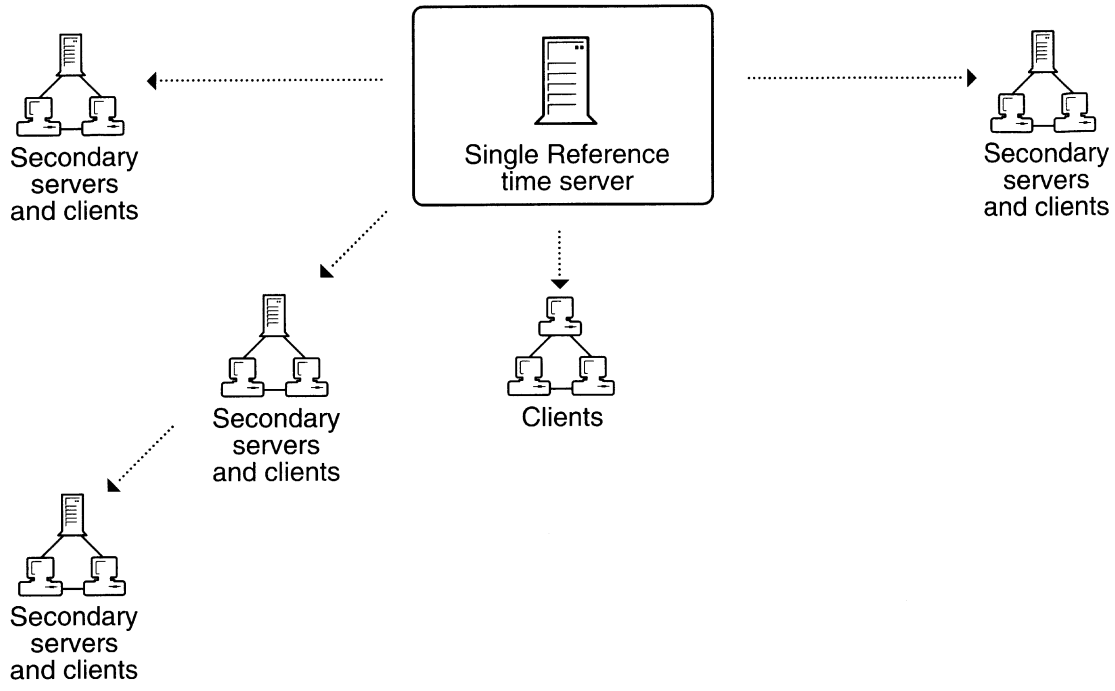
Single Reference time servers provide time to Secondary time servers and to their own client workstations.

This server determines the time for the entire network and is the only Reference server at the time it is running that provides a source of time on the 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.

Figure 4-1
Single Reference Time Server



The Single Reference time server works on networks of any size, but the TIMESYNC configuration shown inFigure 4-1 is used primarily for small networks that don't include WAN links.

Important

If you use a Single Reference time server, don't use any Primary or Reference time servers on the network because the time references will conflict.

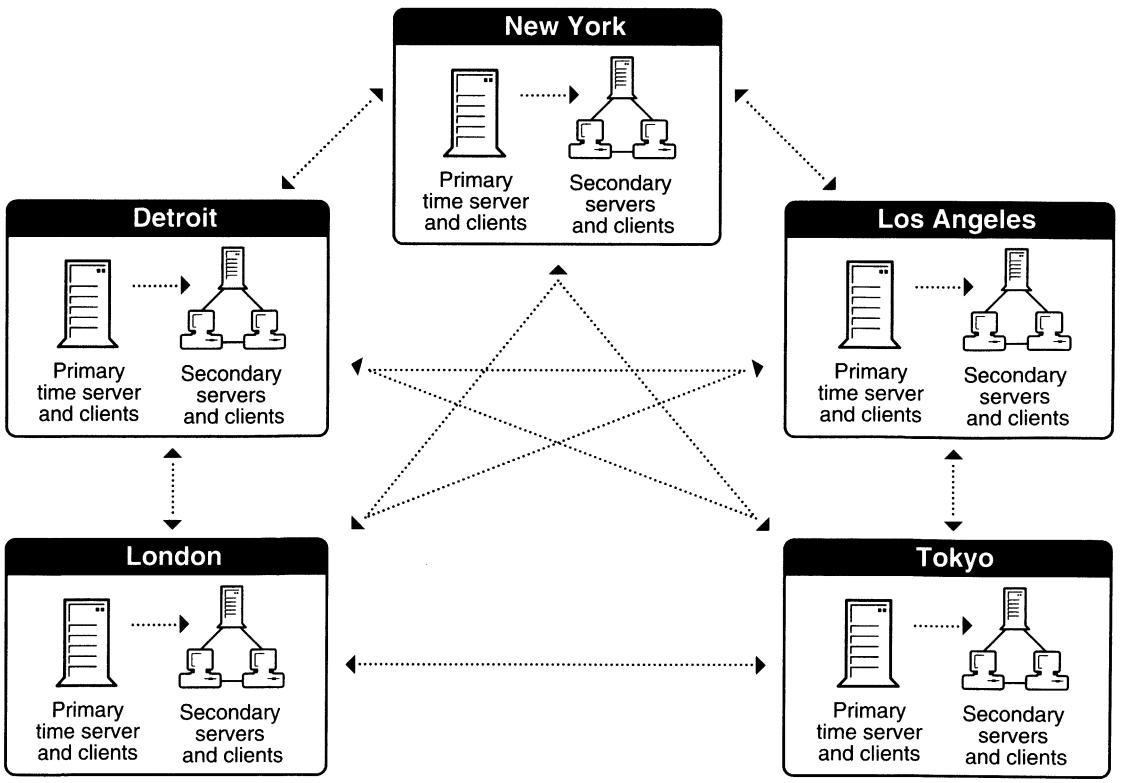
Primary

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 also "poll" with other Primary or Reference time servers to "vote" what the common network time should be. 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.

Figure 4-2
Primary Time Servers



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 a Primary time server 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.

You must have at least one other Primary time server or a Reference time server that a Primary time server can contact. Whenever Primary and Reference time servers are on a network, they must be able to contact each other for polling.

Reference

Reference time servers provide a time that all other time servers and client workstations can synchronize with.

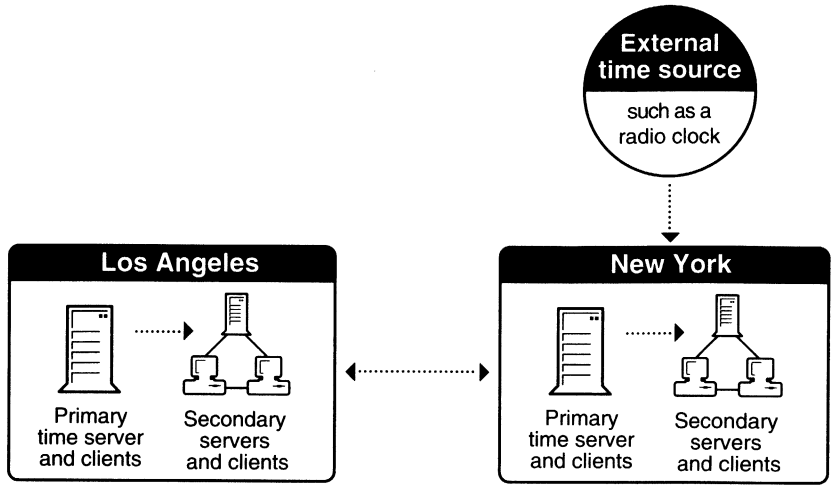
Reference time servers can be synchronized with an external time source, such as a radio atomic clock.

A Reference time server acts as a central point to set network time by. 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, the Primary 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 its own Secondary servers and client workstations, as well as to a Primary time server at another location.

Figure 4-3
Reference Time Server



Use a Reference time server when it is important to have a central point to control 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 poll the servers to vote on a correct time.

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

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 (polling and voting) the correct network time.

If you have designated a server on the network as a Single Reference time server, then designate all other servers on the network as Secondary time servers.

If you have designated several servers on the network as Primary or Reference time servers, then designate all other servers on the network as Secondary time servers.

To keep network traffic to a minimum, use Secondary time servers that are 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.

Time Source Server Functions

The Single Reference, Primary, and Reference time servers are all *time source server*s. 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.

The SAP method, however, generates additional network traffic.

The SAP method can 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, or Primary time server).

Custom Configuration

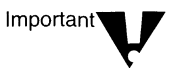
Custom configuration of your time servers gives you more control over time synchronization, but it requires more planning to synchronize servers efficiently.

An advantage of custom configuration is that you maintain complete control of the time synchronization environment.

Also, custom configuration helps eliminate nonessential network SAP traffic, as well as errors associated with accidental reconfiguration.

It is also possible to list the specific time source servers that a server should contact.

It is possible to specify that a server should not listen for SAP information from other time source servers and that it is not to advertise its presence using SAP.



The custom configuration does require additional time for planning and installation.

It is also 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 each 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.

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. (These are the installation defaults.)

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

The following tables and figures discuss and illustrate the major points of each type of time server.

Table 4-1
Single Reference Time Server

Function	Provides time to Secondary time servers and client workstations.
Use	Works for all networks, regardless of size, but is used primarily for a small LAN. Installed by default with NetWare 4 installation.
Cautions	Because the Single Reference time server is the sole source of time on the network, all other servers must be able to contact it. No Primary or Reference time servers are allowed on the network.

Figure 4-4
Example of a Network Using a Single Reference Time Server

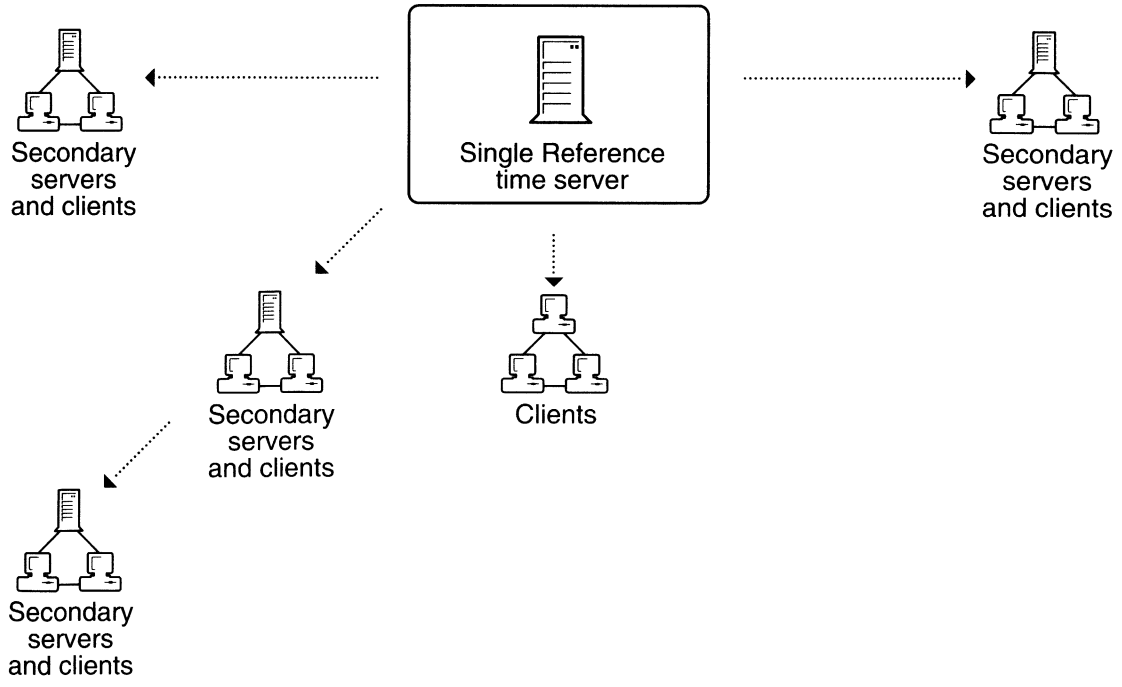


Table 4-2
Primary Time Server

Function	Synchronizes the time with at least one other Primary time server or with a Reference time server, and provides the time to Secondary time servers and workstations. Polls other Primary or Reference time servers and then votes with them to synchronize the time.
Use	Use on larger networks to increase 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. Use a Primary time server for every 125-150 Secondary servers. Place one in each geographically distant area so that Secondary time servers and client workstations can access them without using WAN links.
Cautions	You must have at least one other Primary time server or with a Reference time server that a Primary time server can contact.

Figure 4-5
Example of a Network Using Primary Time Servers

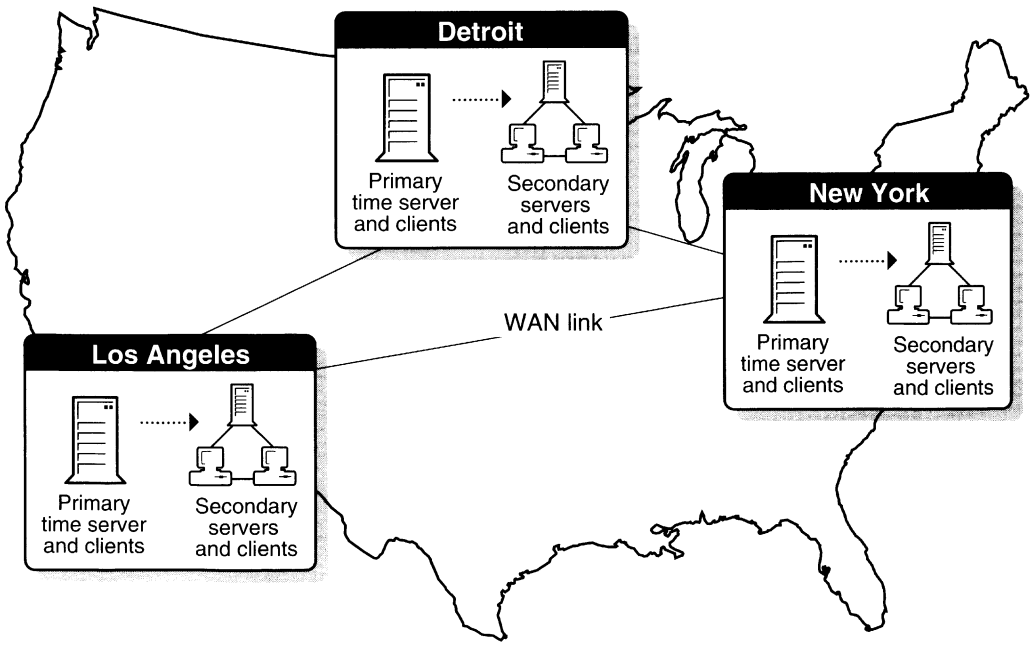


Table 4-3
Reference Time Server

Function	Provides a time that all other time servers and client workstations can synchronize with. Polls other Primary or Reference time servers and then votes with them to synchronize the time. Because a Reference time server does not change its clock, Primary time servers must reach consensus with the time that a Reference time server provides.
Use	Use when it is important to have a central point to control time on the network.
Cautions	Usually, only one Reference time server is installed on a network; however, because the Reference time server must poll at least one other time server, you must install a Primary time server or another Reference time server on the same network. If there is more than Reference time server, each must be synchronized with the same external time source.

Figure 4-6
Example of a Network Using a Reference Time Server

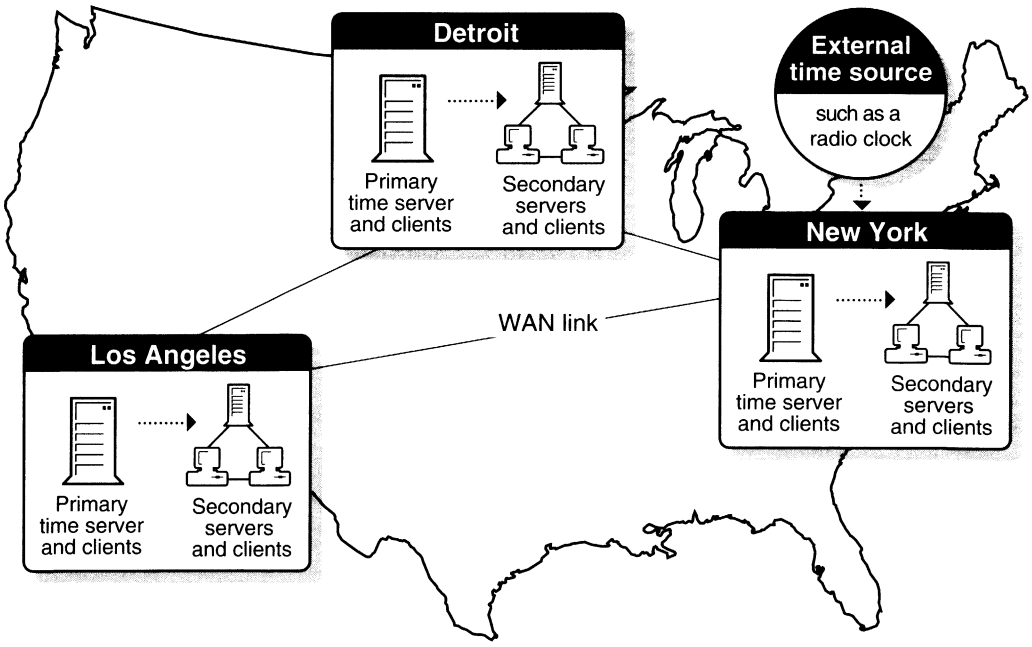


Table 4-4
Secondary Time Server

Function	Obtains the time from a Single Reference, Primary, or Reference time server and provides the time to client workstations.
Use	If you have designated a server on the network as a Single Reference time server, then designate all other servers on the network as Secondary time servers. If you have designated several servers on the network as Primary or Reference time servers, then designate all other servers on the network as Secondary time servers. Associate Secondary time servers to Primary or Reference time servers that are physically close in order to keep network traffic to a minimum. For the best time synchronization, require each Secondary time server to contact a Single Reference, Primary, or Reference time server with as few intervening routers and slow WAN segments as possible.
Cautions	You can have a Secondary time server contact another Secondary time server for the time. However, you should be cautious when removing a Secondary time server if other Secondary time servers are contacting it for time. This might cause the other Secondary servers to be too many hops away from a time source server to synchronize with.

Where to Go from Here

If you want to	Go to
Use NDS on your network	Chapter 1, "Understanding NetWare Directory Services," on page 3
Use the management features included with NDS	Chapter 2, "Understanding Management Features," on page 31
Use bindery services with NDS	Chapter 3, "Understanding Bindery Services," on page 41
Plan, manage, and implement NDS	"Planning, Implementing, and Managing," on page 69

Planning, Implementing, and Managing

Overview

The NetWare Directory Services™ (NDS) technology requires you to set up a Directory tree for implementing NDS™ on your network. Efficient planning and management can make your implementation simple and easy to do.

This section will provide you with a conceptual overview only. Additional sources of information for planning your Directory tree are Novell Education classes and Novell *Application Notes*™. For more information on classes at a Novell Authorized Education Center™ facility, in the U. S. and Canada call 1-800-233-3382 or 1-801-429-5508.

Contents

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

Purpose	Chapter	Page
To learn more about planning a NetWare Directory tree	Chapter 5, "Planning NetWare Directory Services Implementation"	71
To learn more about implementing NetWare Directory Services on your network.	Chapter 6, "Implementing NetWare Directory Services"	107
To learn more about managing a NetWare Directory tree and database	Chapter 7, "Managing NetWare Directory Services"	131

chapter **5**

Planning NetWare Directory Services Implementation

Overview

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

The following topics are discussed on the indicated pages:

Topic	Page
Guidelines for Implementing NDS	73
Creating Naming Standards	75
Developing an Implementation Strategy	79
Organizing Objects into a Logical Hierarchy	83
Developing a Replication Strategy	95
Developing a Time Synchronization Strategy	99
Developing a Security Strategy for the Directory Tree	100
Developing an Integration Strategy for Bindery Services	103

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, a good implementation of 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 the implementation of NetWare Directory Services, 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 very flexible and has been designed to allow restructuring as you change the structure of organizations, change locations, or merge Directory trees.

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 the 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 relative 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, "Creating a Standards Document for NDS Object Classes and Properties," on page 175 for more information.

   See also "Creating Container Objects," "Creating Leaf Objects," and "Searching for Objects" in Chapter 1 of *Supervising the NetWork*.

2. Choose a method of implementation.

   There are several ways of implementing a NDS:

   - Set up small trees for different groups or departments in your organization and then merge those trees into one larger tree.

     See "Planning a Departmental Directory Tree (Merge Tree Method)" on page 79.

   - Plan your directory tree from the top, or root level, down to the branches.

     See "Planning an Organizational Directory Tree" on page 82.

3. Organize objects into a logical hierarchy.

   The hierarchy of your Directory tree should be as shallow as possible (five to eight 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" on page 76 and "Placing Leaf Objects in the Directory Tree" on page 90.

5. Develop strategies for adequate replication of the partitions to

   - Provide fault tolerance
   - Decrease traffic over WAN links

   Divide the Directory database into partitions based on logical boundaries, and then replicate the partitions where you want users to access them without having to go over WAN links.

   See "Developing a Replication Strategy" on page 95.

6. Choose a method for providing time synchronization for the Directory. You must designate which time servers you want to use as a time source servers (Single Reference, Reference, Secondary, or Primary).

   See "Developing a Time Synchronization Strategy" on page 99.

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" on page 100.

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

   If you are upgrading from NetWare 2 or NetWare 3™ software, you should prepare binderies.

   See Chapter 3, "Understanding Bindery Services," on page 41; and also see "Bindery Services," in Chapter 2 of *Upgrade*.

Creating Naming Standards

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 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™ software, 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, "Creating a Standards Document for NDS Object Classes and Properties," on page 175.

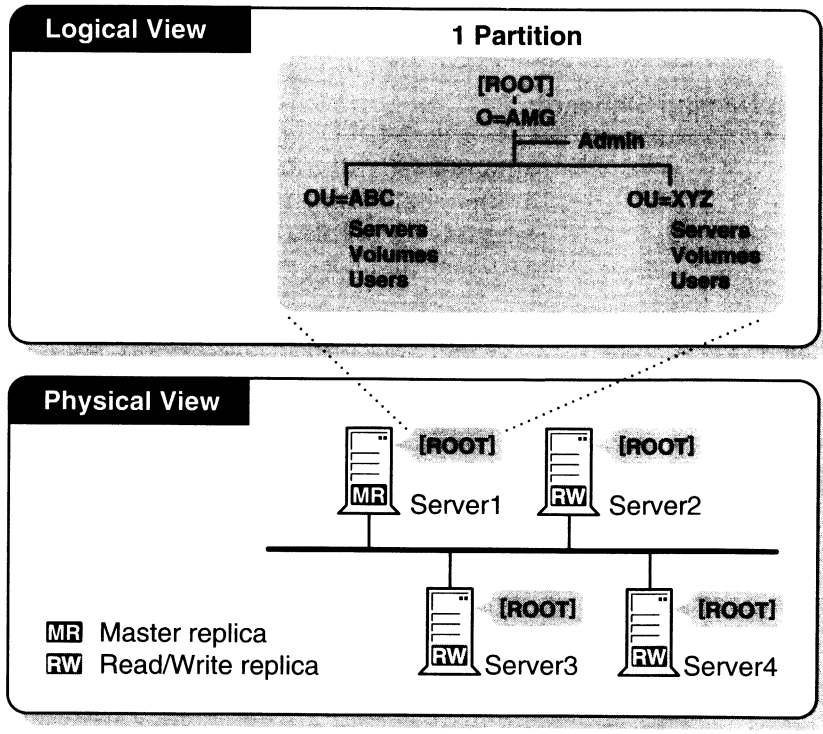
Creating Directory Tree Maps

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

The second is a physical view 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 5-1
Directory Tree View Maps



Planning 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 the Directory, then the 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 the object names follow a standard, then searching is simpler.

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

See Appendix C, "Creating a Standards Document for NDS Object Classes and Properties," on page 175 for more information.

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

Using Naming Standards

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 be using bindery services, make sure the names are compatible with standards for bindery-based versions of NetWare. (See Appendix C, "Creating a Standards Document for NDS Object Classes and Properties," on page 175.)

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 makes it easy for users to identify the resources available to them to search for specific items in the Directory tree.

Although a consistent naming standard for the corporate network is important, you do not need to have it perfected before you implement NDS. You can rename containers and leaf objects and move subtrees to reflect any changes you want to implement. See Chapter 7, "Managing NetWare Directory Services," on page 131 for more information.

Name Length

Make sure the naming schemes are short, yet as descriptive as possible. For example, "SW 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 (OU) 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.

Developing an Implementation Strategy

There are several ways of implementing NDS:

- Planning a Departmental Directory Tree (Merge Tree Method)
- Planning an Organizational Directory Tree
- Using a combination of the first two planning methods

Planning a Departmental Directory Tree (Merge Tree Method)

The defaults in the server installation program help you set up a small Directory tree with very little planning. By choosing the installation defaults, you are provided with some leaf objects (such as the ADMIN object), Directory partitions, and network time synchronization.

You can set up small trees for different groups or departments in your organization and then merge those trees into one larger tree.

Time services can be synchronized acrosscontainer objects.

Once the Directory tree has been implemented in a departmental network, that network can easily be merged into a larger organizational Directory tree.

The advantages of using this method of setting up your Directory tree include

- Less planning required
- Easier modification of the Directory tree

Tree Restructuring

Although some planning is important to implement NDS successfully, your first plan is not your only chance to define the tree structure. NDS utilities allow for tree restructuring as organizations change structure, change location, or merge. See Chapter 7, "Managing NetWare Directory Services," on page 131 or the individual entries in *Utilities Reference* for more information.

Merging Directory Trees

When you merge Directory trees using the DSMERGE utility, the objects in the source tree become part of the target tree. The target [Root] object becomes the new root for objects that are moved from the source tree.

The DSMERGE utility does not change Directory names or contexts within the containers. The Distinguished Names of objects in the new tree are the same as the names before the trees were merged.

This utility does not allow you to merge container or leaf objects.

Use the following guidelines for merging Directories trees:

- Before performing a merge operation, ensure that the state of synchronization for all servers affected by the operation is stable. The following table provides recommendations for preparing source and target trees for merging.

Condition	Action
No leaf object or aliases can exist at the root of the source tree.	Delete any aliases or leaf objects at the root of the source tree.
No similar names can exist on source and target trees.	Rename objects on the source and target tree if identical names exist.
No login connections can exist on either the source or target tree.	Close all connections on the source and target trees.
The version of NDS must be the same on both source and target trees.	Upgrade all non-4.1 servers that have a replica or the [Root] object.

Condition	Action
Any server that contains a replica of the [Root] object on both source and target trees must be up and running.	Ensure that all servers containing a replica of the root partition on both source and target trees are up and running. Ensure that any WAN links affected are stable.
Schema on both source and target trees must be the same.	Ensure that source and target trees are time synchronized within two seconds of each other.

- In general, it is easier and faster to merge the smaller of the trees into the larger.

You cannot share resources across separate Directory trees. Each tree has its own database of objects that is not visible from another tree. Be aware of this limitation if you are planning to create multiple trees. You can, however, access other trees through bindery services, or you can merge two trees in order to share data.

For more information about	Refer to
Merging trees	"DSMERGE" in *Utilities Reference* and "Merging NDS Trees" in Chapter 5 of *Supervising the Network*
Moving subtrees	"NWADMIN," in *Utilities Reference*
Moving objects	"Managing NetWare Directory Services Objects" in Chapter 1 of *Supervising the Network*
Renaming objects	"Managing NetWare Directory Services Objects" in Chapter 1 of *Supervising the Network*

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 administrative utilities to create the next Organizational Unit (OU) levels in the Directory tree. Then create a User object in each OU with all rights in that container.

- Following the creation of the User objects, use a partition utility (PARTMGR or NetWare Administrator) to change each OU into a partition root (if the tree will be large enough for separate partitions at these OU levels).

  This allows the root network supervisor to give those User objects and passwords to other supervisors. Then those 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 features in NetWare 4 security to control the access to any part of the Directory tree.

Organizing Objects into a Logical Hierarchy

Keeping your Directory tree structure as shallow as possible (five to eight 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 organization chart, and then 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 organization 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 the 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 objects, other container objects, and leaf objects.

Consider the following when planning the 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.
- [Root] is named by the *true name* of the Directory tree, which is the name you use to name the Directory tree; therefore, it is part of the Directory tree and part of every object's context.
- 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.

Placing Container Objects in the Directory Tree

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

Country and Organization Objects

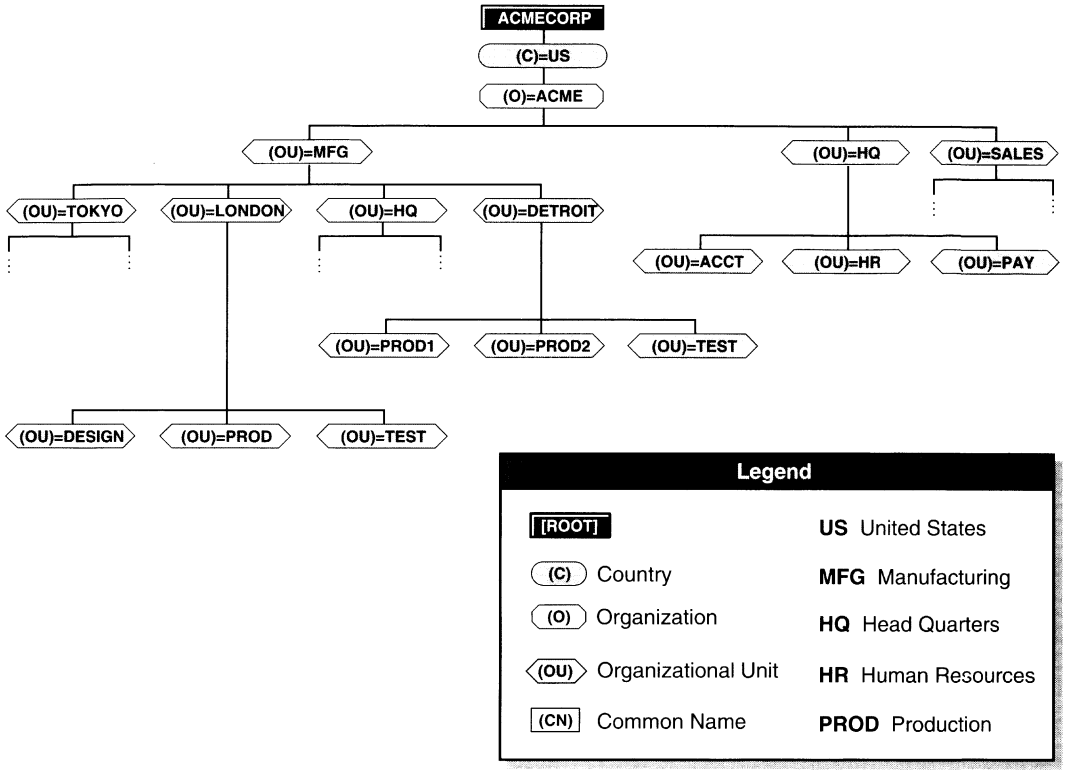
You must include one or more Organization (O) objects in your Directory tree. Organizational Unit (OU) objects are optional. However, you should develop most of your Directory tree structure with these objects.

Use of the Country (C) object is useful only in specific cases when the Country properties apply to your organizational structure or you are planning on participating in the information superhighway. See Appendix A, "NDS Object Classes and Properties," on page 155 for more information.

You can create as many sibling Organization (O) objects as you need, and as many Organizational Unit (OU) objects underneath the Organization (O) objects as you need, to best structure your Directory tree.

The following figure shows an example tree with one Organization (O) object and multiple Organizational Unit (OU) objects.

Figure 5-2
Directory Tree with an Organization Object and Multiple Organizational Unit Objects



Another way to designate different site locations is to create the container object Country, which is placed between the [Root] object and the Organization object. Adding a Country object (C) to your tree, however, adds one more level of complexity to an object's context that might not be necessary.

Normally, you need to create a Country object (C) when you have a global network that spans multiple countries or you plan to participate in the information superhighway.

Much of the current development for accessing services available on the information superhighway is being done according to the X.500 specification. NetWare Directory Services (NDS) is consistent with this emerging international standard.

The X.500 specification was developed by the IEEE (Institute of Electrical and Electronic Engineers) 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.

The Country object is not part of the NetWare 4 default server installation; that is, you are not prompted for a Country object when you install NetWare 4 software.

Nevertheless, you can create a Country object during the server installation. See "Install Server Software," in Chapter 2 of *Installation* for more information on installing a server.

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.

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

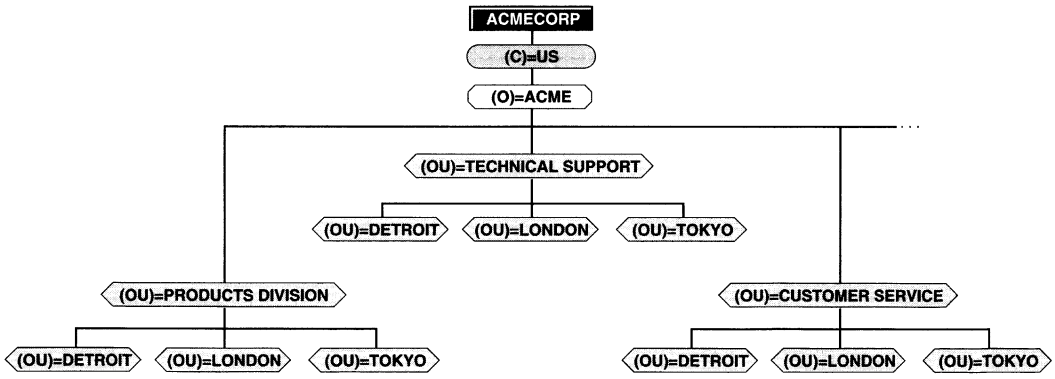
Because one goal of having a Directory tree is to provide a static database that is updated infrequently, broad geographic designations in which objects remain static, such as states or cities, might provide a more stable structure for your tree than one that is continually changing.

The following figure shows an example Directory tree in which the

- Organization (O) object is designated as the company ACME Corporation
- Organizational Unit (OU) objects 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 requires data from each management OU in each site which makes it easy to administer. However, this tree may not facilitate placement of replicas.

Figure 5-3
Directory Tree with Organizational Unit Designations at Different Levels

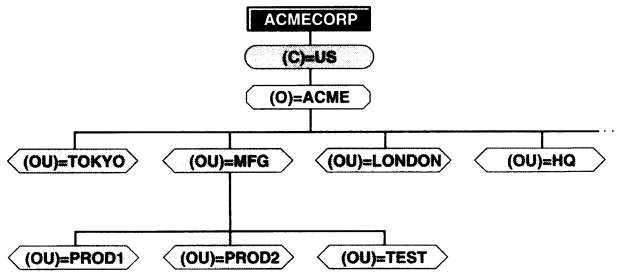


Organizational Unit (OU) objects do not all have to be the same type. That is, you can designate a workgroup as an Organizational Unit and also designate a project as another Organizational Unit.

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 5-4
Directory Tree with Mixed Organizational Unit Object Types



Some areas of your tree might need more than one Organizational Unit (OU). In the current example, the Organizational Unit (OU) 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 department, etc.

Therefore, another level of Organizational Units resides under the MFG Organizational Unit (OU) 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 5-3 on page 88.

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 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 how to use user templates in specific Organization (O) objects and Organizational Unit (OU) objects.

  For more information, see "Managing User Templates" in Chapter 1 of *Supervising the Network*.

- 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 "Container Rights" on page 101.

  For more information on inheriting rights, see "Security" in *Concepts*. For more information on Group objects, see "Managing Group Objects" in Chapter 1 of *Supervising the Network*.

Directory Tree Planning Examples

The following examples represent some of the planning conventions used for implementing NDS in small-to-medium- and medium-to-large-sized organizations.

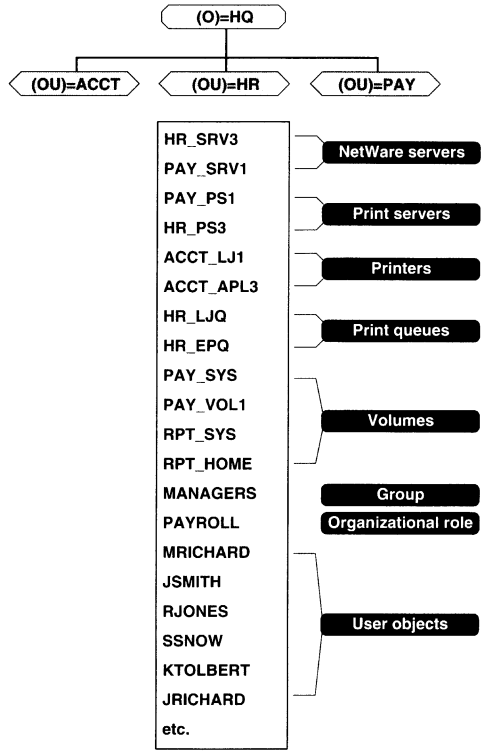
Small-to-Medium-Sized Tree

As an example of a small-to-medium-sized Directory tree, assume that the ACME Corporation headquarters has the following four offices within the city of New York:

- Sales
- Accounting
- Payroll
- Human Resources and Personnel

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

Figure 5-5
Physical Layout of a Small-to-Medium Directory Tree



Notice in this example that all the usernames start with the initial of the first name, followed by the last name. Also notice that the printers in ACCT.HQ end with "LJ" for LaserJet* printers and "APL" for Apple* printers.

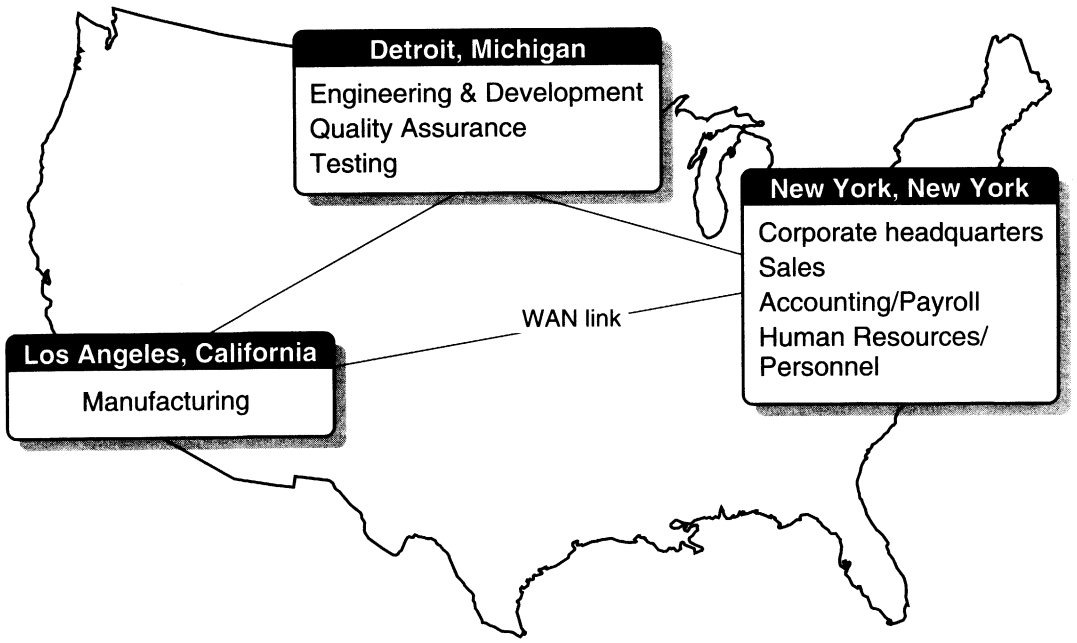
Medium-to-Large-Sized Tree

As an example of a medium-to-large-sized Directory tree, 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

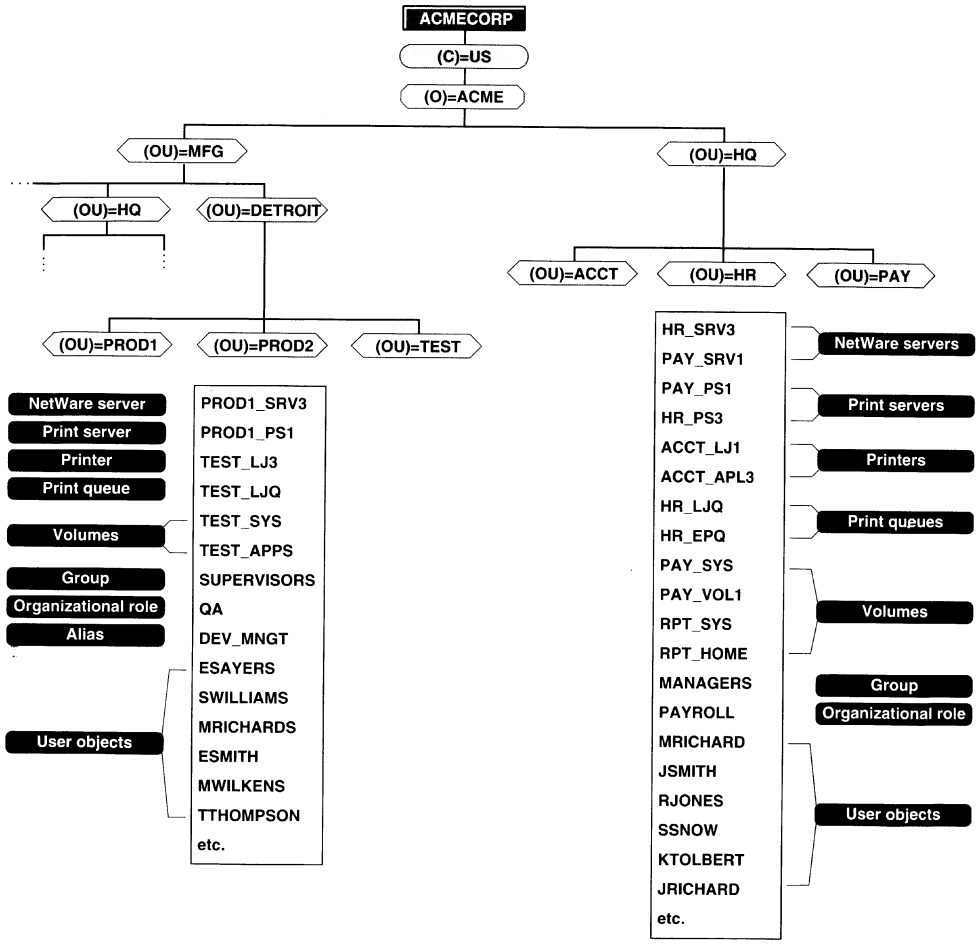
The following figure shows the physical layout of the offices, facilities, and sites used in the example, illustrating some of the previously discussed planning guidelines.

Figure 5-6
Physical Layout of a Medium-to-Large Directory Tree



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

Figure 5-7
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 printers in ACCT.HQ ACME.US end with "LJ" for LaserJet printers and "APL" for Apple printers. The printers inTEST.DETROIT.MFG.ACME.US are using a similar convention.

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

See Appendix C, "Creating a Standards Document for NDS Object Classes and Properties," on page 175 for ideas on how to standardize the naming of objects and properties in your Directory tree.

Developing a Replication Strategy

Replicas serve two purposes:

- Provide fault tolerance
- Decrease WAN link traffic at login and authentication

Providing Fault Tolerance

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

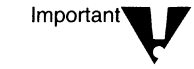
- It allows users in that area to access your partition.
- It protects the existence of your partition if a disaster destroys your own servers and replicas.

You should have at least three replicas of every Directory partition. If you lose a partition and you do not have a replica of that partition, you could permanently lose access to a part of your Directory tree.

A tape backup can be used for disaster recovery (when all copies of a partition are gone). Ensure that your backup software uses object name and not object ID to back up the Directory.

Restoring from a tape backup replaces data for objects being restored, but it deletes any changes made to objects after the tape backup was performed.

The restored objects are synchronized with the same objects on other replicas if they exist.



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

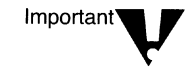
To provide fault tolerance for your files, you must mirror or duplex your hard disks and enable the Transaction Tracking System™ (TTS) feature.

The TTS™ feature must be enabled for NDS to work. For more information, see "Protecting Database Integrity with TTS" in Chapter 7 of *Supervising the Network*.

Decreasing WAN Link Traffic

If any users must access the Directory tree through a WAN, you can place a read-only replica of the necessary partition on a local server so you 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 the other 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 replica of a server 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 that information needs to cross the link is when the replicas are being updated.

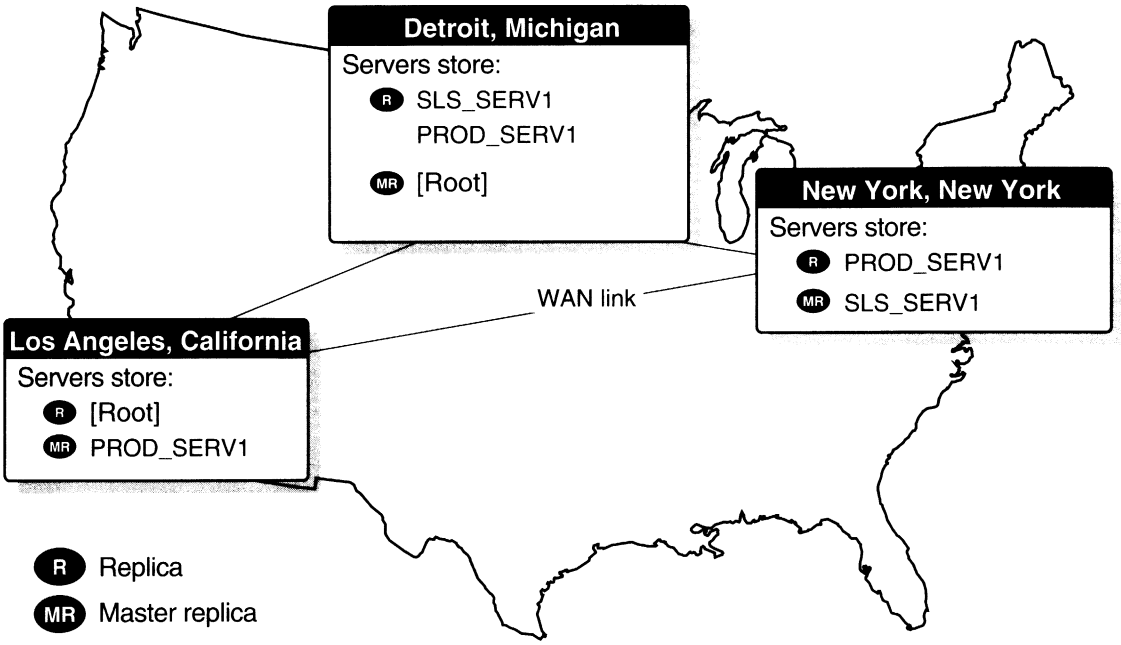
However, remember that every server that carries a replica must receive all changes to any object within that partition. The more replicas of a given partition you have, the more time needed and the more WAN traffic that exists to fully synchronize the replicas.

Before you begin distributing the replicas of the partitions, 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 then replicate only the necessary portion.

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

Figure 5-8
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 SLS_SERV1, a server at the Los Angeles site stores the master replica of PROD_SERV1, etc.
- Servers in the Detroit location store replicas of SLS_SERV1 and PROD_SERV1 so that information is locally accessible to users in the Engineering and Quality Assurance departments.
- A server in the DESIGN office in Los Angeles stores a replica of [Root] 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 PROD1.DETROIT.MFG.ACME US to allow local access.
- Resource objects in the Detroit office seldom access objects in other parts of the Directory tree within the WAN. (This is because this office operates as a self-contained business unit within the organization.)

This is only *one* way 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 Chapter 5 of *Supervising the Network*
PARTMGR text utility	"PARTMGR" in *Utilities Reference*
Partition Manager in NetWare Administrator	"Managing the NetWare Directory Tree" in Chapter 5 of *Supervising the Network*

Developing a Time Synchronization Strategy

With time synchronization in NetWare 4, you can choose the default server installation or a custom configuration. See Chapter 4, "Understanding Time Synchronization in NDS," on page 53 for more information.

Before you install NetWare 4, 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 the time servers be located on the network so that fault tolerance is provided and network traffic is kept to a minimum?

To enable time synchronization with INSTALL, you need to specify the type of time server the server you are installing on will be, the time zone you're in, and daylight savings time rules.

When you install NetWare 4 on a server, you can choose the default time server, Single Reference (or Secondary, if this installation is not the first server you've installed in the Directory), to establish the type of time services that server provides.

Or, you can designate the server as a Reference, Primary, or Secondary time server. Each designation performs a particular time synchronization function.

If you do not want to use the installation default, you must 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 plan to all the local network supervisors who will install NetWare 4 on servers within 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" on page 62
Service Advertising Protocol	"SAP (Service Advertising Protocol)" on page 61
Time servers	"Time Servers" on page 55
Time synchronization	"Choosing a Time Synchronization Method" on page 62
	"Managing Network Time Synchronization" in Chapter 7 of *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 any object or object's properties for any other object.

Container Rights

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

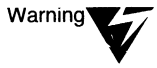
By granting rights at the container, those rights are automatically available for every object in that container unless masked by an Inherited Rights Filter (IRF). See "Inherited Rights Filter" on page 101.

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 (IRF) is a list of rights that can be created for any object. It controls the rights that a trustee can inherit from parent container objects.



Never delete user ADMIN without having assigned the Supervisor right to the [Root] object to another User object. Neglecting to do so can be disastrous, because you eliminate Supervisor 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 an explicit Supervisor right.

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 masked at a container and if there is not a user in that container with all rights, then there is a container without full supervisory rights. If you experience this type of problem, contact a Novell Authorized Reseller[CLM] or Novell Technical Support[SM] representative.

Security Equivalency

Use the Security Equal To property to give a user *temporary* 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.

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

Users who manage other users should be granted the Write right to this property. This allows user account managers to make users security equivalent to other users that they manage.

Every object is security equivalent to all container objects that are part of its Distinguished Name. Because of this, you can make a container a trustee.

Every object in a container object has the rights that are granted to that container, through the Security Equal To property. However, none of these container objects is listed in a user'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.

To add an object to a user's "Security Equal To" list, you must have at least the Write property right to the ACL property of the object you want to add to the list. However, you need the Browse object right to the Security Equal To property of the user to add a user to the list.

In networks containing confidential data that only selected users should have access to, 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" in *Concepts*
Security and security examples	"Security" in *Concepts*

Developing an Integration Strategy for Bindery Services

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

Bindery-based users can access any object in the Directory 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 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.

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.

Managing Bindery Services

Once NDS is installed, user 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 administrative tools, such as NETADMIN and NetWare Administrator, for managing bindery services.

See Chapter 7, "Maintaining the NetWare Server," in *Supervising the Network* for more information.

The NetWare shell software (NETX) does not support the NDS administrative utilities.

Changing Bindery Context

The bindery context is automatically set to the context that you upgrade the server to.

To change the default bindery context to, for example, O=Novell, you would use the following command at the server console or in the AUTOEXEC.NCF file on the server users are logging in to:

```
set bindery context=novell
```

You must use this SET command on each server that will use this bindery context. See "Modify the AUTOEXEC.NCF File" in Chapter 3 of *Installation*.

Changing Directory Tree Structure

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

If you need to change the Directory, make sure that bindery services users will still have access to the necessary services.

Moving Bindery Objects

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

Where to Go from Here

If you want to	Go to
Use the management utilities for NDS	Chapter 7, "Managing NetWare Directory Services," on page 131
Implement NDS on your network	Chapter 6, "Implementing NetWare Directory Services," on page 107
Install NetWare 4 software	Chapter 2, "Installing a New NetWare 4 Server," in *Installation*

chapter 6

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.

Topic	Page
Completing General Tasks and Guidelines for All Networks	108
Implementing NDS on Various Sizes of Networks	114

Introduction

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

You can also create an enterprise-wide system, using NDS, that spans multiple sites and countries, maintaining multiple partitions and partition replicas within a multilevel hierarchy of containers and objects.

In either case, the NetWare 4™ software and NDS provide

- Simple administration and advanced security features
- Reliable fault tolerance

- Flexible and usable implementation in both heterogeneous and mixed environments
- Interoperability with other directory services being developed for the information superhighway

The following discussion outlines the recommended tasks to be performed for implementing NDS features and functionality for 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, "Creating a Standards Document for NDS Object Classes and Properties," on page 175 for help in creating this document.

2. Sort the 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 the objects into a logical hierarchy.

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

   Organization chart structure. Base the Directory tree on the structure of your organization. When planning the Directory, you can begin with the organization 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 the organization chart 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 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.

   When you install NetWare 4, the INSTALL utility automatically installs NetWare Directory Services and prompts you to name the [Root] with the tree name. Then you can create and name an Organization (O) object and up to three Organizational Unit (OU) objects.

   You must then set the server context within the Directory tree. If you want to participate in the information superhighway, add a country code when setting the server context and a Country object will be created directly below the tree name or [Root] object.

   The 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, you might need to increase the amount of memory on that file server to improve performance.

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

   The NetWare Administrator utility is an MS 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 MS 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 centralized NetWare 4 file server can be placed in the O container for management purposes. Application servers and print queues can be placed in OU containers with the users or groups that utilize them.

   Profile objects. Create Profile objects that provide organization or department login scripts in the respective container 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 the authority over the objects within that container.

6. Add new servers to appropriate contexts.

   To add a new server, first create the container you want to install a new server into by using the NetWare Administrator or NETADMIN utility. Then use the INSTALL utility to set the appropriate context within the Directory tree.

   If you need to relocate a server, you should delete it and then re-create it in its proper context using the INSTALL utility.

7. Set the appropriate container and property rights.

   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.

The auditing facility also provides a high level of security management.

Security and auditing features can be set up in the NetWare Administrator, NETADMIN, or AUDITCON utility.

8. Manage time synchronization by selecting the type of time server each NetWare server is set up as.

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

   Time synchronization is set up and managed in the INSTALL utility and with SET parameters at the server console or in configuration files.

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

   For security, performance, and reliability reasons, it is a good idea to group servers at various bindery contexts, depending on department or site. If, for example, your organization is spread over three cities, use site-specific bindery contexts for the following reasons:

   - To provide local control over the servers 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 increase security

     If, for example, the network supervisors in three different cities had supervisory rights over the same bindery context container, there would be a great risk of problems due to possible conflicting configurations imposed by the different supervisors.

- 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 users and objects within one of the container units created with INSTALL, you must set the bindery context. The bindery context is automatically set to the location where you install a server. You must enter this command on each server that will use this bindery context.

10. Optimize and manage Directory trees.

    Use DSMERGE and PARTMGR (or the Partition Manager option in the 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.

    Create your tree in a triangular pattern with a small number of partitions at the top levels of the tree and greater numbers, if needed, as you progress towards the bottom of the tree. This implementation creates fewer subordinate references at the top of the 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
Merge partitions	Parent and child partitions

Replicas. Do not create unnecessary replicas of the [Root] partition and other parent partitions. This implementation creates fewer subordinate references at the top of the tree, allowing NDS to keep track of fewer child partition references.

If you keep the network user and resources in mind when creating the tree, you will find that the most efficient use of replications—reducing WAN traffic by placing directory services close to users and resources, and fault tolerance—means you should not need many replicas.

Merge. Before performing a merge operation, ensure that the state of synchronization for all servers affected by the operation is stable. The following table provides recommendations for preparing source and target trees for merging.

Necessary Precondition	Required Action
No aliases or leaf objects can exist at the root of the source tree.	Delete any aliases or leaf objects at the root of the source tree.
No identical names can exist between the source and target trees.	Rename objects on the source and target trees if similar names exist.
No login connections can exist on source or target trees.	Close all connections on the source and target trees.

Necessary Precondition	Required Action
The NDS version must be the same on both source and target trees.	Upgrade all non-4.1 servers that have a replica of the [Root] object.
Any server that contains a replica of the [Root] object on both source and target trees must be up and running.	Ensure that all servers containing a replica of the root partition on both source and target trees are up and running. Ensure that any WAN links affected are stable.
Schema on both source and target trees must be the same.	Ensure that source and target trees are time synchronized within two seconds of each other.

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 5, "Planning NetWare Directory Services Implementation," on page 71 for help in developing an implementation plan for NDS.

Small-Sized Network

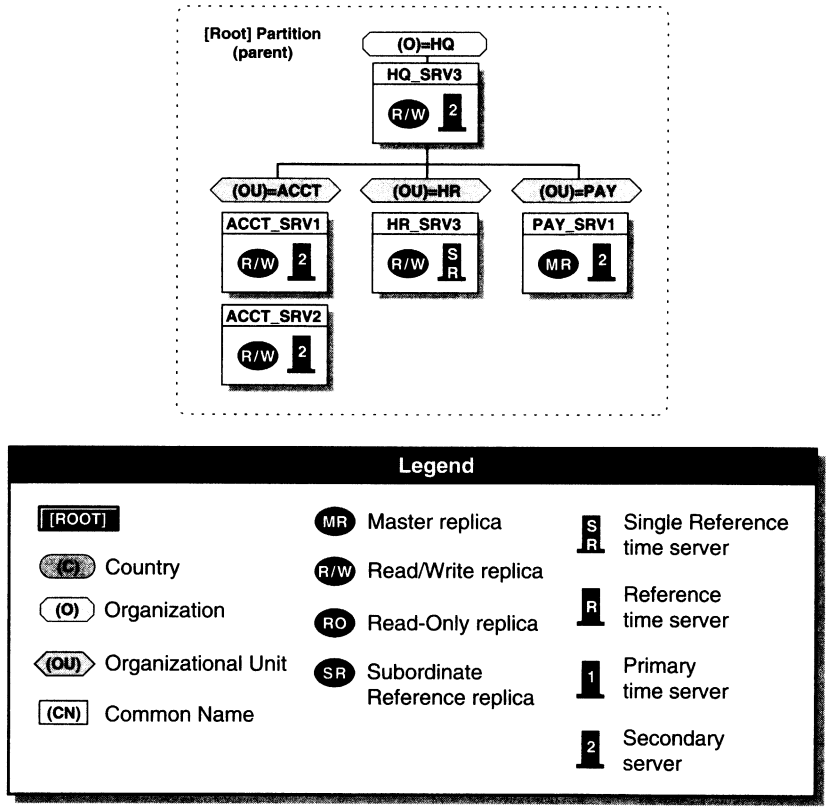
In small-sized networks, the primary reason to implement NDS is the centralized management it provides.

Implementing NDS for centralized management concentrates on two possible NDS 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 6-1
Example of a Small-Sized Directory Tree



Directory Tree Structure

Small-sized networks are commonly site, workgroup, and department oriented in their structure. (See "Planning a Departmental Directory Tree (Merge Tree Method)" on page 79.) They maintain a system-wide administrative group with central management at the organizational and departmental levels.

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

Resources are usually shared by all network users and groups.

Network resources are managed by Organization container supervisors. The supervisors are responsible for managing local resources and user and group accounts.

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 locations of network resources. You should create partitions only if they provide better performance or fault tolerance to the network and tree.

Create your tree in a triangular pattern with a small number of partitions at the top levels of the tree and greater numbers, if needed, as you progress towards the bottom of the tree. This implementation creates fewer subordinate references at the top of the tree.

Replicas

Each server within the network should contain all the resources needed within its location, because users rarely connect to servers in other locations.

If users are connecting to servers in other locations, then make considerations for replicating the Directory partitions based on redundancy for fault tolerance rather than on speed across a WAN.

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

You should allow for all master replicas on a server to be copied to one or two other servers within the same physical location and to a server located at a different site.

Medium-Sized Network

In medium-sized networks, there are four primary reasons to implement NDS:

- Flexibility of the centralized management features

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

- 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, electronic mail, etc.

- High network performance

  NDS allows you to determine how and where network traffic is maintained within the network. Network traffic can remain on a local server by implementing partitioning and replicas.

- Advanced security features

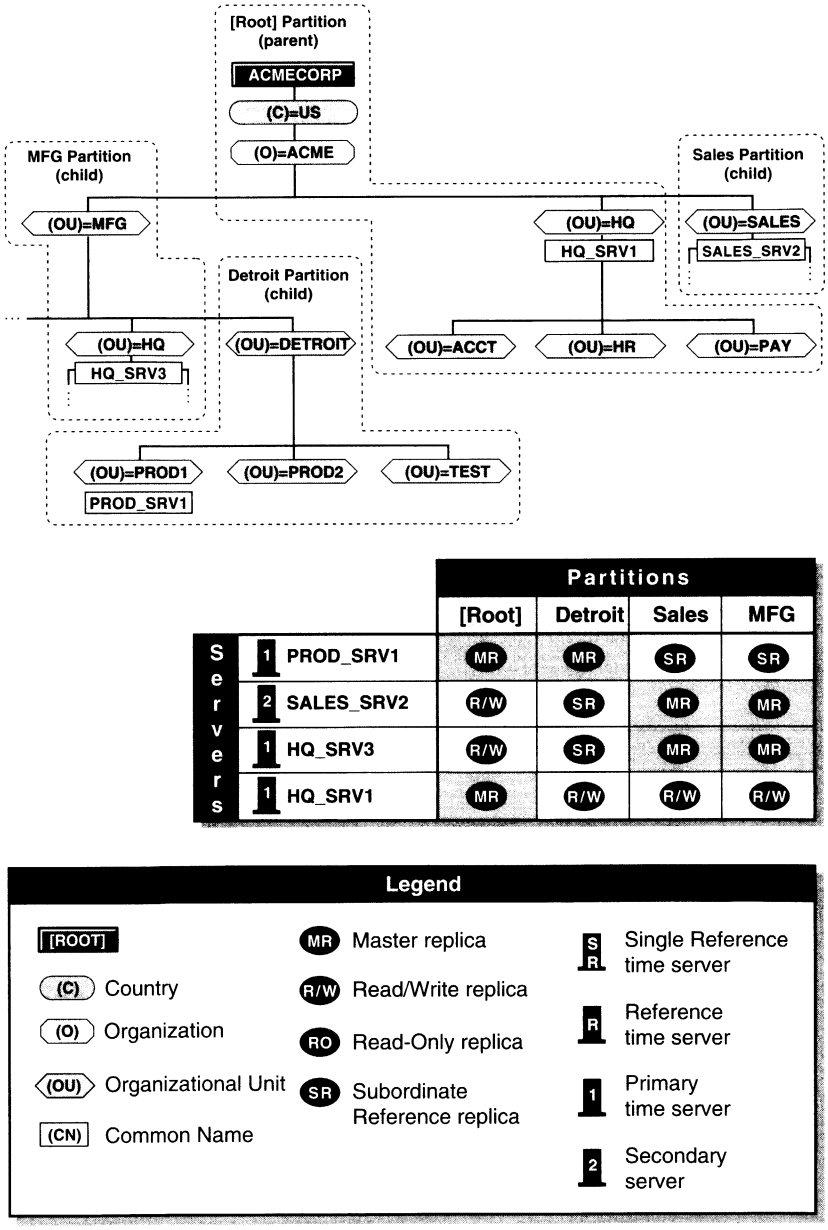
  NDS incorporates the advanced RSA security features that make encrypted, single-login authentication to network resources possible.

  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.

  The auditing facility also provides a high level of security management.

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

Figure 6-2
Example of Medium-Sized Directory Tree



Directory Tree Structure

Medium-sized networks are commonly workgroup and department oriented in their structure. (See "Planning a Departmental Directory Tree (Merge Tree Method)" on page 79.) They maintain a system-wide administrative group with central management at the organizational and departmental levels.

The Directory tree begins with a general Organization (O) object container with multiple Organizational Unit (OU) container objects below. OU containers are based on functional groups, projects, departments, etc.

Within O and OU containers are enterprise resources that are managed centrally. For example:

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

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 accounts for centralized supervisors and Organizational Unit supervisors within their respective containers.

The centralized supervisors are responsible for general network management and overall support for the Directory tree. The OU supervisors are responsible for day-to-day tasks, such as user 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 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 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.

In addition, choose servers in different locations throughout the network.

Set up the remaining servers that you identified as Secondary time servers.

Partitions

Partitioning of medium-sized networks should model the structure of Organizational Unit (OU) containers. Create one partition of each OU within the tree.

This allows each partition to contain all of the resource objects that a particular department needs to access. Place the [Root] and Organization (O) containers 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 directories or applications accessed by multiple departments or the entire organization.

Place replicas of those servers' partitions on servers in different locations in the network that access resources in the partitions. This allows all users to authenticate to an enterprise resource without creating high network traffic.

For servers that provide local services only, place replicas of those servers' partitions on other local servers.

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

Large-Size Networks

In large-sized networks, there are six primary reasons to implement NDS:

- Centralization of resources
- Reduced cost of management, equipment, and critical applications
- Global network resource management and flexibility of the centralized management features

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

- Support for enterprise applications in a seamless manner

  NDS supports enterprise applications, such as demographics research, database applications, human resources/payroll applications, scheduling systems, statistical services applications, document management, electronic mail, etc., regardless of their location on the network.

- High network performance

  NDS allows you to determine how and where network traffic is maintained within the network. Network traffic can remain local by implementing partitioning and replicas.

- Advanced security features

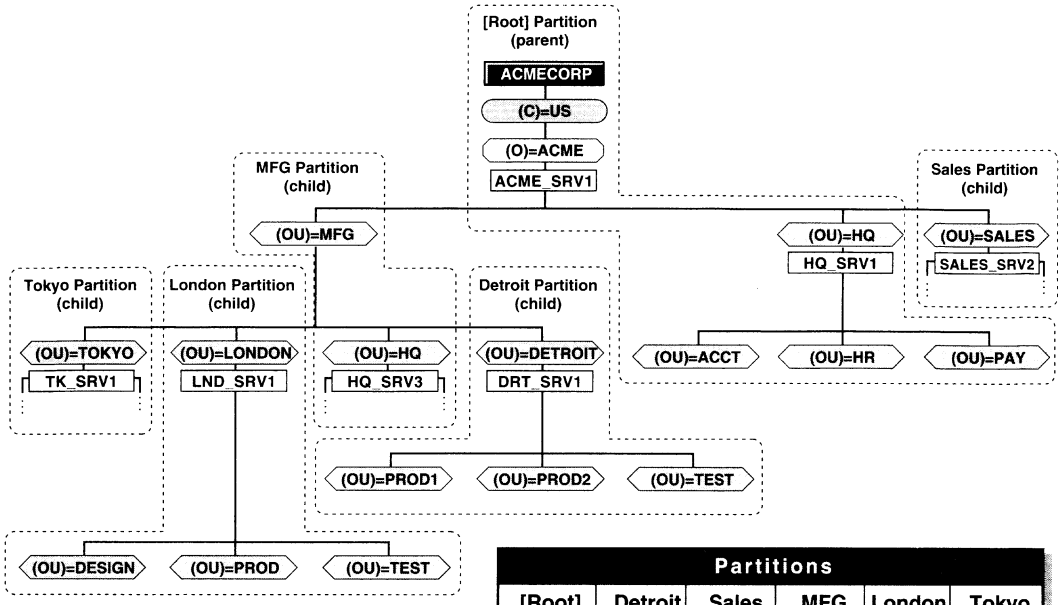
  NDS incorporates the advanced RSA security features that make encrypted, single-login authentication to network resources possible.

  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.

  The auditing facility also provides a high level of security management

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

Figure 6-3
Example of Large-Sized Directory Tree



Servers	Partitions	[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

Legend

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

Directory Tree Structure

Large-sized networks are enterprise focused, linking large, organizational networks with many other equal- or smaller-sized networks. They require a high level of flexibility to merge or split networks as needed.

These networks consist of multiple Directory trees that are constructed to a hybrid model of both functional and location-based structures.

The Directory trees each begin with a general Organization (O) container object with multiple Organizational Unit (OU) container objects below. OU containers are based on functional groups, projects, departments, etc., and also on site locations such as cities or countries.

They maintain both system-wide administrative groups with central management at the organizational and departmental levels and site-based administrative groups that manage local resources and accounts.

They maintain a number of high-level divisions within the organization that form the top layer of OU containers. Most of these divisions are divided into sub-departments which form a second layer of OU containers. A third layer of OU containers consists of locations or functional groups.

Organizational and Departmental Containers

Within O and OU containers are enterprise resources that are managed centrally. For example:

- Servers that function as SAA* or TCP/IP gateways or as a NACS™ system
- User accounts 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 accounts for centralized supervisors and Organizational Unit supervisors within their respective containers.

The centralized supervisors are responsible for general network management and overall support for the Directory tree. The OU supervisors are responsible for day-to-day tasks, such as user 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 is a very important consideration.

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 a single 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 accessed by the entire organization. From the servers you identified, choose which ones you will set up as Primary time servers and which one you will set up as the Reference 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, MD.

In addition, all servers should be adjusted to reduce SAP (Service Advertising Protocol) traffic by making fewer time requests per hour than the default allows.

Partitions

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

Each division OU container has its own partition representing that container and its objects. Each sublevel OU container is the root for a lower level partition that includes itself and all the other containers and leaf objects lower in that branch of the tree.

The [Root] level and the Organization container should form one partition. This partitioning structure ensures that all of 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 directories or applications accessed by multiple departments or the entire organization.

Place replicas of those servers' partitions on servers in different locations in the network that access resources in the partitions. This allows all users to authenticate to an enterprise resource without creating high network traffic.

For servers that provide local services only, place replicas of those servers' partitions on other local servers.

If only one server exists at the location, place one replica of the partition that the server is contained in 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 master server in the Organization level of Directory trees. This allows the central network supervising staff to maintain a complete Directory database in one location.

Also ensure that every partition has up to four replicas available on the network, including replicas on appropriate distant servers.

Place a read/write replica on a master file server, to allow for at least one single location in a tree for all replicas. This enables supervisors to more quickly restore a damaged or missing replica.

Replicas of every partition should serve as an additional form of redundancy as well as a facilitator of reduced network traffic.

Most of the 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 of these servers.

Additional Information

Topic	Reference
Container objects	"Container objects" in *Concepts* "Container Objects" on page 12
Context	"Container objects" in *Concepts* "Context and Names" on page 22
Leaf objects	"Container objects" in *Concepts* "Leaf Objects" on page 14
NetWare Administrator utility	"Installing an MS Windows Workstation and Starting NetWare Administrator" in Chapter 1 of *Supervising the Network* "Installing an OS/2 Workstation and Starting NetWare Administrator" in Chapter 1 of *Supervising the Network*
Objects	"Container objects" in *Concepts* "Directory Objects" on page 8
Partitions and replicas	"Container objects" in *Concepts* "Directory Partitions" on page 34 "Partition Replicas" on page 36
Printing with NetWare Administrator	"Setting Up Print Services with NetWare Administrator" in Chapter 3 of *Supervising the Network*
Rights	"Container objects" in *Concepts* "Object and Property Rights" on page 17

chapter 7

Managing NetWare Directory Services

Overview

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

The following topics are discussed on the indicated pages:

Topic	Page
DSMERGE	132
DSREPAIR	135
DSTRACE	138
INSTALL	139
NETADMIN	141
NetWare Administrator	144
PARTMGR	146
SET (NDS Parameters)	147
TIMESYNC	149
UIMPORT	151

Introduction

The 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 single information system is the basis for increased productivity and reduced administrative costs.

The management utilities and programs discussed in this chapter can help you build and maintain your Directory Tree hierarchy and objects, as well as helping you maintain the Directory databases within your network.

DSMERGE

Use this utility at the server console to

- Merge the roots of two separate Directory trees
- Rename a tree
- View name and time synchronization information

Creating one Directory tree from two separate trees allows for communicating and sharing data.

The following table shows the functions available in DSMERGE.

Option	Use to
Check Servers in This Tree	Contact all servers in the local tree to verify that each server has the correct version, status, and tree name. The server you are on must have a replica of the [Root] partition. It does not require the master replica.
Check Time Synchronization	Display a list of all servers in this tree, along with information about time sources and time synchronization. The server you are on must have a replica of the [Root] partition. It does not require the master replica.

Option	Use to
Merge Two Trees	Merge the [Root] of the local (source) tree to the [Root] of the target tree. The server you are on must have the master replica of the local tree's [Root] partition.
Rename Tree	Rename the local tree. Use this option if you are merging two roots with the same name. You can rename only the local tree name. To rename the target tree name, load DSMERGE on a server in the target tree. Then load DSMERGE on the source tree to perform the merge. The server you are on must have the master replica of the root partition.

Using the DSMERGE Utility

The DSMERGE utility requires that the following conditions exist before two Directory trees can be merged:

- No leaf object or aliases can exist at the root of the source tree
- No similar names can exist on source and target trees
- No login connections can exist on either source or target tree
- The version of NDS must be the same on both source and target trees
- Any server that contains a replica of the [Root] for both source and target trees must be up and running
- Schema on both source and target trees must be the same
- All servers in both source and target trees should be synchronized within two seconds of each other, and all servers should use the same time source

You can't merge container or leaf objects with DSMERGE.

To move leaf objects, use NetWare Administrator or NETADMIN. To merge partitions, use PARTMGR or Partition Manager in NetWare Administrator.

Completing the Tree Merge

Following the merging of two trees, it might be necessary to complete the following tasks:

- If any servers were not upgraded to NetWare 4.1 before you ran DSMERGE, copy a new replica to those servers.
- Re-create any leaf objects or aliases at the [Root] that were deleted before you ran DSMERGE.
- Carefully evaluate and change the partitioning as needed, because merging trees might significantly change replica placement on the source tree
- Update client workstations' PREFERRED TREE statements in their NET.CFG files, or rename the target tree so that the final tree name corresponds to the majority of the client workstations' NET.CFG files.

Additional Information

Topic	Reference
DSMERGE utility	"Merging NDS Trees" in Chapter 5 of *Supervising the Network* "DSMERGE" in *Utilities Reference*
Merging partitions	"Merging Partitions" in Chapter 5 of *Supervising the Network*
Moving leaf objects	"Moving Objects in the Directory Tree" in Chapter 5 of *Supervising the Network*
Renaming a Directory tree	"Renaming the Tree" in Chapter 5 of *Supervising the Network*
Time synchronization	"Managing Network Time Synchronization" in Chapter 7 of *Supervising the Network*

DSREPAIR

Use this utility at the server console to check and repair or correct such problems in the Directory database concerning records, schema, bindery objects, and external references.

It is similar to the way VREPAIR fixes volumes on a server.

The following table shows the functions available in DSREPAIR.

Option	Use to
Unattended Full Repair	Automatically perform all possible repair operations to the Directory database that do not require operator assistance.
Time Synchronization	Contact all servers within this server's local database to request information about Directory Services and time synchronization. If a replica of the root partition is contained on this server, then all servers in the Directory tree are contacted.

Option	Use to
Replica Synchronization	Determine the status of synchronization for every replica in the replica table for the Directory tree. The status of synchronization can inform you of the current condition of the Directory tree.
View/Edit Repair Log File	Track all operations of the DSREPAIR utility to a single file. The default log file is SYS:SYSTEM\DSREPAIR.LOG. You can configure options for the log file by accessing "Log File And Login Configuration" in the "Advanced Options" menu.
Advanced Options Menu	The "Advanced Options" menu allows you to manually perform individual or global repair operations on the Directory tree. You can also access diagnostic information about the Directory tree database to analyze the status of the tree. The "Advanced Options" menu includes the following options: ◆ Repair local database ◆ Repair all known network addresses ◆ View, verify, and edit remote server ID list ◆ Repair replicas, replica lists (rings), and server objects ◆ Synchronize security equivalence attributes for tree ◆ Update schema ◆ View local partition information ◆ View and edit log ◆ Copy compressed NDS database files to disk

Using the DSREPAIR Utility

The DSREPAIR utility is provided with NetWare 4™ software to repair problems with NetWare Directory Services on an individual-server basis (not from a single, centralized location).

You can do any of the following with DSREPAIR:

- Check Directory information
  - Local Directory tree structure
  - Records that make up the Directory on the server
  - Directory schema
  - Objects in replicas
  - Local references
  - Mandatory and optional properties
  - Initial states
  - File system trustees on mounted volumes
  - Mail directories
  - Stream syntax files
- Designate a new master replica for a partition that has lost this replica due to server failure
- Check or perform replica or schema synchronization
- Repair all initial states
- Guarantee local Database recovery
- Remove lost Server objects
- Configure time synchronization

Additional Information

Topic	Reference
DSREPAIR utility	"Repairing the NetWare Directory Database" in Chapter 5 of *Supervising the Network.* "DSREPAIR" in *Utilities Reference*

DSTRACE

Use this utility at the server console to

- Determine whether NDS synchronization processes are complete
- Diagnose NDS errors

Using the DSTRACE Utility

DSTRACE is a simple console utility to monitor how NDS is functioning. It is not intended to be used as an end-user tool, but it can often be used to help identify network problems which might be related to NDS.

Comparing trace information from server to server can help determine the source of a Directory problem.

The NDS TRACE TO FILE parameter is also useful for capturing information that you might want to have when calling your Novell Authorized Reseller[CLM] or Novell Technical Support[SM] representative for assistance.

Additional Information

Topic	Reference
DSTRACE utility	"NetWare Directory Services Parameters" under "SET" in *Utilities Reference* "Viewing and Managing NDS Synchronization Status" in Chapter 5 of *Supervising the Network.*
NDS system messages	Messages numbering –601 through –699 and F966 through F9FE)in System Messages

INSTALL

Use this program at the server console to

- Install or upgrade the NetWare operating system on your server
- Modify your NetWare server configuration
- Perform some server maintenance operations

Using the INSTALL Program

During the installation process, 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 INSTALL program assists you in setting up time synchronization.

It also assists you in naming the server and in setting up the server context or name context. This determines the location of the server in the Directory tree.

Use the INSTALL program to create partitions for the first server in a tree or to upgrade from a bindery server. Then use INSTALL to install the server into the partition you created.

For additional servers, use the PARTMGR utility or Partition Manager in NetWare Administrator to create a new partition before installing a new server into the tree.

If you want to install a server into an Organization (O), create the O before you install the server. If you want the O to become a new partition, you can split it off before or after INSTALL is completed.

Make sure that you install the server into a context that currently exists. Do not create a new context within INSTALL.

Removing NDS with the INSTALL Program

When experiencing problems with NDS, you should not attempt to unload the software and then reload it. Doing this might unsynchronize Directory tree.

Rarely is it necessary to remove the NDS software. You should remove NDS only if you are sure that doing so will assist you in recovering your Directory tree.

Try correcting any problems first by removing and reinstalling replicas with the partition management utilities. Remove the NDS software only as a last resort.

Additional Information

Topic	Reference
INSTALL utility	"INSTALL" in *Utilities Reference*
Installing NetWare	"Install Server Software" in Chapter 2 of *Installation*
Partition management utilities	"Creating and Managing Directory Services Partitions" in Chapter 5 of *Supervising the Network*
	"PARTMGR" in *Utilities Reference*
Removing NDS software	"Managing the NetWare Directory Tree" of Chapter 5 in Supervising the Network
Upgrading a NetWare server	"Upgrade Options" in Chapter 1 of *Upgrade*

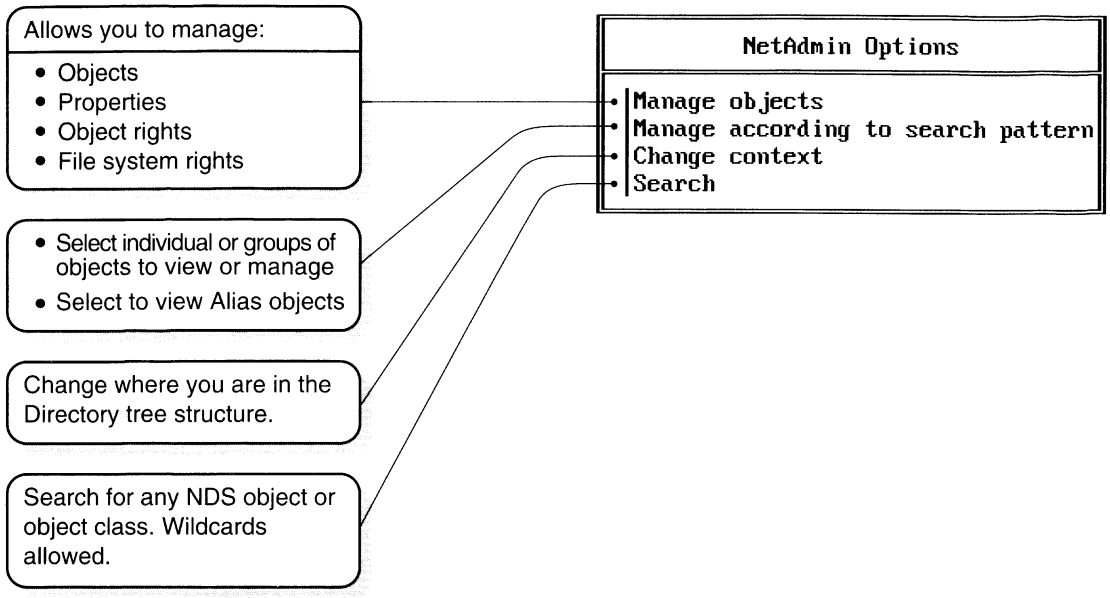
NETADMIN

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 appropriate rights to. This utility helps you manage access to these objects.

The following figure shows the functions you can select from the "NetAdmin Options" main menu.

Figure 7-1
Functions in NETADMIN



Using the NETADMIN Utility

You can perform the following management tasks with NETADMIN:

- Change object property values
- Create and name container and leaf objects
- Create searchable 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 and container objects
- Search for objects
- Manage object and property rights

Additional Information

Topic	Reference
Container objects	"Container object" in *Concepts*
Context	"Context" in *Concepts*
Leaf objects	"Leaf objects" in *Concepts*
NETADMIN utility	"NETADMIN" in *Utilities Reference*
Objects	"Objects" in *Concepts*
	Appendix A, "NDS Object Classes and Properties" on page 155
Rights	"Default Objects and Rights for NetWare 4.1" in Chapter 1 *Supervising the Network*

NetWare Administrator

Use this 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 appropriate rights to. This utility helps you manage access to these objects.

NetWare Administrator is a graphical user interface utility that runs as a multiple-document interface (MDI) application.

Before you use NetWare Administrator for the first time in MS Windows or OS/2, create an "NWADMIN" icon. Thereafter, you can choose the icon to start the utility.

Using the NetWare Administrator Utility

You can perform the following tasks in MS Windows or OS/2 Windows, or 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 Directory objects
- Set up printing services
- Set up and manage Directory partitions and replicas

Additional Information

Topic	Reference
Container objects	"Container object" in *Concepts* "Container Objects" on page 12
Context	"Context" in *Concepts* "Context and Names" on page 22
Leaf objects	"Leaf objects" in *Concepts* "Leaf Objects" on page 14
NetWare Administrator utility	"Installing an MS Windows Workstation and Starting NetWare Administrator" in Chapter 1 of *Supervising the Network* "Installing an OS/2 Workstation and Starting NetWare Administrator" in Chapter 1 of *Supervising the Network*
Objects	"Object" in *Concepts* "Directory Objects" on page 8
Partitions and replicas	"Partition (NetWare Directory)" in *Concepts* "Directory Partitions" on page 34 "Partition Replicas" on page 36
Printing with NetWare Administrator	"Setting Up Print Services with NetWare Administrator" in Chapter 3 of *Supervising the Network*
Rights	"Rights" in *Concepts* "Object and Property Rights" on page 17

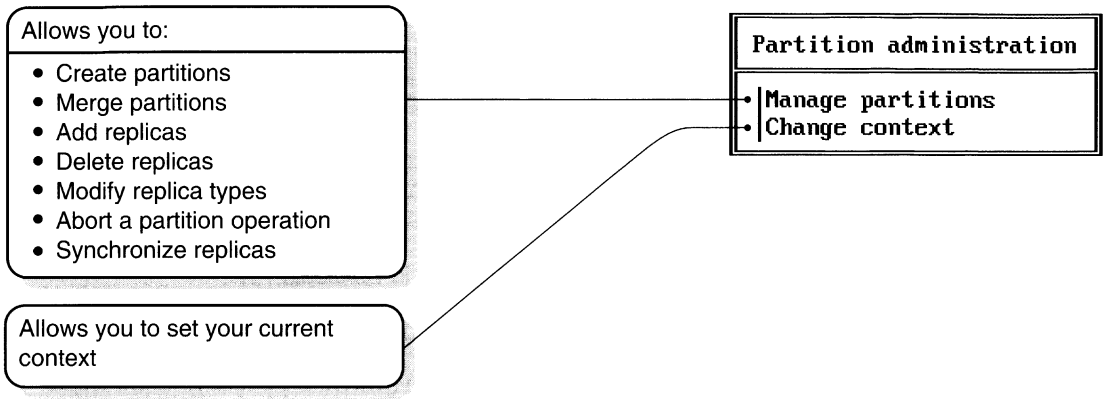
PARTMGR

Use this utility at a client workstation to

- Distribute your Directory database
- Manage partitions and replicas

The following figure shows the functions available in PARTMGR.

Figure 7-2
Functions in PARTMGR



Using the PARTMGR Utility

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 object 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
- Merge a partition with its parent partition
- Abort a partition operation in progress

Additional Information

Topic	Reference
Partition and replica operations	"Managing the NetWare Directory Tree" in Chapter 5 of *Supervising the Network*
Partitions	"Partition" in *Concepts*
PARTMGR utility	"PARTMGR" in *Utilities Reference*
Replicas	"Replica" in *Concepts*

SET (NDS Parameters)

Use this utility at the server console to view and configure NDS parameters.

The default SET parameters give maximum performance for most systems. Network supervisors should rarely need to modify parameters.

Using the SET Utility (NDS Parameters)

Although most parameters don't need to be modified, you might increase the performance of your system by changing some parameters.

NDS SET parameters enable you to perform the following tasks:

- Control the NDS trace file
- Set time intervals for maintenance processes that reclaim disk space, remove external references, and check the consistency of backlinks
- Set NDS synchronization intervals and restrictions
- Specify the number of NetWare Core Protocol™ (NCP) packet retries before timing out
- Mark the status of other servers in the namebase as UP or DOWN
- Specify bindery services contexts

You can use SET parameters in two ways: at the system console and in the configuration file.

Entering at the System Console

When a parameter is used at the console prompt, the system is immediately configured to that setting.

This overrides any related setting in the AUTOEXEC.NCF file until the server is rebooted. The system then returns to the default setting for the parameter.

Saving in the AUTOEXEC.NCF File

When a parameter is saved in the AUTOEXEC.NCF file, the server configures itself to the parameter each time the server is booted.

You can edit the AUTOEXEC.NCF file with INSTALL. If you use SERVMAN to change SET parameters, you are prompted to update .NCF files before exiting the utility.

Additional Information

Topic	Reference
Editing AUTOEXEC.NCF file	"INSTALL" in *Utilities Reference* "SERVMAN" in *Utilities Reference*
Improving server performance	"Monitoring and Optimizing the Server" in Chapter 7 of *Supervising the Network*
NDS SET parameters and options	"NetWare Directory Services Parameters" under "SET" in *Utilities Reference*

TIMESYNC

Use this utility at the server console to monitor the internal time on a server to ensure that the time reported by all servers across the network is consistent (synchronized).

TIMESYNC autoloads when the server boots. Rarely will you need to load or unload TIMESYNC. You might, however, want to load TIMESYNC to use an alternate configuration (.CFG) file.

Using the TYMESYNC Utility

Time synchronization ensures that all servers in a Directory tree report the same time and order NetWare Directory Services events correctly.

NetWare 4.1 uses TIMESYNC, which automatically loads when the server is booted, to control time synchronization.

TIMESYNC uses two types of SET parameters:

- Those which control how time synchronization works

  These are set in the TIMESYNC.CFG configuration file.

- Those which determine how the system handles time synchronization

  These are either entered at the system console or set in the AUTOEXEC.NCF file.

Using SET Parameters in the TIMESYNC.CFG File

Time synchronization SET parameters that control how the feature works are stored in their own configuration file, by default named TIMESYNC.CFG, in the SYS:SYSTEM directory.

To change the time synchronization configuration on a NetWare server, modify the parameters in the TIMESYNC.CFG file, and then reboot the server or set TIMESYNC RESTART FLAG=ON.

Using SET Parameters at the System Console or in the AUTOEXEC.NCF File

The time synchronization parameters that determine how the system handles time synchronization can be entered at the system console. But unless you also add them to the AUTOEXEC.NCF file, changes made with SET at the console are lost the next time the server boots.

Additional Information

Topic	Reference
Editing TIMESYNC.NCF file	"Editing the TIMESYNC.CFG File" in Chapter 7 of *Supervising the Network*
Modifying TIMESYNC SET parameters for medium or large networks	"Choosing a Time Synchronization Method" on page 62
SAP	"Service Advertising Protocol" in *Concepts*
SET parameters and options for TIMESYNC	"SET Parameters for TIMESYNC" in Chapter 7 of *Supervising the Network*
Time server types	"Time Servers" on page 55
Time synchronization	"Time synchronization" in *Concepts*
TIMESYNC utility	"TIMESYNC" in *Utilities Reference*

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 the UIMPORT Utility

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 Directory database using records from another database
- Update User properties in the Directory 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 Directory Services Database" in Chapter 5 of *Supervising the Network*

Appendixes

Overview

The NetWare® Directory tree supports a large number of object classes and properties available to the NetWare Directory Services™ and bindery services technologies.

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	Page
To reference lists and explanations of the object classes and properties available in NetWare Directory Services	Appendix A, "NDS Object Classes and Properties"	155
To reference lists of available leaf objects in NetWare Directory Services	Appendix B, "Referencing and Using Leaf Objects"	167
To reference guidelines and samples for creating a standards document for objects in NetWare Directory Services	Appendix C, "Creating a Standards Document for NDS Object Classes and Properties"	175

appendix **A**

NDS Object Classes and Properties

Overview

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

The following topics are discussed on the indicated pages:

Topic	Page
NDS Object Classes and Their Functions	156
NDS Object Classes and Their Properties	158

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 A-1
Object Class, Function, and Possible Container

Object Class	Function	Possible Container
AFP Server	Represents an AppleTalk File Protocol-based server that is operating as a node on your NetWare network—and possibly also acting as a NetWare router to, and as the AppleTalk server for, several Apple Macintosh workstations.	Organization Organizational Unit
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
External Entity	Used by services (such as messaging) to store information about entities (such as E-mail users) outside of Directory	Organization Organizational Unit
Group	Defines unordered list of users that comprise group for purpose of assigning access rights	Organization Organizational Unit
List	Defines unordered set of names without implying security equivalence	Organization Organizational Unit

Table A-1 *continued*

Object Class, Function, and Possible Container

Object Class	Function	Possible Container
Message Routing Group	Represents group of messaging servers with direct connectivity for transferring messages between any two of them	Organization Organizational Unit
Messaging Server	Represents server that picks up messages submitted by messaging applications or transferred from other messaging server	Organization Organizational Unit
NetWare Server	Represents server that provides file and other services	Organization Organizational Unit
Organization	Defines organization within network	Country, Root, or Locality 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
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 A-2
Object Class and Properties

Object Class	Properties	
AFP Server	ACL	Object Class
	Account Balance	OU
	Allow Unlimited Credit	Private Key
	Back Link	Public Key
	Bindery Property	Resource
	CN	See Also
	Descriptions	Security Equals To
	Full Name	Security Flags
	Host Device	Serial Number
	L	Status
	Minimum Account Balance	Supported Connections
	Network Address	User
	O	Version
Alias	ACL	Bindery Property
	Aliased Object Name	Object Class
	Back Link	
Bindery Object	ACL	Bindery Type
	Back Link	CN
	Bindery Object Restrictions	Object Class
	Bindery Property	
Bindery Queue	ACL	Network Address
	Back Link	O
	Bindery Property	Object Class
	Bindery Type	Operator
	CN	OU
	Description	Queue Directory
	Device	See Also
	Host Resource Name	Server
	Host Server	User
	L	Volume

Table A-2 *continued*

Object Class and Properties

Object Class	Properties	
Computer	ACL	Object Class
	Back Link	Operator
	Bindery Property	OU
	CN	Owner
	Description	See Also
	L	Serial Number
	Network Address	Server
	O	Status
Country	ACL	C
	Back Link	Description
	Bindery Property	Object Class
Directory Map	ACL	L
	Back Link	Name
	Bindery Property	O
	CN	Object Class
	Description	OU
	Host Resource	Path
	Host Server	See Also
External Entity	ACL	L
	Authority Revocation	Last Referenced Time
	Back Link	Obituary
	Bindery Property	Object Class
	CA Private Key	OU
	CA Public Key	Physical Delivery Office Name
	Certificate Revocation	Postal Address
	Certificate Validity Interval	Postal Code
	CN	Reference
	Cross Certificate Pair	Revision
	Description	S
	E-Mail Address	SA
	External Name	See Also
	Facsimile Telephone Number	Title
	Group Membership	

Table A-2 *continued*

Object Class and Properties

Object Class	Properties	
Group	ACL	Mailbox ID
	Back Link	Mailbox Location
	Bindery Property	Member
	CN	O
	Description	Object Class
	E-Mail Address	OU
	Full Name	Owner
	GID	Profile
	L	Profile Membership
	Login Script	See Also
List	ACL	Last Referenced Time
	Authority Revocation	Mailbox ID
	Back Link	Member
	Bindery Property	O
	CA Private Key	Obituary
	CA Public Key	Object Class
	Certificate Revocation	OU
	Certificate Validity Interval	Owner
	CN	Reference
	Cross Certificate Pair	Revision
	Description	See Also
	L	
Message Routing Group	ACL	Last Referenced Time
	Authority Revocation	Login Script
	Back Link	Mailbox ID
	Bindery Property	Mailbox Location
	CA Private Key	Member
	CA Public Key	O
	Certificate Revocation	Obituary
	Certificate Validity Interval	Object Class
	CN	OU
	Cross Certificate Pair	Owner
	Description	Profile
	E-Mail Address	Profile Membership
	Full Name	Reference
	GID	Revision
	L	See Also

Table A-2 *continued*

Object Class and Properties

Object Class	Properties	
Messaging Server	Account Balance	Messaging Server Type
	ACL	Minimum Account Balance
	Allow Unlimited Credit	Network Address
	Authority Revocation	O
	Back Link	Obituary
	Bindery Property	Object Class
	CA Private Key	OU
	CA Public Key	Postmaster
	Certificate Revocation	Private Key
	Certificate Validity Interval	Public Key
	CN	Reference
	Cross Certificate Pair	Resource
	Description	Revision
	Full Name	Security Equals To
	Host Device	Security Flags
	L	See Also
	Last Referenced Time	Status
	Message Routing Group	Supported Services
	Messaging Database Location	User Version
NCP Server	Account Balance	Object Class
	ACL	Operator
	Allow Unlimited Credit	OU
	Back Link	Private Key
	Bindery Property	Public Key
	CN	Resource
	DS Revision	See Also
	Full Name	Security Equals To
	Host Device	Security Flags
	L	Status
	Messaging Server	Supported Services
	Minimum Account Balance	User
	Network Address	Version
	O	

Table A-2 *continued*

Object Class and Properties

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

Table A-2 *continued*

Object Class and Properties

Object Class	Properties	
Print Server	Account Balance	Operator
	ACL	OU
	Allow Unlimited Credit	Print
	Back Link	Private Key
	Bindery Property	Public Key
	CN	Resource
	Description	SAP Name
	Full Name	Security Equals To
	Host Device	Security Flags
	L	See Also
	Minimum Account Balance	Status
	Network Address	User
	O	Version
	Object Class	
Printer	ACL	O
	Back Link	Object Class
	Bindery Property	Operator
	Cartridge	OU
	CN	Owner
	Default Queue	Page Description Language
	Description	Print Server
	Host Device	Printer Configuration
	L	Queue
	Memory	See Also
	Network Address	Serial Number
	Network Address Restrictions	Status
	Notify	Supported Typefaces
Profile	ACL	Login Script
	Back Link	O
	Bindery Property	Object Class
	CN	OU
	Description	See Also
	L	

Table A-2 *continued*

Object Class and Properties

Object Class	Properties	
Queue	ACL	O
	Back Link	Object Class
	Bindery Property	Operator
	CN	OU
	Description	Queue Directory
	Device	See Also
	Host Resource Name	Server
	Host Server	User
	L	Volume
	Network Address	
Unknown	ACL	Bindery Property
	Back Link	Object Class

Table A-2 *continued*

Object Class and Properties

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

appendix **B**

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:

Topic	Page
User-Related Leaf Objects	168
Server-Related Leaf Objects	170
Printer-Related Leaf Objects	171
Messaging-Related Leaf Objects	172
Informational Leaf Objects	173
Miscellaneous Leaf Objects	174

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 B-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.
Message Routing Group	Represents a group of messaging servers that can transfer messages directly with each other.	Multiple messaging servers need to communicate with one another.
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 B-1 *continued*

User-Related Leaf Object Name, Function, and Usage

Leaf Object	Function	Usage Situation
User	Represents a person who uses the network.	Required for every user who needs to log in to the network.
	In the User object properties, login restrictions, intruder detection limits, password and password restrictions, security equivalences, etc., can be set.	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 4.1 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 B-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 B-2 *continued*

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 NetWare 4.1 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 B-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. It cannot be created with NETADMIN. See *Print Services* for more information.
Print Server	Represents a network print server.	Required for every print server on the network. It cannot be created with NETADMIN. See *Print Services* for more information.

Table B-3 *continued*

Printer-Related Leaf Object Name, Function, and Usage

Leaf Object	Function	Usage Situation
Printer	Represents a physical printing device on the network.	Required for every printer on the network. It cannot be created with NETADMIN. See *Print Services* for more information.

Messaging-Related Leaf Objects

This section lists the available leaf objects that are related to the NetWare Message Handling Service™ (MHS) system, explains what each is used for, and indicates when to use each.

These objects are created and controlled using the MHS utilities.

Table B-4

Messaging-Related Leaf Object Name, Function, and Usage

Leaf Object	Function	Usage Situation
Distribution List	Represents a list of mail recipients.	You want to simplify sending mail. For example, you could create a Distribution List object called "Recreation Committee." Then to send a message to all the members in this committee, simply address the message to "Recreation Committee" rather than to each member individually.
External Entity	Represents a non-native NDS object that is imported into NDS or registered in NDS. The NetWare MHS™ system uses this object to represent users from bindery-based directories to provide an integrated address book for sending mail.	If your messaging environment contains non-MHS servers (such as SMTP hosts, SNADS nodes, or X.400 MTAs), you could add users and lists from these servers to your NetWare database as External Entities. This would add them to the address books of your messaging applications. Then, when addressing messages, local users can choose non-MHS users and lists from a directory list.

Table B-4 *continued*

Messaging-Related Leaf Object Name, Function, and Usage

Leaf Object	Function	Usage Situation
Message Routing Group	Represents a group of messaging servers that can transfer messages directly among each other.	You have multiple messaging servers that need to communicate with each other.
Messaging Server	Represents a messaging server that resides on a NetWare server.	Automatically created in the Directory tree during NetWare MHS installation on a NetWare server.

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 B-5

Informational Leaf Object Name, Function, and Usage

Leaf Object	Function	Usage Situation
AFP Server	Represents an AppleTalk File Protocol-based server that is operating as a node on your NetWare network—and possibly also acting as a NetWare router to, and as the AppleTalk server for, several Apple Macintosh workstations.	You have an AFP server that you need represented on the network. Use this object to store information about the server—such as its description, location, and network address. This object has no effect on the network operations; it only stores information about the AFP server.
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 B-6
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. Also, when you move or rename a container object in a Directory tree, you have the option of creating an Alias in place of the moved or renamed object. If you select this option, NetWare Administrator automatically creates the Alias for you and assigns it the same name as the original object. Creating an alias in place of a moved or renamed container object allows users to continue logging in to the network and to see the container object (and the objects it contains) in its original Directory location.
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.

appendix **C**

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 Service™ database.

The following topics are discussed on the indicated pages:

Topic	Page
Sample Object Naming Standards	176
Sample Object Property Standards	178

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 C-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 printrer 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 C-1 *continued*

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".
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 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.

Across-the-Wire migration

A method of upgrading using the NetWare Migration Utility. In an across-the-wire migration, data files are migrated across the network to the destination server.

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.

AFP Server object

An object that represents an AppleTalk Filing Protocol (AFP) server that allows workstations to share files and programs.

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.

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 within its container 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 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.

DSREPAIR

An NLM program that repairs and 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* or OS/2 Presentation Manager*.

Group object

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

High Capacity Storage System (HCSS)

Extends the storage capacity of a NetWare server by integrating an optical disk library, or *jukebox*, into the NetWare file system.

HCSS moves files between faster low-capacity storage devices (the server's hard disk) and slower high-capacity devices (optical disks in a jukebox).

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.

INSTALL

An NLM program used to initiate the installation of NetWare 4 software.

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 parts.

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 organize the 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

The replica that contains the first instance of partition information used to change the structure of the Directory in relation to that partition.

name type

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

NCF file

A NetWare command file that can be used by the network supervisor to automate server commands.

Two special NCF files (STARTUP.NCF and AUTOEXEC.NCF) can be used to add commands to a server's boot process. NCF files are comparable to DOS batch files.

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. However, 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 Loadable Module™ (NLM)

A program you can load and unload from server memory while the server is running. NLM™ software are disk drivers, LAN drivers, name spaces, and other NetWare server management and enhancement utilities used by the operating system.

NetWare server

A computer running the NetWare operating system software, regulating communications among the personal computers attached to it, and managing shared resources such as printers and volumes.

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.

NLM

See **NetWare Loadable Module (NLM).**

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

Access rights to an object that are assigned to another object (which becomes a trustee of the object). Object rights don't affect properties or property rights, with the exception of the Supervisor object right, which also allows access to all property values.

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

The original partition. A partition that has a boundary in the Directory tree above another partition.

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 volume 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)

A right that grants the right to read and compare the values of a property.

read-only replica

A type of replica that can be used only to read NetWare Directory Services database information.

read/write replica

A type of replica that can be used to read or update NetWare Directory Services database information. It cannot be used to create a new partition.

fReference 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. Changing other parts of the name implies a move operation.

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 four types of replica: master, read-only, read/write, and secondary.

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 partition.

Supervisor object right (S)

Grants all access privileges. A trustee who has the Supervisor right automatically has access to all properties.

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 in the NetWare 4 software, 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 four types of time source servers: Single Reference, Primary, and Reference.

time stamp

A unique code that identifies an event and includes the time is occurred. It is reported by the Directory tree at the time of the event, such as a password change or an object renaming.

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.

TIMESYNC

An NLM program that controls time synchronization on servers running NetWare Directory Services.

See also **time synchronization**.

tree

See **Directory tree**.

tree name

See **Directory tree name**; **true 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

Grants rights to an object to perform actions on another object or its properties, on a file, or on a directory.

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.

true name

The default name of the [Root] object when it is automatically placed at the top of the tree by the NetWare 4 installation program.

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 create many User objects at the same time, using a 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 comprise 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.

Trademarks

Novell, Inc. has attempted to supply trademark information about company names, products, and services mentioned in this manual. The following list of trademarks was derived from various sources.

Novell Trademarks

Certified NetWare Engineer and CNE are service marks of Novell, Inc.

LANalyzer is a registered trademark of Novell, Inc.

N-Design is a registered trademark of Novell, Inc.

NetWare is a registered trademark of Novell, Inc.

NetWare 3 is a trademark of Novell, Inc.

NetWare 4 is a trademark of Novell, Inc.

NetWare Core Protocol and NCP are trademarks of Novell, Inc.

NetWare Directory Services and NDS are trademarks of Novell, Inc.

NetWare DOS Requester is a trademark of Novell, Inc.

NetWare Loadable Module and NLM are trademarks of Novell, Inc.

NetWare Logotype is a trademark of Novell, Inc.

NetWare Management System and NMS are trademarks of Novell, Inc.

NetWare Message Handling Service and NetWare MHS are trademarks of Novell, Inc.

NetWire is a registered service mark of Novell, Inc.

Network Support Encyclopedia Professional Volume and NSE Pro are service marks of Novell, Inc.

Novell is a registered trademark of Novell, Inc.

Novell Authorized Reseller is a collective mark of Novell, Inc.

Novell Authorized Education Center and NAEC are collective marks of Novell, Inc.

Novell Authorized Service Center and NACS are service marks of Novell, Inc.

Novell Press is a trademark of Novell, Inc.

Novell Research Reports is a trademark of Novell, Inc.

Novell Technical Support is a service mark of Novell, Inc.

Transaction Tracking System and TTS are trademarks of Novell, Inc.

UNIX is a registered trademark of Novell, Inc. in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

Third-Party Trademarks

CompuServe is a registered trademark of CompuServe Incorporated.

DynaText is a trademark of Electronic Book Technologies, Inc.

LaserJet is a registered trademark of Hewlett-Packard Company.

Microsoft is a registered trademark of Microsoft Corporation.

SAA is a trademark of International Business Machines Corporation.

Unicode is a registered trademark of Transoft Ltd.

Index

A

Access Control List (ACL), explained 20. *See also* Security
Access to Directory tree, controlling 100. *See also* Security
Accessing
  multiple servers in same bindery context 49
  objects in different bindery contexts 51
Account Restrictions information property
  standards suggestions 178. *See also* Security
ACL. *See* Access Control List
Add or Delete Self property right, explained 19. *See also* Rights
ADMIN, User object (explained) 32. *See also* Objects
AFP Server object, explained 173. *See also* Objects
Alias object. *See also* Objects
  explained 174
  using 24
Authentication, explained 25. *See also* Security
AUTOEXEC.NCF file, setting bindery context in 43, 45

B

Backups, providing Directory fault tolerance with tape 96
Bindery context
  changing, caution 49
  changing, explained 104
  defined 41
  removing container from, caution 49
Bindery context, setting
  in AUTOEXEC.NCF 43, 45
  multiple, in same Directory tree (with SET) 52
  multiple, overview 42
  for multiple servers in same context 49
  in multiple-level Directory tree 46
  necessary for NetWare Directory Services 43
  for objects in different contexts 51
  to replica 44
  with SET 43, 148
  in single-level Directory tree 45
  for specific server 47
Bindery emulation. *See* Bindery services
Bindery Object object, explained 174. *See also* Objects
Bindery objects. *See also* Objects
  created/upgraded under NetWare Directory Services 43
  moving 105
Bindery Queue object, explained 174. *See also* Objects
Bindery services
  context, specifying with SET 148
  Directory tree organization suggestion, if using 83
  disabling, explained 43
  illustrated 42
  integration strategy, developing 103
  managing 104
  setting up, considerations 43
  structure within Directory tree, guidelines 83 (*see also* Directory tree, planning)
  when NetWare Directory Services can't support 43

Bindery services, using with NetWare Directory Services
bindery context not set, caution 43
inaccessible information 44
limited partitioning 44
objects created/upgraded 43
overview 41
Browse object right, explained 18. *See also* Rights

C

Changing. *See also* Modifying
bindery context, caution 49
bindery context, explained 104
context, explained 26
Directory tree structure, considering bindery services when 105
Directory tree structure, explained 80
names after implementing NetWare Directory Services 78
Characters, using in names
special 27
Unicode 27, 29
Child partition, defined 34. *See also* Partitions
Classes, object. *See* Object classes
Combining (merging) Directory trees. *See* DSMERGE
Command, SET. *See* SET
Common name, defined 25
Compare property right, explained 19. *See also* Rights
Complete name. *See* Distinguished Name
Computer object, explained 173. *See also* Objects
Configuration, using custom (to set time synchronization) 61. *See also* Time synchronization
Configuring
NetWare Directory Services parameters with SET 147
NetWare server with INSTALL 139
Container objects. *See also* Objects; specific container object name or type
Country (C) 13, 85
explained 12
Locality (L) 13
Organization (O) 14
Organizational Unit (OU) 14
parent 12
placing, in large Directory tree 85
Container rights. *See also* Rights
ensuring Directory tree security with 101
granting 91, 101
Context
bindery (*see* Bindery context)
changing, caution 49
changing name, explained 26
defined 22
Directory tree full, length limitation 84
Conventions used in this manual vii
Country (C) container object. *See also* Container objects
designating 86
explained 13
using 85
Create object right, explained 18. *See also* Rights
Creating, for Directory tree
maps 76 (*see also* Directory tree, planning)
standards (*see* Information standards; Naming standards)
structure (*see* Directory tree structure)
Custom configuration, using to set time synchronization 61. *See also* Time synchronization

D

Database
Directory (*see* Directory database)
distributed, defined 34
Delete object right, explained 18. *See also* Rights
Delete Self property. *See* Add or Delete Self property

Departmental Directory tree, planning 79. *See also* Directory tree, planning
Directory
schema, explained 7 (*see also* Directory tree)
subtree, defined 34
synchronization, explained 38
Directory database
correcting, problems with DSREPAIR 135
managing, with PARTMGR 146
updating, information with UIMPORT 151
Directory fault tolerance, providing
with Primary time servers 58
with replicas 36, 95
with tape backups 96
Directory Map object. *See also* Objects
explained 170
naming standard suggestion 176 (*see also* Naming standards)
Directory objects. *See also* Objects
explained 8
naming standards (*see* Naming standards)
property information standards (*see* Information standards)
Directory partition replicas. *See* Replicas
Directory partitions. *See* Partitions
Directory services, explained
NetWare 4 (*see also* NetWare Directory Services)
standard 3
Directory tree
access, controlling 100 (*see also* Security)
departmental, planning 79 (*see also* Directory tree, planning)
examples (*see* Planning examples)
full context length limitation 84
hierarchical structure, explained 6
implementation strategy, developing 79 (*see also* NetWare Directory Services technology, implementing)
leaf objects, placing in 90 (*see also* Objects)
levels, planning 84 (*see also* Directory tree, planning)
managing, with NetWare utilities 131 (*see also* Managing)
maps, creating 76 (*see also* Directory tree, planning)
merging (*see* DSMERGE)
name requirement 84 (*see also* Naming)
naming standards, creating 75 (*see also* Naming standards)
planning (*see* Directory tree, planning)
restructuring (*see* Directory structure, changing)
schema, explained 7
security (*see* Security)
subtree, defined 34
Directory tree, changing. *See also* Directory tree, planning
considerations when using bindery services 105 (*see also* Bindery services)
restructuring, explained 80
Directory tree, planning. *See also* Directory tree structure; Planning examples
and NetWare Directory Services implementation (*see* NetWare Directory Services technology, implementing)
departmental 79
explained 71
large 82
levels 84
maps, creating for 76
naming standards, using for 75 (*see also* Naming standards)
organizational 82
using bindery services 83 (*see also* Bindery services)
Directory tree structure, creating (guidelines). *See also* Directory tree, planning
for large network 125
for medium network 120
for small network 116
if using bindery services 83
Disabling bindery services, explained 43. *See also* Bindery services
Distinguished Name
defined 22
maximum length allowed 78
Distributed database, defined 34. *See also* Directory database

Distribution List object, explained 172. *See also* Objects
Documentation conventions used in this manual vii
Drive mapping, using Directory Map object for 170. *See also* Directory Map object
DSMERGE utility
  functions in 132
  managing Directory tree with 132
  merging Directory trees with 80
DSREPAIR utility
  correcting Directory database problems with 135
  functions in 135
DSTRACE utility, monitoring NetWare Directory Services functions with 138

E

Effective rights, explained 21. *See also* Rights
Environment property information standards suggestions 178
Examples. *See* Planning examples
External Entity object, explained 172. *See also* Objects

F

Fault tolerance. *See* Directory fault tolerance

G

Glossary 183
Granting
  group rights 101 (*see also* Rights)
  group rights 91 (*see also* Rights)
  trustee assignments 100 (*see also* Security)
  trustee assignments 91 (*see also* Security)
Group object. *See also* Objects
  explained 168
  naming standard suggestion 176 (*see also* Naming standards)
  rights, ensuring Directory tree security with 101 (*see also* Rights)
Group rights, granting 101 (*see also* Rights)
Group rights, granting 91 (*see also* Rights)
Guidelines
  for copying replicas (*see* Replicas, copying)
  for creating partitions (*see* Partitions, creating)
  for granting rights 91 (*see also* Rights)
  for implementing NetWare Directory Services (*see* NetWare Directory Services technology, implementing)
  for merging Directory trees 80 (*see also* DSMERGE)
  for planning Directory tree (*see* Directory tree, planning)
  for setting up time services (*see* Time services, setting up)

H

Hierarchy, organizing Directory objects into logical 83. *See also* Directory tree structure

I

Identification Page information property standards suggestions 179, 181
Implementing NetWare Directory Services technology. *See* NetWare Directory Services technology, implementing
Information, inaccessible (when using bindery services with NetWare Directory Services) 44
Information standards, listed and explained
  document guidelines 175
  Organization object 181
  User object 178

Informational leaf objects, listed and explained 173. *See also* Objects
Inherited Rights Filter (IRF). *See also* Security
  ensuring Directory tree security with 101
  explained 17, 21
  planning 33
INSTALL utility
  installing, upgrading, or modifying NetWare with 139
  removing NetWare Directory Services with, caution 140
Installation, rights granted at. *See also* Rights
  to ADMIN 32
  to others 32
Integration strategy, developing for bindery services 103. *See also* Bindery services
IRF. *See* Inherited Rights Filter

L

Large network, implementing NetWare Directory Services on
  general guidelines 108
  specific guidelines 82, 122
Leaf objects, listed and explained. *See also* Objects
  general 14
  informational 173
  messaging-related 172
  miscellaneous 174
  printer-related 171
  server-related 170
  user-related 168
Leaf objects, placing in Directory tree 90. *See also* Objects
Levels, planning Directory tree 84. *See also* Directory tree, planning
Locality (L) container objects, explained 13. *See also* Container objects

M

Management utilities, listed and explained 39
Managing
  bindery services 104
  Directory database with PARTMGR 146
  Directory tree with NetWare utilities 131
  NetWare Directory Services 31
Managing Directory objects and properties. *See also* Object properties; Objects
  with NETADMIN 141
  with NetWare Administrator 144
  from different code pages 27
Maps, Directory tree (creating) 76. *See also* Directory tree, planning
Master replica, explained 37. *See also* Replicas
Medium network, implementing NetWare Directory Services on
  general guidelines 108
  specific guidelines 117
Merging Directory trees. *See* DSMERGE
Message Routing Group object, explained 168. *See also* Objects
Message Routing Group object, explained 173. *See also* Objects
Messaging Server object, explained 173. *See also* Objects
Messaging-related leaf objects, listed and explained 172. *See also* Objects
Miscellaneous leaf objects, listed and explained 174. *See also* Objects
Modifying. *See also* Changing
  AUTOEXEC.NCF to change system handling of time synchronization 150
  AUTOEXEC.NCF with INSTALL 148
  NET.CFG to complete tree merge 134
  NetWare server configuration with INSTALL 139
  TIMESYNC.CGF to change time synchronization server configuration 150
Moving bindery objects 105. *See also* Objects

N

Name
  context, changing 26 (*see also* Context)
  context, changing 78 (*see also* Context)
  Directory tree, requirements 84 (*see also* Directory tree)
  spaces, using 24
  types, explained 26 (*see also* specific name type)
Naming rules, listed
  for bindery services objects 28
  for international support 29
  for NetWare Server objects 28
  object, general 27 (*see also* Objects)
Naming scheme
  guidelines, discussed 24
  using 73
Naming standards
  consistency considerations 78
  creating 75
  Directory Map object suggestion 176
  document guidelines 175
  modifying 78
  name length considerations 78
  planning 77
  sample 176
  using 78
NCP Server object, explained 170. *See also* Objects
NDS. *See* NetWare Directory Services
NETADMIN utility
  functions in 142
  managing bindery services with 104
  managing Directory objects and properties with 141
NetWare Administrator utility
  explained 144
  managing bindery services with 104
  managing Directory objects and properties with 144
NetWare Directory Services (NDS)
  context, server (*see* Server context)
  database (*see* Directory database)
  overview 4
  standards document, developing 73 (*see also* Information standards; Naming standards)
NetWare Directory Services software
  installing, with INSTALL 139
  managing bindery services with 104
  removing, with INSTALL (caution) 140
  using NETX with 104
NetWare Directory Services technology
  defined 1
  explained 4
  features and benefits 4
  managing 31 (*see also* Managing)
NetWare Directory Services technology, advantages of using
  on any network 107
  on large network 122
  on medium network 117
  on small network 114
NetWare Directory Services technology, implementing (guidelines)
  on any network 108
  on large network 122
  on medium network 117
  overview, general 73
  overview, specific 107
  on small network 114
  strategy, developing 79
NetWare DOS Requester software, managing bindery services with 104. *See also* Bindery services; Managing
NetWare Server object naming rules, explained 28. *See also* Naming rules
NetWare shell. *See* NETX
Network time, determining common 57. *See also* Time
NETX, using with NetWare Directory Services 104

O

Object classes
  listed and explained 156
  properties, listed 158 (*see also* Object properties)
Object naming
  rules (*See* Naming rules)
  standards, sample 176 (*See also* Naming standards)
Object properties. *See also* specific object property name or type
  classes, listed 158
  explained 15
  information standards, sample 178 (*see also* Information standards)
  managing 141 (*see also* Managing Directory objects and properties)
Object rights, listed and explained 18. *See also* Rights
Objects. *See also* specific object name or type
  managing (*see* Managing Directory objects and properties)
  setting, information and naming standards (*see* Information standards; Naming standards)
  setting, name length 78 (*see also* Naming)
Optimizing time synchronization, explained 60. *See also* Time synchronization
Organization (O) container object. *See also* Container objects
  explained 14
  naming standard suggestion 176 (*see also* Naming standards)
  property information standards, sample 181 (*see also* Information standards)
  using, in large Directory tree 85
Organizational Directory tree, planning 82. *See also* Directory tree, planning
Organizational Role object. *See also* Objects
  explained 168
  naming standard suggestion 176 (*see also* Naming standards)
Organizational Unit (OU) container object. *See also* Container objects
  designating 87
  explained 14
  naming standard suggestion 176 (*see also* Naming standards)
  using, in large Directory tree 87

P

Parameters
  NetWare Directory Services, configuring with SET 147
  SET, using with TIMESYNC 149
Parent
  object, explained 12 (*see also* Objects)
  partition, defined 34 (*see also* Partitions)
Partial name. *See* Relative Distinguished Name
Partition replicas. *See* Replicas
Partition Root Entry, defined 34. *See also* Partitions
Partitioning, limited (when using bindery services with NetWare Directory Services) 44
Partitions, creating (guidelines)
  in large network 127
  in medium network 121
  in small network 116
Partitions. *See also* specific partition type
  explained 34
  parent and child, illustrated 34
PARTMGR utility
  explained 146
  functions in 146
  managing Directory database with 146
Performance, increasing system (with SET) 148
Planning Directory tree. *See* Directory tree, planning
Planning examples, Directory tree. *See also* Directory tree, planning
  medium-to-large 93
  small-to-medium 91
Planning maps, creating Directory tree 76. *See also* Directory tree, planning

Postal Address property information standards suggestions 181
Primary time server. *See also* Time servers
caution 64
explained 57
function 64
use 64
Print Queue object. *See also* Objects
explained 171
naming standard suggestion 177 (*see also* Naming standard)
Print Server object. *See also* Objects
explained 171
naming standard suggestion 177 (*see also* Naming standards)
Printer object. *See also* Objects
explained 172
naming standard suggestion 176 (*see also* Naming standards)
Printer-related leaf objects, listed and explained 171. *See also* Objects
Profile object. *See also* Objects
explained 168
naming standard suggestion 177 (*see also* Naming standards)
Properties, object. *See* Object properties
Property rights. *See also* Rights; specific property right name
explained 19
listed 19

R

RDN. *See* Relative Distinguished Name
Read property right, explained 20. *See also* Rights
Read/write replica, explained 37. *See also* Replicas
Read-only replica, explained 37. *See also* Replicas
Reference time server. *See also* Time servers
cautions 65
explained 58
function 65
use 65
Relative Distinguished Name (RDN), defined 22
Removing, caution
container from bindery context 49
NetWare Directory Services with INSTALL 140
Rename object right, explained 18. *See also* Rights
Replicas
decreasing WAN link traffic with 96
defined 36
distribution across WAN, example 97
list, defined 38
purpose 36
ring, defined 38
strategies, developing 95
types, listed and explained 37
Replicas, copying (guidelines)
on large network 128
on medium network 121
on small network 117
Replication strategy, developing 95
Restructuring Directory tree. *See* Directory tree, changing
Rights. *See also* Security; specific right type
categories, listed and explained 17
granted at installation, to ADMIN 32
granted at installation, to others 32
guidelines for granting 91
[Root] object, explained 11. *See also* Objects
Root partition, defined 34. *See also* Partitions

S

SAP. *See* Service Advertising Protocol
Schema, Directory 7. *See also* Directory tree
Secondary time server. *See also* Time servers
cautions 66
explained 60
function 66
use 66
Security Equal To property. *See also* Properties
ensuring Directory tree security with 102
explained 21

Security. *See also* Directory fault tolerance; Properties; Rights; Time synchronization
Access Control List, explained 20
access to Directory tree, controlling 100
authentication, explained 25
container rights, ensuring Directory tree security with 101
Directory tree, developing strategy for 100
effective rights, explained 21
group object rights, ensuring Directory tree security with 101
Inherited Rights Filter, explained 21 (*see also* Inherited Rights Filter)
Security Equal To property, ensuring Direectory tree security with 102
Security Equal To property, explained 21
security equivalency, explained 102
trustee assignments, ensuring Directory tree security with 100
Server
configuration, modifying NetWare (with INSTALL) 139
context, explained 139 (*see also* Context)
object, naming standard suggestion 177 (*see also* Naming standards; Objects)
Server-related leaf objects, listed and explained 170. *See also* Objects
Service Advertising Protocol (SAP), setting time synchronization with 61. *See also* Time synchronization
SET
command, using to set up bindery context 43
parameters, using with TIMESYNC 149 (*see also* Time synchronization)
SET utility
configuring NetWare Directory Services parameters with 147
increasing system performance with 147
setting multiple bindery contexts with 52
Setting up
bindery services, considerations 43 (*see also* Bindery services)
time services (*see* Time services)
Single Reference time server. *See also* Time servers
cautions 63
explained 55
function 63
use 63
Site locations, designating in Directory tree 86
Small network, implementing NetWare Directory Services on
general guidelines 108
specific guidelines 114
Spaces, using in names 24
Special characters, using in names 27
Standards. *See* Information standards; Naming standards
Strategies, developing
bindery services integration 103
replication 95
time synchronization 99
Structure, Directory tree. *See* Directory tree structure
Subordinate reference replica, explained 38. *See also* Replicas
Subtree, Directory (defined) 34
Supervisor right, explained. *See also* Rights
object 18
property 20
Synchronization
Directory, explained 38
time (*see* Time synchronization)
System performance, increasing with SET 148

T

Time
determining common network 57
source server, defined 60
stamps, explained 54
Time servers. *See also* specific time server type
designating, during NetWare 4.1 installation 55
explained 55
using custom configuration for 61

Time services, setting up (guidelines)
for large network 126
for medium network 121
for small network 116
Time synchronization. *See also* TIMESYNC
explained 53
method, choosing 62
monitoring, with TIMESYNC 149
optimizing 60
strategy, developing 99
TIMESYNC utility, explained. *See also* Time synchronization
monitoring server synchronization with 149
using SET parameters with 149
Tree. *See* Directory tree
Trustee assignments
ensuring Directory tree security with 100
granting 100
Typeful name type, explained 26
Typeless name type, explained 26

U

UIMPORT utility
explained 151
updating Directory database information with 151
Unicode characters, using in names 27, 29
Unknown object, explained 174. *See also* Objects
Upgrading NetWare with INSTALL 139
User object. *See also* Objects
ADMIN, explained 32
explained 169
naming standard suggestion 177 (*see also* Naming standards)
property information standards, sample 178 (*see also* Information standards)
User-related leaf objects, listed and explained 168. *See also* Objects
Utilities, using to manage Directory tree (listed and explained) 131. *See also* specific utility name or type

V

Volume object, explained 171. *See also* Objects

W

WAN link
traffic, decreasing with replicas 96
using partitions for faster access across 37
Write property right, explained 20. *See also* Rights

User Comments

We want to hear your comments and suggestions about this manual. Please send them to the following address:

Novell, Inc.
Technical Publications
MS C-23-1
122 East 1700 South
Provo, UT 84606
U.S.A

Fax: (801) 429-3002

NetWare 4.1
Introduction to NetWare Directory Services
Part #100-002069-001
December 1994

For technical support issues, contact your local dealer.

Your name and title: _____

Company: _____

Address: _____

_____

_____

Phone number: _____ Fax: _____

I use this manual as ❑ an overview ❑ a tutorial ❑ a reference ❑ a guide ❑ _____

	Excellent	Good	Fair	Poor
Completeness	❑	❑	❑	❑
Readability (style)	❑	❑	❑	❑
Organization/Format	❑	❑	❑	❑
Accuracy	❑	❑	❑	❑
Examples	❑	❑	❑	❑
Illustrations	❑	❑	❑	❑
Usefulness	❑	❑	❑	❑

Please explain any of your ratings: _____

_____

_____

_____

_____

In what ways can this manual be improved? _____

_____

_____

_____

_____

You may photocopy this comment page as needed so that others can also send in comments.