

**HP 9000 Networking**  
**NetWare® 4.1/9000**  
**Concepts**

**HP Part No. J2771-90012**  
**Printed in U.S.A. 12/96**

**Edition 1**



## Notice

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

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

This document contains proprietary information, which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated into another language without the prior written consent of Hewlett-Packard Company. The information contained in this document is subject to change without notice.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited. Microsoft<sup>®</sup>, MS<sup>®</sup>, and MS-DOS<sup>®</sup> are registered trademarks, and Windows is a trademark of Microsoft Corporation. NetWare, and Novell are registered trademarks of Novell, Inc.

© Copyright 1996, Hewlett-Packard Company

### **Restricted Rights Legend**

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

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

---

## Printing History

The manual printing date and part number indicate its current edition. The printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date. The manual part number will change when extensive changes are made.

Manual updates may be issued between editions to correct errors or document product changes. To ensure that you receive the updated or new editions, you should subscribe to the appropriate product support service. See your HP sales representative for details.

First Edition: December, 1996

## How to Use This Manual

### Introduction

Concepts is an extended glossary that includes terms related to the NetWare, network operating system and to networking in general. Use this manual as a reference if you have questions during the installation and operation of your network.

Concepts are arranged alphabetically. Some entries contain “See” or “See also” references to other entries where concepts are explained in detail. Some entries refer you to related information in other manuals.

### Documentation Conventions

This manual uses the following Hewlett-Packard 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.

#### 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.

#### 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.

HP-UX commands and command option letters are shown in bold monospace font. For example, **ls -l**. Because HP-UX commands are case sensitive, type them exactly as shown.

#### 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.

HP-UX filenames, directory names, and pathnames are shown in bold monospace font. For example: NWInodes. Because HP-UX is case-sensitive, type these names exactly as shown.

#### 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

## How to Use This Manual

“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).

### NET.CFG File Section Headings and Parameter Settings

NET.CFG section headings and parameter settings are shown in uppercase when used as a reference item and lower case when used in syntax or working examples.

For example:

NETBIOS VERIFY TIMEOUT specifies how often in (ticks) NetBIOS sends a keep-alive packet to the other side of a session to preserve the session.

If no packets are being exchanged on the NetBIOS session by the software that established the session, NetBIOS sends packets at regular intervals to make sure that the session is still valid.

<b>Syntax</b>	netbios verify timeout <i>number</i> Replace <i>number</i> with a number of ticks.
<b>Default</b>	54 (approximately 3 seconds)
<b>Range</b>	4 to 65,535

<b>Example</b>	<p>To make NetBIOS wait longer before sending a request-for-acknowledgment packet, you could place the following lines in your NET.CFG file:</p> <pre>netbios netbios verify timeout 1350</pre>
----------------	---

Because interpretation of this file is not case-sensitive, you can type its contents in uppercase or lowercase letters.

### 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 the computer on your network in place of *remote\_host*.

## How to Use This Manual



---

# Contents

## Chapter 1 NetWare Glossary 1-1

### A 1-2

- Access Control List (ACL) 1-2
- Access Control right 1-2
- Access privileges 1-3
- Accounting 1-3
- ACL 1-5
- Add or Delete Self right 1-5
- Add-on board 1-5
- Address 1-5
- Address Resolution Protocol (ARP) 1-6
- ADMIN object 1-6
- Alias object 1-7
- Application 1-7
- Archive 1-8
- Archive Needed attribute 1-8
- ARP 1-8
- Attaching 1-8
- Attributes 1-9
- Authentication 1-12
- AUTOEXEC.BAT 1-13

### B 1-14

- Backup 1-14
- Bindery 1-14
- Bindery emulation 1-15
- Bindery object 1-15
- Bindery Queue object 1-15
- Bindery services 1-15
- Binding and unbinding 1-18
- Boot files 1-19
- BOOTCONF.SYS 1-19
- Bridge 1-20
- Browse right 1-20

---

## Contents

Browsing	1-20
Buffer	1-21
<b>C</b>	<b>1-22</b>
Cabling system	1-22
Cache memory	1-22
Can't Compress attribute	1-22
Client	1-22
COM port	1-23
Command format	1-23
Common name	1-23
Communication	1-23
Communication protocol	1-24
Compare right	1-24
Complete name	1-25
Compressed attribute	1-25
Computer object	1-25
Configuration	1-25
Connection number	1-26
Connectivity	1-26
Console	1-26
Container object	1-26
Context	1-26
Country object	1-28
Create right	1-28
<b>D</b>	<b>1-29</b>
Daemon	1-29
Default drive	1-29
Default server	1-29
Delete Inhibit attribute	1-29
Delete right	1-30
Delimiter	1-30
Device sharing	1-30
Director/directory	1-30

---

## Contents

Directory and file rights	1-30
Directory management request	1-31
Directory Map object	1-31
Directory path	1-32
Directory rights	1-32
Directory Services	1-32
Directory Services daemons	1-32
Directory services request	1-33
Directory structure, file system	1-33
Directory structure, NetWare Directory Services	1-40
Directory tree	1-40
Disk	1-41
Disk format	1-41
Don't Compress attribute	1-41
Don't Migrate attribute	1-41
DOS boot record	1-42
DOS client	1-42
DOS filenames	1-42
DOS Requester	1-42
DOS setup routine	1-43
DOS version	1-43
Drive	1-43
Drive mapping	1-44
Dynamic memory	1-45
<b>E</b>	<b>1-47</b>
Effective rights	1-47
Engine	1-49
Erase right	1-49
Ethernet configuration	1-49
Execute Only attribute	1-53
Extended attribute	1-53
<b>F</b>	<b>1-55</b>
Fake root	1-55

---

## Contents

FAT	1-55
File Allocation Table (FAT)	1-55
File caching	1-56
File compression	1-56
File handle	1-56
File locking	1-56
File rights	1-56
File Scan right	1-57
File sharing	1-57
File Systems, NetWare Services	1-57
HP-UX Partitions and NetWare Volumes	1-58
File system, HP-UX	1-59
File Transfer Protocol (FTP)	1-59
Filename extensions, NetWare	1-59
Flag	1-60
Form	1-60
Frame	1-61
FTP	1-61
<b>G</b>	<b>1-62</b>
Gateway	1-62
Group object	1-62
<b>H</b>	<b>1-63</b>
Handle	1-63
Hard disk	1-63
HCSS	1-63
Hexadecimal	1-63
Hidden attribute	1-63
High Capacity Storage System (HCSS)	1-64
High Performance File System (HPFS)	1-64
Home directory	1-64
Hop count	1-64
HPFS	1-65
Hybrid user	1-65

---

## Contents

### I 1-67

- Identifier variables 1-67
- Immediate Compress attribute 1-67
- Indexed attribute 1-67
- Inherited Rights Filter, file system 1-67
- Inherited Rights Filter, NDS object 1-68
- Internal network number 1-70
- Internet Protocol (IP) 1-70
- Internetwork 1-71
- Internetwork Packet eXchange (IPX) 1-71
- Internetwork Packet eXchange Open Data-Link Interface (IPXODI) 1-71
- Interoperability 1-72
- Interprocess communication (IPC) 1-72
- IPX 1-73
- IPX external network number 1-73
- IPX internal network number 1-73
- IPX internetwork address 1-74
- IPXODI 1-74

### J 1-75

- Jumper block 1-75

### L 1-76

- LAN 1-76
- LAN driver, client 1-76
- Large Internet Packet (LIP) 1-76
- How LIP works 1-77
- Leaf objects 1-77
- Link Support Layer (LSL) 1-77
- LIP 1-78
- Loadable module 1-78
- Local Area Network (LAN) 1-78
- Lock manager 1-78
- Logical memory 1-78

---

## Contents

Login 1-78  
LOGIN directory 1-79  
Login restrictions 1-79  
Login scripts 1-80  
Logout 1-93  
Long filename 1-93  
Long machine type 1-93  
LPT1 1-94  
LSL (Link Support Layer) 1-94

### M 1-95

MAIL directory 1-95  
Map 1-95  
Memory, DOS management 1-95  
Memory allocation 1-96  
Memory board 1-96  
Message packet 1-96  
Message system 1-96  
Migrated attribute 1-97  
MLID 1-97  
Modify bit 1-97  
Modify right 1-97  
Multiple-byte character 1-97  
Multiple Link Interface Driver (MLID) 1-97  
Multiple name space support 1-98  
Multiserver network 1-99

### N 1-100

Name context 1-100  
Name space 1-100  
NCP 1-100  
ncp\_engine 1-100  
NCP Packet Signature 1-101  
NDS 1-101  
NEMUX 1-101

---

## Contents

NETBIOS.EXE	1-101
NET.CFG	1-102
NetWare daemon	1-102
NetWare Core Protocol (NCP)	1-103
NetWare Diagnostic Daemon (NWDIAGD)	1-103
NetWare Directory database	1-104
NetWare Directory Services (NDS)	1-104
NetWare inode	1-116
NetWare Loadable Module (NLM)	1-117
NetWare managed node	1-117
NetWare management agents	1-117
NetWare operating system	1-118
NetWare Protocol stack daemon (NPSD)	1-118
NetWare Server object	1-119
Network	1-120
Network address	1-120
Network backbone	1-120
Network board	1-121
Network communication	1-121
Network File System (NFS)	1-121
Network Interface Card (NIC)	1-122
Network management	1-122
Network node	1-123
Network number	1-123
Network numbering	1-123
Network printer	1-124
Network supervisor	1-124
NETX	1-124
NFS	1-124
NIC	1-125
NLM	1-125
Node address	1-125
Node number	1-125
Normal attribute	1-125

---

## Contents

Novell Virtual Terminal (NVT)	1-126
Novell Virtual Terminal 2 (NVT2)	1-126
NPS daemon	1-126
NSE	1-126
NVT	1-126
NVT2	1-126
nwcm	1-127
nwconfig	1-127
NWDIAGD	1-128
<b>O</b>	<b>1-129</b>
Object	1-129
Object rights	1-138
ODI	1-138
ODINSUP	1-138
Open Data-Link Interface (ODI)	1-138
Open Data-Link Interface Network driver interface specification SUPport (ODINSUP)	1-141
Organization object	1-143
Organizational Role object	1-143
Organizational Unit object	1-143
<b>P</b>	<b>1-144</b>
Packet	1-144
Packet Burst protocol	1-144
Parallel port	1-146
Parent directory	1-146
Parent objects	1-146
Parity	1-146
Partition, Directory Services	1-147
Partition, file system	1-148
Partition management	1-148
Password	1-149
Path	1-149
Permissions	1-149



---

## Contents

Port, hardware	1-150
Port, software	1-150
Power conditioning	1-150
Primary time server	1-151
Print device	1-151
Print device mode	1-152
Print job	1-152
Print job configuration	1-153
Print queue	1-153
Print queue name	1-154
Additional information	1-154
Print queue operator	1-155
Print queue polling time	1-155
Print server	1-155
Print Server object	1-156
Print Server Operator	1-156
Printer	1-156
Printer definition	1-157
Printer form	1-158
Printer object	1-158
Printing	1-158
Process	1-159
Profile login script	1-160
Profile object	1-160
Prompt	1-160
Property	1-161
Property rights	1-162
Protected mode	1-162
Protocol	1-162
Protocol stack	1-163
PUBLIC directory	1-163
Public files	1-163
Public trustee	1-163
Purge attribute	1-164

---

## Contents

- Q 1-165
  - Queue 1-165
  - Queue polling time 1-165
- R 1-166
  - RAM 1-166
  - Read-ahead cache 1-166
  - Read Only attribute 1-166
  - Read right 1-166
  - Real mode 1-166
  - Record locking 1-167
  - Recursive copying 1-168
  - Reference time server 1-168
  - Registered resources 1-168
  - Remote administration 1-169
  - Remote boot 1-169
  - Remote connection 1-169
  - Remote Reset 1-169
  - Remote workstation 1-170
  - Rename Inhibit attribute 1-170
  - Rename right 1-171
  - Replica 1-171
  - Resources 1-172
  - Restore 1-173
  - Rights 1-173
  - RIP 1-178
  - Root directory 1-178
  - Root file system 1-178
  - Root object 1-178
  - Root user 1-179
  - Router 1-179
  - Router Information Protocol (RIP) 1-180
- S 1-182
  - Salvageable files 1-182

---

## Contents

SAP 1-182  
SAP daemon 1-182  
SCSI 1-182  
SCSI bus 1-182  
Search drive 1-182  
Search modes 1-183  
Secondary time server 1-184  
Security 1-184  
Security equivalence 1-190  
Semaphore 1-192  
Sequenced Packet Exchange (SPX) 1-192  
Sequenced Packet eXchange II (SPXII) 1-192  
Serial port 1-193  
Server console 1-193  
Server protocol 1-194  
Service Advertising Protocol (SAP) 1-194  
Service Advertising Protocol daemon (SAPD) 1-195  
Shareable attribute 1-195  
Shared memory 1-196  
Short machine type 1-196  
Single Reference time server 1-196  
Small Computer Systems Interface (SCSI) 1-197  
Socket 1-197  
SPX 1-198  
SPX2 1-198  
SPXII 1-198  
Station 1-198  
Station address 1-198  
STREAMS 1-198  
Subdirectory 1-198  
Supervisor right 1-199  
Switch block 1-199  
Synchronization 1-199  
Synchronization services 1-199

---

## Contents

System attribute	1-201
SYSTEM directory	1-201
System login script	1-201
<b>T</b>	<b>1-202</b>
Tape backup unit	1-202
TCP/IP	1-202
Terminal emulation software	1-202
Time synchronization	1-202
Topology	1-208
Transmission Control Protocol/Internet Protocol (TCP/IP)	1-209
Transaction Tracking System (TTS)	1-209
Transactional attribute	1-209
Transmission Control Protocol (TCP)	1-209
Trustee	1-209
Trustee database	1-212
TTS	1-212
<b>U</b>	<b>1-213</b>
Unbinding	1-213
Unknown object	1-213
Unicode	1-213
UNIX client	1-214
UNIX host locking	1-214
UNIX operating system	1-215
HP-UX	1-216
User login script	1-216
User object	1-216
User template	1-220
Utilities	1-220
<b>V</b>	<b>1-222</b>
Value-added server	1-222
Virtual Loadable Module (VLM)	1-222
VLM	1-223
voltab	1-223

---

## Contents

Volume 1-223

Volume object 1-224

### W 1-226

Wait state 1-226

WAN 1-226

Watchdog 1-226

Wide Area Network (WAN) 1-226

Windows client 1-227

Workstation 1-227

Write right 1-227

---

## Contents

---

**NetWare Glossary**

## A

### **Access Control List (ACL)**

A property of NetWare Directory Services (NDS) objects that controls how other objects can access the object.

An ACL contains trustee assignments that include NDS object and property rights. The ACL also contains the Inherited Rights Filter. When you view an object's trustees or Inherited Rights Filter, you are seeing the values of that object's ACL.

An ACL for NDS objects is like the list of trustees for a file or directory.

To change an ACL (and therefore a trustee's rights to an object), you must have a property right that allows you to modify the ACL for that NDS object.

“Security.” See also “Object”;

### **Access Control right**

A NetWare right that allows users to modify trustee assignments and Inherited Rights Filters to a file or directory.

At the directory level, this right allows users to modify both directory and file trustee assignments and Inherited Rights Filters. Users with Access Control right can grant any right to any other user, including rights that they themselves have not been granted. They can, however, only grant Supervisor rights if they themselves have Supervisor rights.

If more than one user has access to the directory, the Access Control right allows a user to modify the restrictions.

At the file level, this right allows users to modify only the file's trustee assignments and Inherited Rights Filter. Users who have this right can grant any right to any other user, including rights that they themselves have not been granted. They can, however, only grant Supervisor rights if they themselves have Supervisor rights.



See also “Inherited Rights Filter, file system”; “Inherited Rights Filter, NDS object”; “Rights”; “Trustee.”

### **Access privileges**

Any of the directory or file rights that control access to files or directories. Types of access privileges differ for HP-UX users, NetWare users, and hybrid users.

See also “HPFS”; “Rights”; “Security.”

### **Accounting**

The process of tracking resources used on a network so that the network supervisor can charge for network services and resource usage. This is done by assigning account balances to users to draw from as they use the services and resources.

#### **Charging for network services and resources**

The network supervisor can assign different charge rates for services at different times of the day, in half-hour increments.

The network supervisor can charge for

- Blocks read. The charge amount is for each block of data read from the server.
- Blocks written. The charge amount is for each block of data written to the server.
- Connect time. The charge amount is for each minute a user is logged in. User logins and logouts are tracked automatically.
- Service requests. The charge amount is for each request to the server.

To calculate the charge rate for services, the network supervisor should

- Determine network costs and the amount to charge over a given period of time.
- Decide which services to charge for (based on network costs) and determine the amount that needs to be made from each service.

For example, if server disk storage capacity is a concern, then charge for disk storage. If network use is high, charge for service requests. To encourage users to log out when they aren't working, charge for connect time.

- Estimate how much of each service is being used by monitoring the server for two

**A**

or three weeks.

For example, if 30% of the server's charges are from service requests, the network supervisor would want to recoup 30% of the cost through charging for service requests.

At the end of the monitoring period, use *ATOTAL* to determine the total use for each service. (*ATOTAL* is located in *SYS:SYSTEM* and requires the Supervisor right.)

After establishing how much each service is used and the amount needed to be made from each service, the network supervisor can calculate a charge rate.

The charge rate is the charge per unit of the specified service. This rate converts the amount of service used to a monetary figure. The unit is arbitrary, but consider beginning with one charge unit equaling one cent.

Use the following formula to calculate a charge rate:

$$\frac{\text{CHARGE (charge rate multiplier)}}{\text{ESTIMATED USAGE (charge rate divider)}} = \text{CHARGE RATE}$$

**Figure 1-1**

**Charge rate formula**

For example, a network supervisor needs to charge \$100 per month for blocks read. There are 250,000 blocks read each month. Therefore, the charge rate is \$100 (or 10,000 cents) divided by 250,000 blocks, or \$.01 (1 cent) per 25 blocks:

$$\frac{10,000 \text{ cents}}{250,000 \text{ blocks read}} = 1 \text{ cent} / 25 \text{ blocks read}$$

**Figure 1-2**

**Sample charge rate**

**Assigning account balances**

The network supervisor can

- Assign each user a balance to control how much service the user can use
- Assign a credit limit (or allow unlimited credit)
- Assign a default account balance for all users
- Increase a user's account balance

The user must log out and log in again before changes take effect.

Related utilities: "AUTOTAL"; "NETADMIN"; "NetWare Administrator" (*Utilities Reference*).

**ACL**

See "Access Control List (ACL)."

**Add or Delete Self right**

A property right that grants a trustee the right to add or remove itself as a value of the property.

See also: "Rights."

**Add-on board**

A circuit board that modifies or enhances a personal computer's capabilities. Examples:

- Memory board. Increases the amount of RAM within a personal computer.
- Network board. Installed in each network station to allow stations to communicate with each other and with the NetWare server.

**Address**

A number that identifies a location in memory or disk storage, or that identifies the location of a device on the network.

See also: "Address Resolution Protocol (ARP)"; "IPX external network number"; "SCSI bus."

### **Address Resolution Protocol (ARP)**

A process in Internet Protocol (IP) and AppleTalk networks that allows a host to find the physical address of a target host on the same physical network when it only knows the target's logical address.

Under ARP, a network board contains a table that maps IP addresses to the hardware addresses of the objects on the network.

To create entries, the ARP broadcasts a request with the target's IP address. The target responds with its physical address. The network board adds the physical address to its ARP table and can then send packets to the target.

### **ADMIN object**

A User object, created automatically during Directory Services installation, that has rights to create and manage objects.

When you first create the Directory tree, ADMIN is given a trustee assignment to the root object. This trustee assignment includes the Supervisor object right, which means that ADMIN has rights to create and manage all objects in the tree.

As you create other User objects in the Directory tree, you can give them the Supervisor object right to create and manage other container objects and all their leaf objects. Control of the network is as dispersed or centralized as you make it.

After you assign the Supervisor object right to other User objects, you can rename ADMIN or delete it.

---

**CAUTION:**

---

Do not delete ADMIN until you've created other User objects and granted someone the Supervisor object right to the root object. Otherwise, no one will have full access to the tree.

ADMIN does not have any special significance like SUPERVISOR did in earlier versions of NetWare. It is only the first User object created and therefore must have the ability to create other objects.

## Alias object

A leaf object that points to the original location of an object in the Directory. Aliases can make NDS easier to use.

Any object located in one place in the Directory can also appear to be in another place in the Directory by using aliases.

For example, an administrator could create aliases pointing to all modems on the network. The aliases could all be created in one container. A user would then need to search only one area of the Directory tree to find out about all modems on the network.

When you create an Alias Object, name it in a way that you can recognize it as an Alias. The name is the only way you can recognize it as an Alias once it is created.

When you add aliases to a list—for example, to add an alias of a user to a group, the name of the User object appears in the list, not the alias that points to the user.

To access the alias and the properties of the object it refers to, you need the Read right to the alias name and the Read right to the properties of the object it refers to.

You can set an option in NETADMIN to see the alias as a reference to another object, and then assign rights or modify the properties of the actual object.

If you don't set the option to see aliases as references, you can't work with the properties or rights of the object the alias refers to.

See also: "Creating Leaf Objects"; "Cautions When Deleting Alias Objects" (*Supervising the Network*); "Object."

## Application

A software program that makes calls to the operating system and manipulates data files, allowing a user to perform a specific job (such as accounting or word processing).

- Stand-alone application. An application that runs from a self-contained,

## A

independent computer. Only one user can access the application.

- Network application. An application that runs on networked computers and can be shared by users.

Network applications use network resources such as printers. Advanced network applications (such as electronic mail) allow communication among network users.

### **Archive**

A transfer of files to long-term storage media, such as optical disks or magnetic tape.

See also “Attributes”; “Backup.”

### **Archive Needed attribute**

A file attribute, set by NetWare, indicating that the file has been changed since the last time it was backed up.

See also: “Attributes.”

### **ARP**

See “Address Resolution Protocol (ARP).”

### **Attaching**

Establishing a connection between a workstation and a NetWare server. The server assigns each station a connection number and attaches each station to its LOGIN directory.

When a user runs the Netware DOS Requester, the NetWare shell attaches the station to the server that responds first, or to the server specified in the station’s NET.CFG file as the preferred server.

Related utilities: “LOGIN”; “LOGOUT”; “MAP” (*Utilities Reference*).

## Attributes

The characteristics of a directory or file that tell NetWare what to do with the directory or file; also called flags.

Attributes cannot be assigned to NetWare Directory Services objects.

NetWare reads the attributes the user sets (for example, Delete Inhibit) and sets other attributes to show what has been done (for example, Archive Needed).

Effective rights cannot override attributes. Attributes can be changed, however, by a user who has been granted at least the Modify right (see “Rights”).

DOS attributes work like attributes of the same name in DOS, but apply to files and directories stored on NetWare volumes.

NetWare attributes are unique to NetWare. They apply only to files and directories stored on NetWare volumes. They do not affect HP-UX system operations and are not visible from the HP-UX system. NetWare attributes are visible only to logged-in NetWare users.

In NetWare Services, NetWare file and directory attributes do not match HP-UX system permissions. NetWare stores attributes for each file or directory in the user inode file in each volume. The information includes the attributes, creation time, and the creator of the file. See “NetWare inode.”

NetWare text utilities display the initial letters of these attributes between brackets (see Table 1). In NetWare graphical utilities and some text utilities, the full name of the attribute is used.

Related utilities: “FLAG,” “FILER,” or “NetWare Administrator” (*Utilities Reference*).

**Table 1-1**                      **Directory Attributes**

Attribute	Description
Delete Inhibit [Di]	Prevents any user from erasing the directory
Don't Compress [Dc]	Prevents directories from being compressed. Not supported in NetWare Services.

**Table 1-1 Directory Attributes**

<b>Attribute</b>	<b>Description</b>
Don't Migrate [Dm]	Prevents directories from being migrated to a secondary storage device. Not supported in NetWare Services.
Hidden [H]	Hides the directory from the DOS DIR command and prevents it from being deleted or copied. This is a DOS attribute.  However, the NetWare NDIR command shows the directory if the user has the File Scan right (see "Rights").
Immediate Compress [Ic]	Causes the directory to be compressed as soon as the operating system can do so. Not supported in NetWare Services.
Normal [N]	Sets the directory to no attributes.
Purge [P]	Causes NetWare to purge the directory when it is deleted. Not supported in NetWare Services.
Rename Inhibit [Ri]	Prevents any user from renaming the directory.
System [Sy]	Prevents DOS directories used only by the operating system from being deleted or copied. The directories are hidden from the DOS DIR command. However, the NetWare NDIR command shows the directory if the user has the File Scan right (see "Rights")

**Table 1-2 File Attributes**

<b>Attribute</b>	<b>Description</b>
Archive Needed [A]	Indicates that the file has been changed since the last time it was backed up. NetWare sets this attribute when a file is modified. Backup programs usually clear this attribute after backing up the file.
Can't Compress [Cc]	Indicates that the file cannot be compressed because of insignificant space savings. Not supported in NetWare Services.
Compressed [Co]	Indicates that the file is compressed. Not supported in NetWare Services.
Delete Inhibit [Di]	Prevents any user from erasing the file.
Don't Compress [Dc]	Prevents the file from being compressed. Not supported in NetWare Services.



**Table 1-2 File Attributes**

Attribute	Description
Don't Migrate [Dm]	Prevents the file from being migrated to a secondary storage device. Not supported in NetWare Services.
Execute Only [X]	Prevents a file from being copied. Only the network supervisor can set this file attribute: it cannot be cleared. It should be set only if you have a second copy of the file.  Backup utilities do not back up the file, and some program files with this attribute set don't execute properly.
Hidden [H]	Hides the file from the DOS DIR command and prevents it from being deleted or copied. However, the NetWare NDIR command shows the file if the user has the File Scan right (see "Rights").
Immediate Compress [Ic]	Causes the file to be compressed as soon as the operating system can do so.
Indexed [I]	A status flag set by NetWare when a file exceeds a set size. Indicates that the file is indexed for fast access. This attribute is shown on attribute lists, but cannot be set by the user. Not supported in NetWare Services.
Migrated [M]	Indicates that the file has been migrated. Not supported in NetWare Services.
Normal [N]	Clears all NetWare attributes.
Purge [P]	Causes NetWare to purge the file when it is deleted. Not supported in NetWare Services.
Read Only [Ro]	Indicates that no one can write to this file. When Read Only is set or cleared, NetWare also sets or clears the Delete Inhibit and Rename Inhibit attributes. Consequently, a user cannot write to, erase, or rename a file when Read Only is set.  A user with the Modify right can remove the Delete Inhibit and Rename Inhibit attributes without removing Read Only. Then the file can be deleted or renamed, but not written to (see "Rights").  NetWare shows Read Write [Rw] if Read Only is not set.
Rename Inhibit [R]	Prevents a user from renaming a file.

**Table 1-2 File Attributes**

Attribute	Description
Shareable [S]	Allows the file to be accessed by more than one user at a time. Usually used in combination with the Read Only attribute.
System [Sy]	Marks DOS files that are used only by the operating system. The files are hidden from the DOS DIR command and cannot be deleted, renamed, or copied. However, the NetWare NDIR command shows the file if the user has the right to see it (see "Rights").
Transactional [T]	Indicates that the file is protected by Transaction Tracking System (TTS). Not supported in NetWare Services.

See also: "Extended attribute."

### Authentication

A means of verifying that an object sending messages or requests to NetWare Directory Services is authorized to do so.

Authentication guarantees that only the apparent sender could have sent a message or request, and that it originated from the workstation where the authentication data was created.

Authentication works with login restrictions and access control rights to provide a secure network.

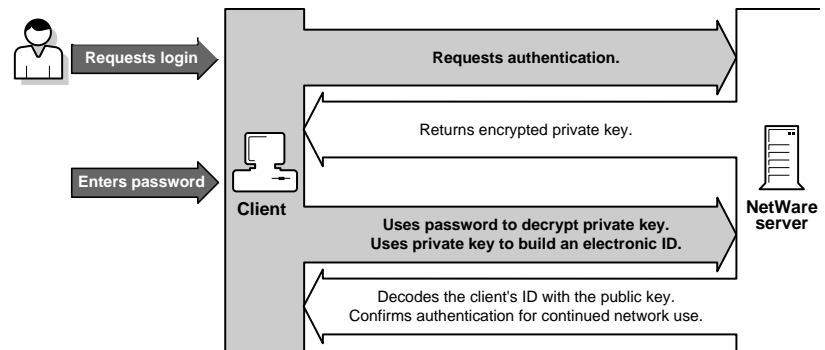
To a user, the only visible sign of authentication is a request for a password during network login. Every subsequent network operation is transparently authenticated using identification information created when the password was entered.

NetWare authentication uses a Public Key Encryption system that is virtually unbreakable. It consists of a private key and a public key. The keys are strings of numbers used in complex mathematical functions.

The workstation uses a private key to encode messages sent to the NetWare server. The server then uses a public key to decode the messages. The server knows that the workstation sent it, because the workstation's private key is required to encode the message.

Neither the two keys nor the user's password are ever sent across the network.

Authentication is illustrated in the following figure.



**Figure 1-3**

### Authentication

Keys are not changed during a login session, but periodically logging out and logging in again does change the keys. Forcing periodic logouts by setting login time restrictions increases network security.

### AUTOEXEC.BAT

A batch file that executes automatically when DOS is booted on a computer.

A workstation's AUTOEXEC.BAT file, located on the bootable floppy or hard disk, can contain commands that

- Load NetWare client files.
- Load other files required by the hardware.
- Set the DOS prompt.
- Change the default drive to the first network drive.
- Log the user in.

The workstation AUTOEXEC.BAT file can also load user-specific programs such as NETBIOS.COM, or it can call other batch files.

**B**

**Backup**

A duplicate of data (file, directory, volume), copied to a storage device (floppy diskette, cartridge tape, hard disk). A backup can be retrieved and restored if the original is corrupted or destroyed.

Related utilities: “ndsrestore”; “ndsbackup” (*Utilities Reference*).

See also: “Backing Up and Restoring the NetWare Directory Services” (*Supervising the Network*).

**Bindery**

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

In NetWare 4, the bindery has been replaced by the NetWare Directory database, under NetWare Directory Services.

Bindery services provides NetWare 4 networks with backward compatibility to NetWare versions that used the bindery. (See “Bindery services.”)

The following table compares features of bindery- and Directory-based versions of NetWare.

**Table 1-3 Comparison of bindery and Directory environments**

<b>Feature</b>	<b>Bindery</b>	<b>Directory</b>
Logical Structure	Flat structure	Hierarchical tree
Partitions	None	Distributed database
Replication	None	Partitions replicated
Synchronization	No replicas	Replicas synchronized
Users	Separate account on each server	Global account for network
Groups	Server-by-server	Network-wide

**Table 1-3 Comparison of bindery and Directory environments**

<b>Feature</b>	<b>Bindery</b>	<b>Directory</b>
Login	Password per server	System-wide with authentication
Printing	No friendly map	User-friendly
Volumes	Server-specific	Global objects
Queues	Local objects only	System-wide objects
Trustees	Server-specific	Global objects

**Bindery emulation**

See “Bindery services.”

**Bindery object**

A leaf object that represents an object placed in the Directory tree by an upgrade utility, but that NetWare Directory Services cannot identify as a genuine Directory Services object.

This object is for backward compatibility with bindery-oriented utilities.

See also: “Object.”

**Bindery Queue object**

A leaf object that represents a queue placed in the Directory tree by an upgrade or migration utility, but that NetWare Directory Services cannot identify.

This object is for backward compatibility with bindery-oriented utilities.

See also: “Object.”

**Bindery services**

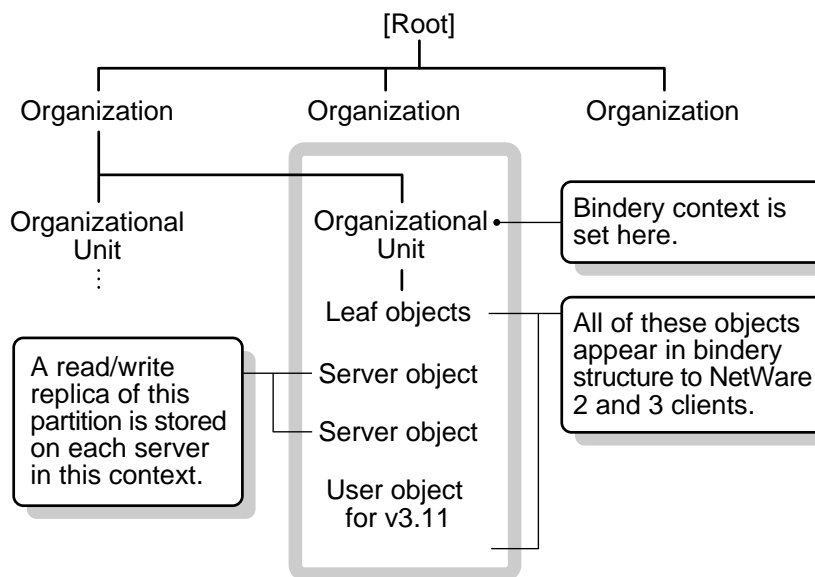
A feature of NetWare 4 that allows bindery-based utilities and clients to co-exist with NetWare Directory Services (NDS) on the network.

**B**

Objects in a bindery exist in a flat database instead of a hierarchical database like a NetWare Directory Services database. Bindery services occur when NDS provides a flat structure for the objects within an Organization object or within an Organizational Unit object.

All objects within that container object can then be accessed both by NDS clients and by bindery-based clients and servers. Bindery services applies only to the leaf objects in that Organizational Unit.

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



**Figure 1-4 Bindery services in a Directory tree**

The container object where bindery services is set is called the bindery context. You can set the bindery context by using either the graphical NetWare Setup or the command line nwcm configuration utility.

---

**WARNING:**

---

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

When you install any NetWare server into the Directory tree, a NetWare Server object is created in the same container object the server object was installed in. The user must set the bindery context for that container object.

A writable replica of the partition that the bindery context is located in must be stored on each server you want bindery services enabled on. This is done by default when you install a server into a new context.

During installation, if a NetWare server object is placed in an existing context and there are less than three replicas on the network, a writable replica of the partition containing that context is placed on the server.

Although bindery services is enabled during installation of NetWare, you can disable it by setting bindery context to null using either the NetWare Setup (graphical) or the nwcm (command line) configuration utility.

The objects in the container object set as the bindery context must conform to bindery object naming rules.

- Names can contain up to 47 characters
- Names cannot contain
  - spaces
  - slashes /
  - backslashes \
  - colons:
  - semicolons;
  - commas,
  - asterisks\*
  - square brackets[]
  - angle brackets<>
  - vertical bars|
  - plus signs+

## B

- equal signs=
- question marks?

For example, the complete name for a User object might be

MRICHARD.ACCOUNTING.HEWLETT-PACKARD US

However, only the common name of the User object MRICHARD can be seen by bindery-based clients and servers when using bindery services. Therefore, the object's common name must match bindery object naming rules.

Related utilities: "NetWare Setup"; "nwcm"; "PARTMGR" (*Utilities Reference*).

See also: "Context"; "Directory tree"; "Object."

### **Binding and unbinding**

The process of assigning a communication protocol to network boards and LAN drivers, or the process of removing it. This process does not apply to NetWare Services, but only to associated native NetWare DOS, and Windows clients and servers.

Each board must have at least one communication protocol bound to the LAN driver for that board. Without a communication protocol, the LAN driver can't process packets.

You can bind more than one protocol to the same LAN driver and board. You can also bind the same protocol stack to multiple LAN drivers on the server.

You can also cable workstations with different protocols to the same cabling scheme.

### **Binding communication protocols to boards and drivers**

To bind communication protocols to boards and drivers in native NetWare, use the INSTALL program or the BIND console command. Use BIND for each board in the server.

Until a protocol is bound to a board, users attached to the cabling scheme from the board can't log in.



When you bind a protocol to a board, you specify the cabling scheme's IPX external network number. This hexadecimal number must be different from all other numbers for cabling schemes attached to this server. The cabling scheme's IPX external network number must also be different from the internal network address for the server as well.

---

**NOTE:**

---

For NetWare OS file services, only the IPX protocol is available.

For more information on network numbers, see "IPX external network number."

**Unbinding communication protocols from boards and drivers**

To remove a communication protocol from a board and driver, use the UNBIND console command. If you have loaded the driver more than once, select the specific board you want to unbind.

Use UNBIND to unbind each board. When the protocol is unbound, users attached to the cabling scheme of the board cannot log in.

If users are already logged in, they receive a message when they attempt to access the server.

See also: "Communication protocol"; "IPX external network number"; "NetWare Protocol stack daemon (NPSD)."

**Boot files**

Files, like AUTOEXEC.BAT and CONFIG.SYS, that

- Start the operating system and its drivers
- Set environment variables

Workstation boot files depend on the client type (DOS, Windows,). For more information, see your client manual.

**BOOTCONF.SYS**

A file used by a workstation using Remote Reset to determine which remote boot image file to use.

See also: "Remote Reset."

**B**

**Bridge**

A device that retransmits packets from one segment of the network to another segment.

A router, on the other hand, is a device that receives instructions for forwarding packets between topologies and determines the most efficient path.

See also: “Router.”

**Browse right**

An object right that grants the right to see an object in the Directory tree.

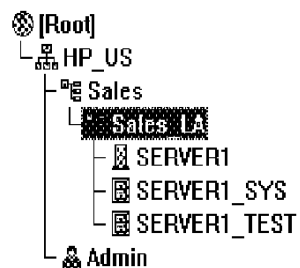
See also: “Rights.”

**Browsing**

A way of finding objects in the Directory.

Objects in the Directory are in hierarchical order. Since the Directory can be very large, it can be difficult to remember an object’s location.

Rather than trying to remember an object’s location, you can browse up or down the Directory tree and view different parts of the Directory to find the object you want.



**Figure 1-5**

**Browsing**

Example: To find object PRINTER3 when you aren’t sure where it is, look at objects in your current context.

If PRINTER3 isn't in your current context, search up or down the Directory tree until you find it, or use browser's search feature.

Related utilities: "NetWare Administrator"; "NETADMIN" (*Utilities Reference*).

### **Buffer**

An area in server or workstation memory set aside to temporarily hold data, such as packets received from the network.

See also: "Cache memory"; "Read-ahead cache."

---

## C

### **Cabling system**

Part of a network's physical layout.

See also: "Topology."

### **Cache memory**

Available random access memory (RAM) that NetWare uses to improve NetWare server access time.

NetWare Services relies mainly on the HP-UX cache system for reading and writing files.

NetWare Services also keeps local caches in shared memory. For example, a read-ahead cache in shared memory predicts what the next block request will be in a sequential file. Database requests cannot be predicted by this read-ahead cache.

Information on file locking and trustee rights is kept separately in shared memory, which can be accessed by all processes.

### **Can't Compress attribute**

A status flag that NetWare uses to indicate that a file cannot be compressed because of insignificant space savings.

NetWare Services does not support file compression or use this attribute.

See also: "Attributes."

### **Client**

A workstation that uses NetWare software to gain access to the network.

In NetWare Services, client types include DOS, and Windows.

With the respective client software, users can perform networking tasks. These tasks include mapping drives, capturing printer ports, sending messages, and changing contexts.

See also: “DOS client”; “MAIL directory”; “Windows client”; “Workstation”; “UNIX client.”

### **COM port**

Asynchronous serial port on IBM PC-compatible computer.

See also: “Serial port.”

### **Command format**

Instructions that show how to type a command at the keyboard; also called syntax.

In NetWare manuals, a command format may include constants, variables, and symbols.

### **Common name**

The naming attribute assigned to leaf objects. All leaf objects are designated by the Common Name (CN) naming attribute. For User objects, the common name is the user’s login name, for example TWILLIAM.

Other leaf objects also have common names displayed in the Directory tree, such as Printer object names or NetWare Server object names, for example PSPRINTER.

Container objects do not have common names.

See also: “Complete name.”

### **Communication**

The process of transferring data from one device to another in a computer system.

See also: “Network communication.”

### Communication protocol

Conventions or rules used by a program or operating system to communicate between two or more endpoints.

Although many communication protocols are used, they all allow information to be packaged, sent from a source, and delivered to a destination.

NetWare Services breaks information to be communicated into blocks called packets. In addition to data, each packet contains control information that can be used for addressing, error checking and other purposes. Communication protocols govern how the control information is presented.

When multiple protocols are used in a packet, “enveloping” or “encapsulating” is used to place control information at the beginning and sometimes at the end of the packet. Control information for the highest level protocol is placed closest to the data and information for the lowest level protocol is placed furthest from the data.

There are seven main communication protocols specific to NetWare:

Protocol	Function
IPX (Internet Packet Exchange)	Datagram (connectionless) transport
SPXII (Sequenced Packet Exchange)	Connection-based transport
RIP (Routing Information Protocol)	Router table maintenance
SAP (Service Advertising Protocol)	Service advertisement
NCP (NetWare Core Protocol)	NetWare file manipulation
Packet Burst Protocol	High-performance reads and writes

### Compare right

A property right that grants the right to compare any other value to a value of that property.

See also: “Rights.”

**Complete name**

An object's complete name consists of its common name (if it has one), followed by a period (.), then the name of the container object, also followed by a period, and on up through succeeding container object names through the root of the tree, for example TWILLIAM.SALES  
PV.SALES.HEWLETT-PACKARD US.

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

See also "Common name"; "Object."

**Compressed attribute**

A status flag in NetWare used to indicate that a file is compressed.

NetWare Services does not support file compression or use this attribute.

See also "Attributes."

**Computer object**

A leaf object that represents a computer on the network.

In the Computer object's properties, you can store information such as the computer's serial number or the person the computer is assigned to.

See also: "Creating Leaf Objects" (*Supervising the Network*); "Object."

**Configuration**

The setting of parameters and creation of files and directory structures that define how the elements of NetWare Services operate and interact with each other.

The key tools for configuring NetWare Services are the "NetWare Setup" (graphical user interface utility) and "nwcm" (command line utility).

See also: "nwcm"; "nwconfig."

### **Connection number**

A number assigned to any workstation that attaches to a NetWare server; it may be a different number each time a station attaches.

Connection numbers are also assigned to processes, print servers, and applications that use server connections.

The server's operating system uses connection numbers to control each station's communication with other stations.

Related utility: "NLIST" (*Utilities Reference*).

### **Connectivity**

The ability to link different pieces of hardware and software (PCs, minicomputers, and mainframes) into a network to share resources (applications, printers, and so forth).

See also: "Internetwork."

### **Console**

The monitor and keyboard where you view and control NetWare server activity.

See also "Server console."

### **Container object**

An object that holds, or contains, other objects. Container objects are used as a way to logically organize all other objects in the Directory tree.

See also "Object"; "Creating Container Objects" (*Supervising the Network*).

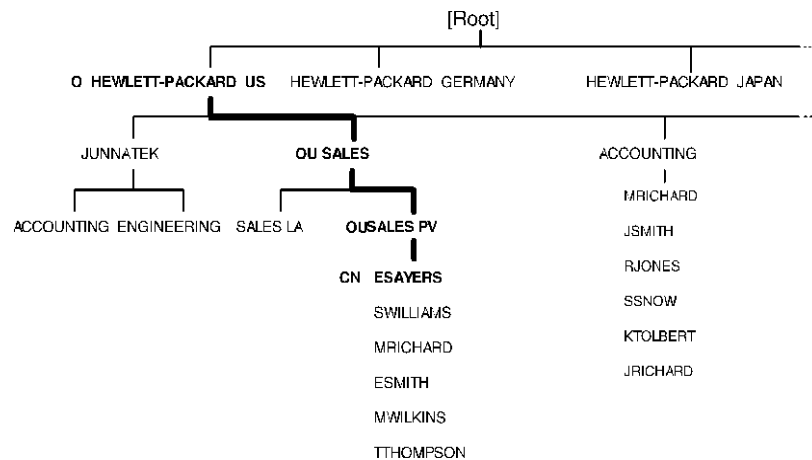
### **Context**

The position of an object in the Directory tree.

NetWare Directory Services allows you to refer to objects according to their positions in a tree. When you add an object (such as a NetWare Server or User) to the network, you place that object in a container object in the Directory tree.



For example, in the following figure, the context for the User object ESAYERS is SALES PV.SALES.HEWLETT-PACKARD US. The context for the User object RJONES is ACCOUNTING.HEWLETT-PACKARD US.



**Figure 1-6**

### Context in a Directory tree

When you move from one container object to another, you change contexts. Whenever you change contexts, indicate the complete name of the object you are changing the contexts to.

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

For example, in the previous figure, if the User object ESAYERS located in SALES PV.SALES.HEWLETT-PACKARD US wants information on User object ESMITH located in the same context, then ESAYERS need only refer to the User object as ESMITH.

See also: “Common name”; “NetWare Directory Services (NDS)”; “Object.”

### **Country object**

A container object that designates the countries where your network resides and organizes other objects in the Country object.

For example, you could use a Country object for the country where your organization headquarters reside. If you have a multinational network, you could use a Country object for each country that is a part of your network.

---

**NOTE:**

The Country object is not part of the default NetWare server installation. To use a Country object, create it at installation. Using a Country object in NetWare Directory Services is not a requirement for interoperability with other X.500-compliant directory services.

---

See also: “Creating Container Objects” (*Supervising the Network*); “Object.”

### **Create right**

In the file system, a right that grants the right to create new files or subdirectories.

In NetWare Directory Services, an object right that grants the right to create a new object in the Directory tree.

See also: “Rights.”

---

## D

### **Daemon**

A HP-UX process running in the background that can perform tasks with no user input and can spawn (initialize) other processes. Daemons provide services for clients, such as printing and server advertising. Some daemon processes such as the NetWare daemon perform administrative functions and access the host file system.

See “NetWare daemon”; “NetWare Directory Services daemons”; “NPS daemon”; “SAP daemon.”

The NetWare Services file systems provide some checks against software corruption. All safeguards against hardware corruption are dependent on the HP-UX host system.

### **Default drive**

The drive a DOS or Windows workstation is using. The drive prompt, such as A> or F>, identifies the current or default drive.

### **Default server**

The server you attach to when you log in. (The default server is specified in your NET.CFG file.)

Also, the server your current drive is mapped to.

### **Delete Inhibit attribute**

A file system attribute that prevents any user from erasing the directory or file.

See also “Attributes.”

### **Delete right**

An object right that grants the right to delete an object from the Directory tree.

See also: “Rights.”

### **Delimiter**

A symbol or character that signals the beginning or end of a command or of a parameter within a command.

For example, in the command `NCOPY F:*.* G:`, the blank space between `F:*.*` and `G:` is a delimiter that marks two distinct parameters.

Other delimiters used in NetWare include the comma (,), the period (.), the slash (/), the backslash (\), the hyphen (-), and the colon (:).

### **Device sharing**

The shared use of centrally located devices (such as printers, modems, and disk storage space) by users or software programs.

By attaching a device to a network so that several users or programs can share it, you can use resources more efficiently.

### **Director/directory**

Directory. A common name for the NetWare Directory database, which organizes NetWare Directory Services objects in a hierarchical tree structure called the Directory tree.

See also: “NetWare Directory Services (NDS).”

directory. A component in the NetWare file system, used to contain files and subdirectories.

See also: “File Systems, NetWare Services.”

### **Directory and file rights**

Rights that control what a trustee can do with a directory or file.

See also: “Rights.”

### **Directory management request**

A request that controls the physical distribution of the NetWare Directory Services database. Through these requests, network supervisors can install new Directory partitions and manage their replicas.

The following Directory management requests are supported:

<b>Request</b>	<b>Description</b>
Add partition	Creates a new Directory partition on the server.
Add replica	Adds a replica of an existing Directory partition to a server.
List replicas	Lists the Directory replicas stored by a server.
Remove partition	Deletes the master replica of a Directory partition.
Synchronize replicas	Initiates an updating of replicas of the specified Directory partition.
Create a new partition	Splits a Directory partition into two at a specified object.

Related utility: “PARTMGR” (*Utilities Reference*)

See also: “Partition, Directory Services”; “Replica.”

### **Directory Map object**

A leaf object that refers to a directory on a volume.

Directory Map objects can point to directories that contain frequently-used files such as applications.

## D

If you create a Directory Map object to point to an application, users can access the application by clicking on the Directory Map icon from the Browser.

If the application's location in the directory structure changes, you can update the object instead of having to change all users' drive mappings.

Related utilities: "MAP"; NETUSER" (*Utilities Reference*).

See also "Using Directory Map Objects" (*Supervising the Network*); "Object."

### **Directory path**

The full specification that includes server name, volume name, and name of each directory leading to the file system directory you need to access.

See also "Drive mapping"; "Directory structure, file system."

### **Directory rights**

Rights that control what a trustee can do with a directory.

See also: "Rights."

### **Directory Services**

A global, distributed, replicated database built into NetWare Services that maintains information about every resource on the network.

See also "NetWare Directory Services (NDS)."

### **Directory Services daemons**

HP-UX background processes that perform NetWare Directory Services (NDS) functions.

See also "NetWare Directory Services daemons."

### **Directory services request**

A request made to the Directory database by users or network supervisors. NetWare Directory Services requests can be divided into three types:

- Directory access requests. These requests are submitted by users who are accessing or network supervisors who are managing the Directory database's contents.

These requests support the Directory database's interface and allow objects to be created, modified, and retrieved.

- Directory access control requests. These requests set access rights to Directory objects. In this respect, directory access rights resemble file access rights.
- Directory management requests. These requests are submitted by network supervisors who manage the Directory database's physical distribution, such as partitioning.

### **Directory structure, file system**

The system NetWare uses to organize data. Each file is given a filename and stored at a specific location in a hierarchical system so that it can be located quickly.

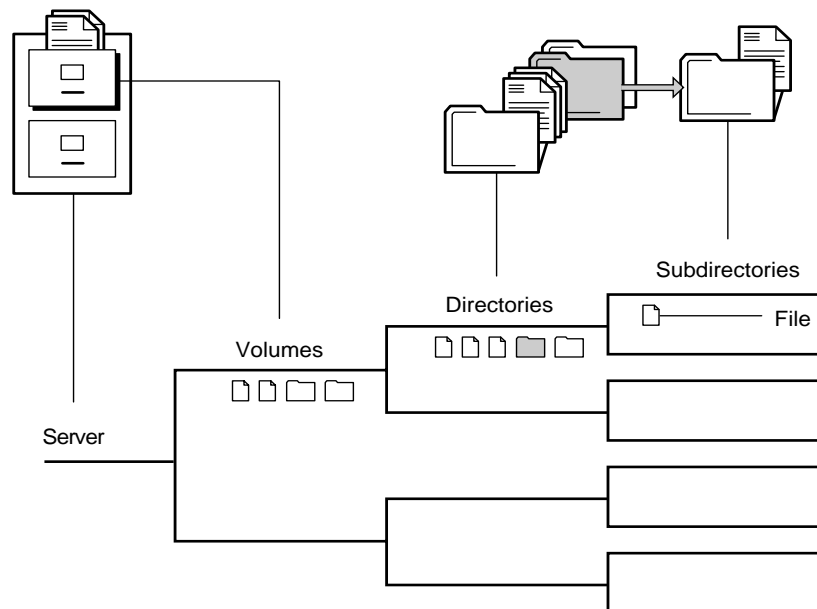
The directory structure consists of

- Directories
- Subdirectories
- Files

NetWare uses a modified hierarchical file system, with multiple root nodes called volumes. Each volume has its own tree structure with the root node as the highest point in the structure. Users must switch volumes to access a volume's resources.

A volume contains one or more directories. A directory can contain subdirectories, files, or both. A subdirectory can contain other subdirectories, files, or both. A file contains data or executable code, but cannot contain other files or subdirectories.

The file system is analogous to an office filing system, as illustrated in the following figure.



**Figure 1-7 Volumes, directories, and files**

In addition to such NetWare file system features as volumes, directories, subdirectories and files, NetWare also provides for file attributes and trustee assignments.

**Volume.** The highest level in the NetWare file system. To NetWare Services users, a volume appears much like a hard disk in a stand-alone system.

Because HP-UX does not recognize the concept of volumes, NetWare Services volumes represent paths to a particular point, called a volume mount point, in the HP-UX file system.

NetWare users accessing these volume mount points can, as rights permit, access everything below the mount point. Everything above the mount point will be invisible to NetWare users but visible to HP-UX users.

**Directory.** An area within a volume that stores files or other directories. Directories-within-directories are called subdirectories.

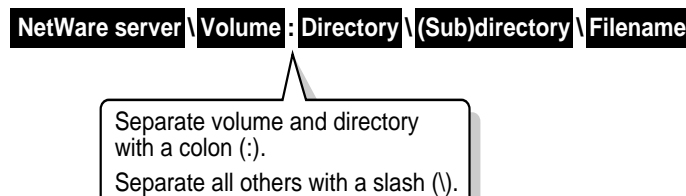
A directory can contain any number of files and subdirectories.



Files. Individual records that can be created in or copied to any level of the directory structure (except, in practice, the volume level).

### Directory path

A file or directory is located by its path, which states where the directory or file is on a volume. The following figure shows how to specify a path.



**Figure 1-8**

### Directory path conventions

Under DOS, directory names and filenames contain one to eight characters, followed by an optional filename extension. In networks with more than one client operating system, other path and naming conventions may apply (see “Filename extensions, NetWare”; “Name space”). Also, some applications restrict the number of characters in the directory path—check the application’s documentation.

### Basic NetWare directories

When volume SYS: is created, it contains predefined directories:

- SYS:ETC contains sample files to assist the network supervisor in configuring the server.
- SYS:LOGIN contains programs necessary for users to log in. .
- SYS:MAIL is used by mail programs compatible with NetWare. (NetWare creates a subdirectory in SYS:MAIL for user ADMIN.)

If you upgrade from an earlier NetWare version, existing users still have subdirectories here, but their login scripts become a property of the User object.

If you create new users after upgrading, the new users won’t have directories in SYS:MAIL.

- SYS:SYSTEM contains NetWare operating system files and NetWare utilities use for managing the network.
- SYS:PUBLIC allows general access to the network and contains NetWare

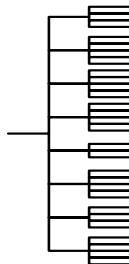
**D**

utilities and programs for network users. SYS:PUBLIC/CLIENT contains the requester software for DOS, and Windows clients.

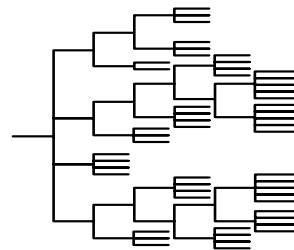
**Types of directory structures**

All directory structures are tree structures, but a directory structure can be flat with many directories coming off the volume, or it can be deep if you create several levels of directories.

Flat directory structure



Deep directory structure



**Figure 1-9**

**Flat and deep directory structures**

The general principle is to keep directory structure clean and logical. Keeping the structure relatively flat (no more than five levels deep) generally increases usability.

**Types of directories**

You can create directories for both executable files and data files, depending on what types of directories best fit the needs of your network.

Operating system directories. These store workstation operating system files.

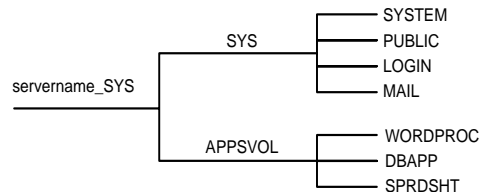
The number of operating system directories you need depends on the number of operating systems, versions, and workstation types on the network.

Application directories. Although applications can be accessed from local drives, installing them on the network provides convenient access.

Several structures are possible for application directories:

- Create a separate volume for applications, with a separate directory for each

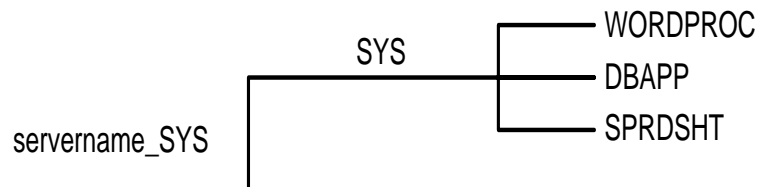
application off the root. Make trustee assignments for each application. Then go into the system or profile login script and map a search drive to each application. The following figure shows this type of directory structure.



**Figure 1-10**

**Application volume**

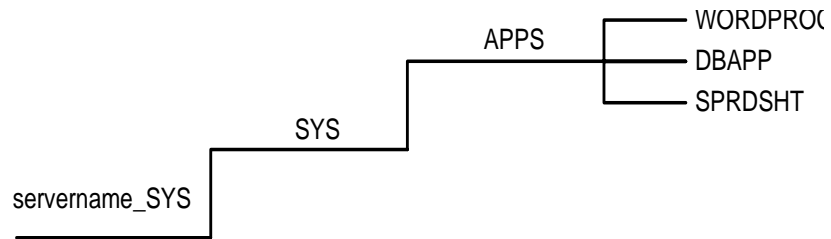
- Create a separate directory off volume SYS: for each application. Make trustee assignments for each application. Then go into the system or profile login script and map a search drive to each application. The following figure shows this type of directory structure.



**Figure 1-11**

**Application directory off volume SYS:**

- Create a parent directory for applications, with subdirectories for each application. Make trustee assignments for each application. Then go into the system or profile login script and map a search drive to each application. The following figure shows this type of directory structure.

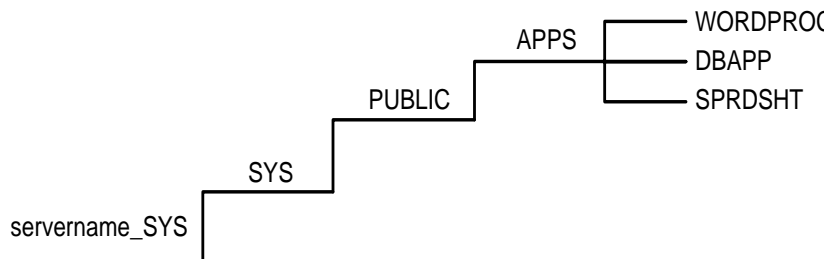


**Figure 1-12** Parent directory for applications

- Create a parent directory for applications in SYS:PUBLIC.

Because users generally have Read and File Scan rights in SYS:PUBLIC, you don't make trustee assignments or map a search drive. However, users can see and use all applications and know they exist.

Use this directory structure only if you want all users to have access to all applications. The following figure shows this type of directory structure.



**Figure 1-13** Application directory in SYS:PUBLIC

Installing applications in SYS:PUBLIC is not recommended (unless you create a subdirectory for each application).

Upgrading a network is made more complicated by mixing NetWare utilities with application program files.

An application file might have the same filename as a NetWare utility file or another application's program file. If so, one file overwrites the other, because two files with the same filename can't coexist in a directory.

**NOTE:**

Some applications write files to the root. For security reasons you don't want users working at the root level. Therefore, use MAP ROOT to map a drive to a fake root—a directory or subdirectory in which the user can be assigned rights (see “Fake root”).

**Data directories.** These are work directories for groups and users to keep work files in. You can also create a directory to transfer files between directories on the network.

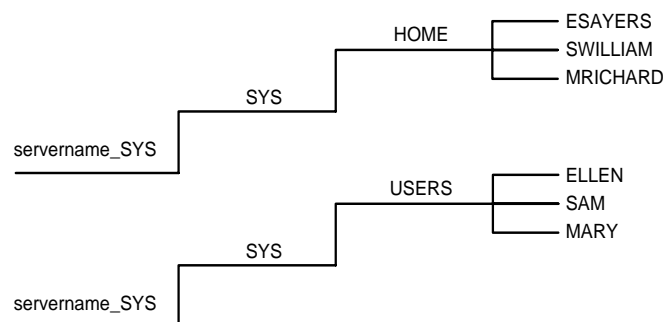
Although data can be created and stored in a home or user directory (see below), when data is stored in a user's directory, no other user (except network supervisors or managers assigned file rights) can access it.

So, to allow users to share data, create work directories and make trustee assignments for groups or users who need access to these directories.

**Home or username directories.** To provide personal workspace for users, create home or username directories.

You can create a parent directory in volume SYS: called HOME or USERS. Or, you can create a separate HOME or USERS volume. Then you can create a subdirectory for each user.

The name of each subdirectory should be the username. Usernames can be up to 47 characters, but DOS will display only 8 characters in a one-level directory name. The following figure shows this type of directory structure.



**Figure 1-14**

**Home or username directories**

- NetWare: “FILER”; “MAP”; “NLIST”; “NETADMIN”; “RENDIR” (*Utilities Reference*).

**D**

See also “Drive mapping”; “File Systems, NetWare Services”; “File system, HP-UX”; “Path”; “Parent directory”; “PUBLIC directory”; “Security”; “Subdirectory”; “SYSTEM directory.”

**Directory structure, NetWare Directory Services**

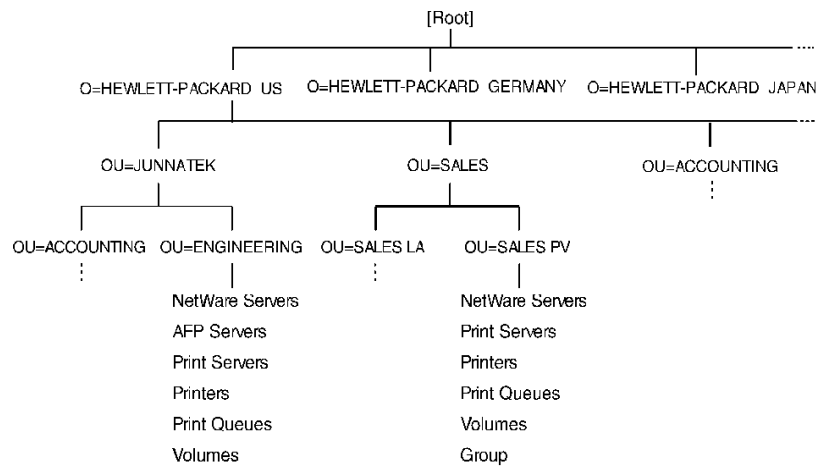
See “Directory tree”; “NetWare Directory Services (NDS).”

**Directory tree**

A hierarchical structure of objects in the Directory database. The Directory tree includes container objects that are used to organize the network.

The structure of the Directory tree can be based on a logical organization of objects, and not necessarily on their physical location.

The following figure shows an example of a Directory tree.



**Figure 1-15**

**Directory tree**

See also “Browsing”; “Object”; “Partition, Directory Services”; “Replica.”

**Disk**

A magnetically encoded storage medium in the form of a plate (also called a platter). The following types of disks are used with personal computers:

- Hard disks use a metallic base and are usually installed within a computer or disk subsystem. (In some cases, they are removable.)
- Floppy disks (also called diskettes) use a polyester base and are removable.
- CD-ROM (Compact Disc Read Only Memory) is a small plastic optical disk that isn't erasable or writable.
- Optical disks are either erasable and writable, or WORM (Write Once, Read Many).

**Disk format**

The way in which a hard disk is prepared or structured so that it can receive data from the computer's operating system. Disk formatting is a function of the operating system.

Hard disk formats include FAT (File Allocation Table) for DOS and NetWare.

For specific information on formatting your hard disk see your computer's operating system manual.

**Don't Compress attribute**

A NetWare file system attribute that prevents files from being compressed.

NetWare Services does not support file compression or use this attribute.

See also "Attributes."

**Don't Migrate attribute**

A NetWare file system attribute that prevents files from being migrated to a secondary storage device (such as a tape drive or optical disk).

NetWare Services does not support file and data migration or use this attribute.

See "Attributes."

### **DOS boot record**

A record containing information that ROM-BIOS uses to determine which device to boot from. The boot record can be on either a floppy diskette, a local hard disk, or a remote boot chip.

ROM-BIOS then runs a short program from the boot record to determine disk format and location of system files and directories.

Using this information, ROM-BIOS loads the system files (including two hidden files, IBMBIO.COM and IBMDOS.COM) and the command processor (COMMAND.COM).

### **DOS client**

A workstation that boots with DOS and gains access to the network through either

- The NetWare DOS Requester and its VLMs (for NetWare 4).
- A NetWare shell (for NetWare versions earlier than NetWare 4).

With NetWare client software for DOS, users can perform networking tasks. These tasks include mapping drives, capturing printer ports, sending messages, and changing contexts.

### **DOS filenames**

DOS versions earlier than 5.x allow the following:

- A maximum filename of 12 characters (8 characters, a dot delimiter, and a three-character extension)
- A maximum directory pathname of either 64 or 128 characters, depending on the environment

### **DOS Requester**

The DOS client software portion of NetWare.

See also "NetWare DOS Requester."



**DOS setup routine**

The routine that sets up the system configuration of your DOS client.

The setup routine records the system's built-in features (add-on boards, hard drives, disk drives, ports, math-coprocessor) and available system memory. It also lets you set date and time, password, and keyboard speed.

The system configuration is accessed from the reference diskette (for IBM PS/2 systems) or from the setup or user diagnostics diskette (for most other systems).

Instructions for running the DOS setup routine are usually contained in the introduction to your PC's operations guide.

**DOS version**

The version number and name of the kind of DOS you are using (Novell DOS 7.0, MS DOS 6.0, etc.).

Different machine types use different versions of DOS that are generally not compatible. You can determine the version of DOS your machine is using by executing the VER command at the DOS prompt.

Since all DOS versions have identically-named utilities and command interpreters the files for different DOS versions cannot be stored in the same directory.

See also: "Login scripts."

**Drive**

Physical drive. A storage device that data is written to and read from, such as a disk drive or tape drive. A drive that is physically contained in or attached to a workstation is called a local drive.

Logical drive. An identification for a specific directory located on a disk drive. For example, network drives point to a directory on the network, rather than to a local disk.

## **Drive mapping**

A pointer to a location in the file system, represented as a letter assigned to a directory path on a volume.

To locate a file, you follow a path that includes the volume, directory, and any subdirectories leading to the file.

You create drive mappings to follow these paths. You assign a letter to the path, and then use the letter in place of the complete path name.

Drive mappings can be temporary or permanent:

- Temporary mappings. To map a drive so you can use it during your current session, use the NetWare MAP utility (from a DOS workstation). The mapping is only valid until you log out.
- Permanent mappings. To make drive mappings so you can use them every time you log in, place MAP commands in your login script (see "Login scripts").

NetWare recognizes four types of drive mappings for DOS workstations: local drive mappings, network drive mappings, network search drive mappings, and Directory Map objects.

### **Local drive mappings**

Local drive mappings are paths to local media such as hard disk drives and floppy disk drives.

In DOS 3.0 and later, drives A: through E: are reserved for local mappings. To change this default, use the DOS LASTDRIVE command in your workstation CONFIG.SYS file.

### **Network drive mappings**

Network drive mappings point to volumes and directories on the network. Drives F: through Z: can be used for network mappings. Each user can map drive letters to different directories.

To create a network drive mapping, use the MAP command, the NETUSER text utility, or the NWTOOLS graphical utility

### **Network search drive mappings**

Network search drive mappings are pointers to directories containing applications, DOS files, etc., similar to the DOS PATH command.

Search drive mappings let you execute a program even if it is not located in the directory you are working in by enabling the system to search for the program.

Search drive mappings are numbered, although they also have drive letters. For example, search drive 1 (or S1) may also be known as network drive Z:

You can map up to 16 network search drives (letters K: through Z:, starting with Z:). You can't map a search drive and a regular network drive to the same letter.

When you request a file and the system cannot find it in your current directory, the system looks in every directory a search drive is mapped to.

The system searches, following the numerical order of the search drives, until either the program file is found or cannot be located.

---

**NOTE:**

---

Search drive mappings aren't supported workstations. The search functionality is provided with the OS/2 PATH, LIBPATH, and DPATH commands.

**Directory Map objects**

Directory Map objects can point to directories that contain frequently-used files such as applications.

If you create a Directory Map object to point to an application, users can access the application by clicking on the Directory Map icon from the Browser.

If the application's location in the directory structure changes, you can update the object instead of having to change all users' drive mappings.

Related utilities: "MAP"; "NETUSER" (*Utilities Reference*).

**Dynamic memory**

The most common form of memory, used for RAM. Dynamic memory requires a continual rewriting of all stored information to preserve data.

If dynamic memory is too slow for a computer's microprocessor, overall performance will suffer while the CPU waits for requested information to arrive from memory.

NetWare Glossary

**D**

A continuous electrical current is necessary to maintain dynamic memory. All data is lost from dynamic memory when the power is turned off.

---

## E

### 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 4 each time that object attempts an action.

In native NetWare, trustee assignments are kept in the file system. In NetWare Services, the trustee assignments are kept in a separate database for each volume. Network supervisors can choose how NetWare Services trustee assignments and HP-UX permissions interact to grant access. They can choose

- None. All users have access to all files and directories.
- NetWare. Netware-only enforcement. All file access is controlled by NetWare rights.
- UNIX. HP-UX-only enforcement. All file access is controlled by HP-UX permissions.
- Both. All file access is controlled by a combination of NetWare and HP-UX enforcement. In each case, the more restrictive of the two applies.

NetWare effective rights to a file or directory are determined by

- An object's trustee assignments to the directory or file.
- Inherited rights from an object's trustee assignments to parent directories.
- Trustee assignments of Group objects that a User object belongs to.
- Trustee assignments of objects listed in a User object's security equivalences list.

If a user has a trustee assignment to a directory on a given level in the directory structure, and also one on a higher level, the current trustee assignment overrides the previous one.

Trustee assignments to a group, however, are added to individual user trustee assignments.

E

Access in the host file system can affect access to files and directories even though effective rights computed by NetWare are valid.

Effective rights to an object are shown in the following figure.

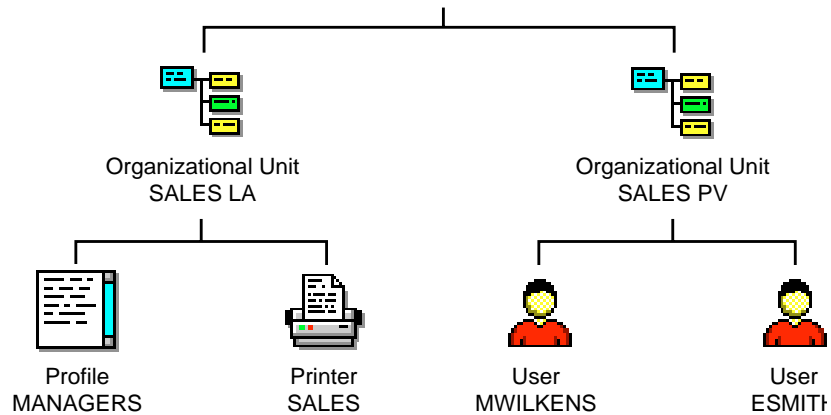


Figure 1-16

**Effective rights**

In the figure, MWILKENS' effective rights to access the MANAGERS profile can come from

- Trustee assignments on MANAGERS that list MWILKENS (explicit rights are granted).
- Trustee assignments on MANAGERS that list SALES PV (inherited from the trustee's container).
- Trustee assignments on SALES LA that list MWILKENS (inherited from the object's container).

Rights must pass through MANAGERS' Inherited Rights Filter before becoming effective.

- Trustee assignments on SALES LA that list SALES PV (inherited from object's container and trustee's container).

Rights must pass through MANAGERS' Inherited Rights Filter before becoming effective.

- Trustee assignments to any Group object that MWILKENS is a member of (only valid when the object requesting rights is a User object).
- Trustee assignments to any object listed in MWILKENS' security equivalences

list (only valid when the object requesting rights is a User object).

If MWILKENS has a trustee assignment to SALES LA and to MANAGERS, the Trustee assignment on MANAGERS overrides the trustee assignment on SALES LA.

Trustee assignments to groups, however, are added to previous trustee assignments for User objects.

No rights are granted by default. They must be granted by a trustee assignment at some point.

The Supervisor right can be masked for object and property rights, but can't be masked for directory and file rights.

Related utilities: NETADMIN"; "NetWare Administrator" (*Utilities Reference*)

See also: "File Access Control" (*Supervising the Network*); "Inherited Rights Filter, file system"; "Inherited Rights Filter, NDS object"; "Security"; "Trustee."

## **Engine**

The engine is the heart of NetWare Services. It does most of the work that makes NetWare Services act like NetWare 4. The number of engines running at any given time is dynamically configurable. The network supervisor can add more engines or delete engines while NetWare Services runs.

See also "NetWare daemon"; "Process"; "Shareable attribute."

## **Erase right**

A NetWare file system right that grants the right to delete directories, subdirectories, or files.

See also: "Rights."

## **Ethernet configuration**

The setup that allows communication using an Ethernet environment.

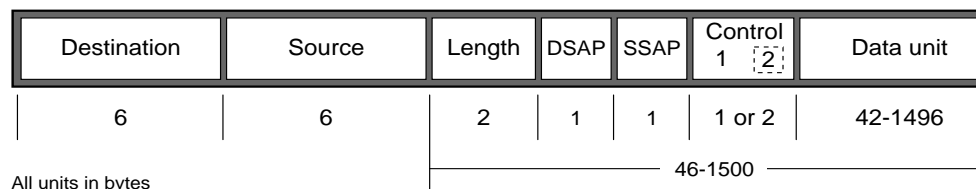
**E**

In an Ethernet environment, stations communicate with each other by sending data in frames along an Ethernet cabling system.

Different Ethernet standards use different frame formats. NetWare 4 uses the IEEE 802.2 standard by default.

NetWare Services uses 802.2 as its default if no IPX networks are found. If Networks are discovered, NWDISCOVER configures the frame type to match.

The following figure illustrates the Ethernet 802.2 frame.



**Figure 1-17**

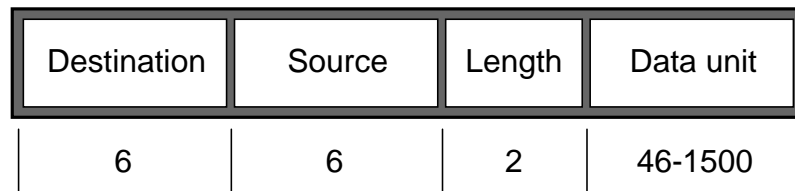
**Ethernet 802.2 frame**

Use a frame statement to configure stations for Ethernet standards other than 802.2. For servers and routers, add the frame statement to the nwconfig file. For workstations, add the frame statement to the NET.CFG file.

In addition to 802.2, you can use one of the following frame types:

- Ethernet 802.3. The default frame type used in NetWare 3.11 and earlier. This frame type is also referred to as the raw frame. Don't use this frame on a network that uses protocols other than IPX. The following figure illustrates the Ethernet 802.3 raw frame.

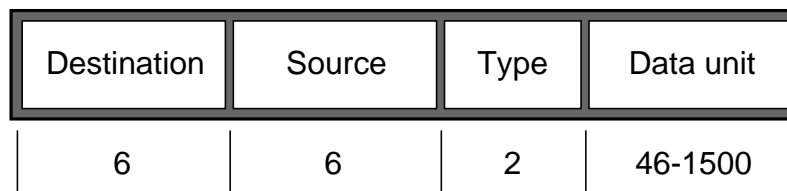




All units in bytes

**Figure 1-18 Ethernet 802.3 raw frame**

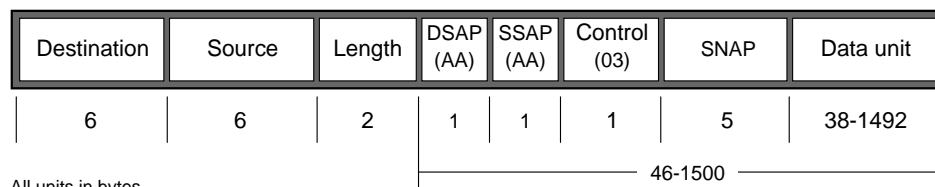
- Ethernet II. This is the default for NetWare Services. The frame type used on networks that communicate with DEC minicomputers, and on computers that use TCP/IP. The following figure illustrates the Ethernet II frame.



All units in bytes

**Figure 1-19 Ethernet II frame**

- Ethernet SNAP. The IEEE standard 802.2 frame type with an extension (SNAP) added to the header. Use this frame on networks that communicate with workstations that use protocols such as AppleTalk Phase II. The following figure illustrates the Ethernet SNAP frame.



All units in bytes

**Figure 1-20 Ethernet SNAP**

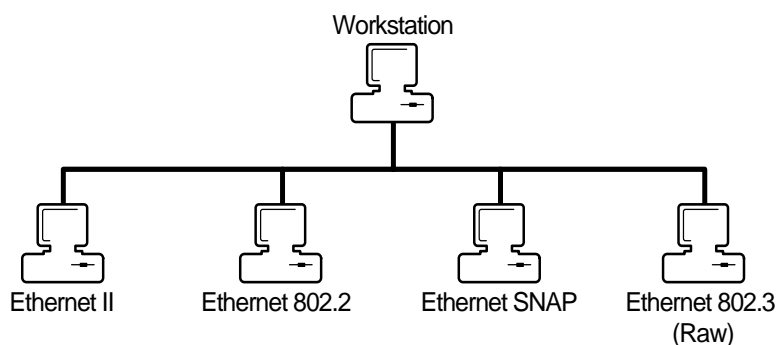
---

**NOTE:**

On Ethernet 802.2, Ethernet II, and Ethernet SNAP cabling systems, stations using different protocol numbers can coexist, but they cannot communicate directly with each other. 802.3 raw frames are able to communicate with other frames using an internal IPX router in the server.

---

Using the Open Data-Link Interface (ODI) technology on workstations and DLPI on servers, NetWare Services allows stations with different Ethernet frame types to coexist on the same Ethernet cabling system, as in the following figure:



**Figure 1-21**

**Coexisting frame types**

Because of ODI's Multiple Link Interface Driver (MLID) and link support layer (LSL), a single workstation with one network board can communicate with other devices using different Ethernet frame types.

Even though there may be only one physical network board in the computer, the MLID gives the effect of having multiple network boards and multiple LAN drivers.

Unlike traditional dedicated LAN drivers, the MLID is responsible for removing the media- (frame-) specific information from the data packets it receives.

The packets are then passed on to the LSL which functions much like a switchboard operator, sending the packet to the assigned protocol stack (such as IPX).

The following figure illustrates the ODI architecture in a multiple Ethernet frame configuration using IPX protocol.

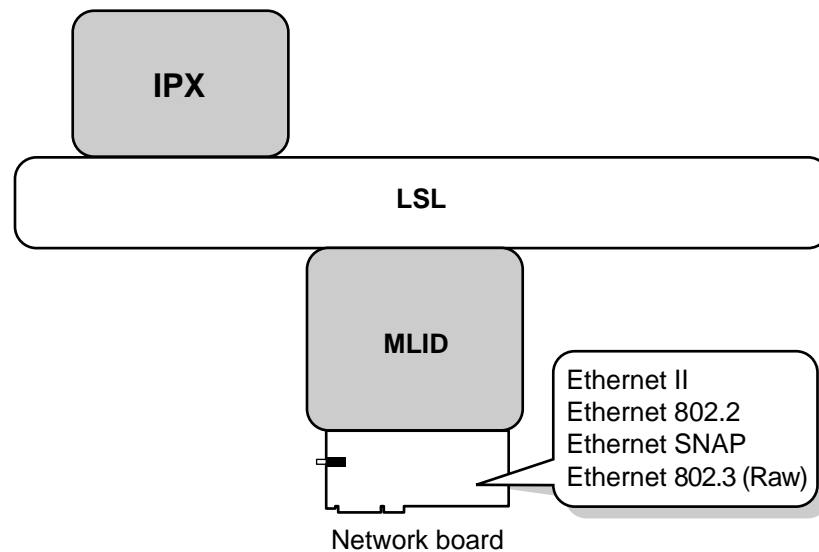


Figure 1-22

**ODI architecture using multiple Ethernet frames**

See also: “Open Data-Link Interface (ODI)”; “Multiple Link Interface Driver (MLID)”; “Link Support Layer (LSL)”; “Packet.”

**Execute Only attribute**

A NetWare file system attribute that prevents a file from being copied.

See also “Attributes.”

**Extended attribute**

A High Performance File System (HPFS) convention that allows information about a file to be attached to a file or directory. Each extended attribute has two parts:

- A name (null-terminated string)
- A value (text, bitmap, or binary data) up to 64KB

Standard extended attributes conventions require that extended attribute names begin with a period.

## NetWare Glossary

### E

Some typical standard extended attributes are: .TYPE, .ICON, .HISTORY, .SUBJECT, .KEYPHRASES, .APPTYPE, and .ASSOCTABLE.

The application that creates the extended attributes and the applications that read the extended attributes must recognize the format and meaning of the data associated with the given name.

In NetWare Services, extended attribute files are stored in a hidden subdirectory off the directory that stores the main files. The extended attribute files have the same names as the main files.

---

## F

### Fake root

A subdirectory that functions as a root directory. NetWare allows you to map a drive to a fake root (a directory where rights can be assigned to users).

---

**NOTE:**

Fake roots work with the NetWare 4 DOS Requester, as well as with NetWare shells included with NetWare 2.2 and 3.x. . ( The search functionality is provided with the OS/2 PATH, LIBPATH, and DPATH commands.)

Some applications cannot be run from subdirectories; they read files from and write files to the root directory. However, for security, don't assign users rights at the root or volume directory level.

To use an application that must be installed at the root, load the files in a subdirectory and designate it as a fake root directory in the login script.

You cannot use the DOS CD (change directory) command at the fake root to return to the original root. To change the fake root back to the original root, remap the drive.

Related utility: "MAP" (*Utilities Reference*).

See also "Security."

### FAT

See "File Allocation Table (FAT)."

### File Allocation Table (FAT)

An index table that points to the disk areas where a file is located on a DOS workstation.

### **File caching**

The method of holding recently-used data in cache memory to enable quick access to frequently requested files. NetWare Services uses the file caching of the host operating system. In addition, it uses read-ahead cache.

See also: “Cache memory”; “Read-ahead cache.”

### **File compression**

In NetWare 4, a means of allowing more data to be stored on server hard disks by compressing (packing) files.

NetWare Services does not support file compression.

### **File handle**

A number used to refer to or identify a file.

See also: “Handle.”

### **File locking**

The means of ensuring that a file is updated correctly before another user, application, or process can access the file. File locking physically locks a file so that access by multiple users does not result in data being overwritten or corrupted, or in the system locking up entirely.

For example, without file locking, if two users attempt to update the same file simultaneously, one user could overwrite the file update of the other user.

Timeout values in most locking functions specify a maximum wait time for a lock to be released before returning to the program.

See also “Record locking”; “Semaphore”; “Synchronization services.”

### **File rights**

Rights that control what a trustee can do with a file.

See “Rights.”

**File Scan right**

A file system right that grants the right to see the directory and file with the DIR or NDIR directory command.

See also: “Rights.”

**File sharing**

A feature of networking that allows more than one user to access the same file at one time.

See also: “Rights”; “Attributes.”

**File Systems, NetWare Services**

NetWare Services supports the Standard file system. The Standard file system is portable to any HP-UX system and to non-HP-UX systems such as VMS.

For each volume, the Standard file system uses an inodes file to store NetWare and client specific information, a database to store NetWare trustee assignments, and a HP-UX file system to store actual data files and file forks. The inodes file uses the same basic format as the NetWare 4 directory entry file.

The Standard file system must synchronize the information in the inodes files with HP-UX file system and duplicate some functions of the HP-UX file system. The level of synchronization is controlled by the `npfs_min_sync_interval` parameter.

The Standard file system provides the following:

- Support by most types of HP-UX partitions and portability to non-HP-UX systems.
- Tight integration between NetWare rights and HP-UX permissions. The network supervisor can set up NetWare security, which NPFS enforces unless overridden by the `file_access_control` parameter.
- NetWare attribute enforcement. NetWare directory and file attributes are stored and enforced by the Standard file system. They can be set using NetWare utilities.
- Name space mapping. Every file and directory has both a DOS and a HP-UX

name.

#### Standard file system volumes

- Can exist on any native HP-UX partition type including ufs, vxfs, s5, and sfs.
- Are created by specifying a path to a mount point.
- Use a NetWare inodes file (USInodes) to maintain NetWare information.
- Use an extended names file for HP-UX names longer than 28 characters and OS/2 names longer than 96 characters.
- Use a NetWare database to maintain trustee assignments and modifications to Inherited Rights Masks.
- Can be mounted on Network File System (NFS). The NFS mount should be a “hard” mount so that if the other system is not accessible the NFS system will halt processing until the other system is restored.
- Support symbolic links.
- Are configurably synchronized with the HP-UX file system.

### **HP-UX Partitions and NetWare Volumes**

HP-UX uses a hierarchical file system where root (labeled “/”) is at the top of the hierarchy. A partition is mounted at a particular HP-UX directory at or below the root and appears to the user as a directory.

NetWare uses a modified hierarchical file system, with multiple root nodes called volumes. Each volume has its own tree structure and users must switch volumes to access the volume’s resources.

Because HP-UX does not recognize the concept of volumes, NetWare volumes are paths to a particular point, called a mount point, in the HP-UX file system. NetWare users accessing these volume mount points can, as rights permit and partitions permit, access everything below the mount point. Everything above the mount point will be invisible to NetWare users but visible to HP-UX users.

See also “Directory structure, file system”; “File system, HP-UX”; “Inherited Rights Filter, file system”; “Rights”; “Volume.”



**File system, HP-UX**

The HP-UX file system is hierarchical and represented as a tree. It has a single root directory but can have many subdirectories, which can contain files and other subdirectories.

Each hard disk may be divided into one or more partition. Each partition is assigned to a particular type of file system. These file systems are then mounted into a single hierarchical tree and appear as a single file system.

HP-UX systems allow multiple file system types, for example, UNIX File System (ufs), Veritas File System (vxfs), and System V (s5).

NetWare volumes are mount points in the HP-UX file system. To the NetWare user, the volume level appears to be the root. It is, however, only one branch of the HP-UX file system.

See also “File Systems, NetWare Services.”

**File Transfer Protocol (FTP)**

A set of control procedures to prevent errors in information transmitted between network stations.

The data is sent from one station to another in packets. Each packet includes a discrete number that is derived from the data that makes up the packet, according to a mathematical algorithm.

The algorithm is applied to each data packet a second time when it arrives on the receiving end.

If the number on the receiving end does not match the number included in the packet, the receiving station sends a signal to the transmitting station requesting that the packet be resent.

**Filename extensions, NetWare**

The extension used after the period in NetWare filenames.

## NetWare Glossary

### F

DOS filename extensions used by NetWare:

- .EXE—Executable file
- .BAT—Executable batch file
- .DAT—ASCII text file (usually)
- .COM—Executable command file
- .ERR—Error log file
- .OVL—Overlay file used with NetWare menu utilities
- .HLP—Help screens (F1 help) in a menu or graphical utility
- .MSG—Message file
- .SYS—Operating system or driver file

Extensions specific to NetWare 4:

- .PDF—NetWare printer definition file
- .QDR—NetWare print queue definition directory
- .Q—NetWare print job file
- .NDS—NetWare Directory Services file
- .001—Unicode table
- .MSG—Text messages

### **Flag**

A marker set for a directory or file that tells NetWare what to do with the directory or file.

See also “Attributes.”

### **Form**

In a NetWare printer command, the name and size of the paper used for a print job.

See also “Printer form.”

### **Frame**

A packet data format for a given media.

Some media support multiple packet formats (frames), such as Ethernet 802.2, Ethernet 802.3, Ethernet II, Ethernet SNAP, Token-Ring, or Token-Ring SNAP.

For NetWare 4 and NetWare Services, the default Ethernet frame type is 802.2.

See also “Ethernet configuration.”

### **FTP**

(File Transfer Protocol) Control procedures to prevent errors in information transmitted between workstations.

For complete information, see “File Transfer Protocol (FTP).”

---

## G

### Gateway

A link between two networks.

A gateway allows communication between dissimilar protocols (for example, NetWare and non-NetWare networks) using industry standard protocols such as TCP/IP, X.25, or SNA.

### Group object

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

A Group object isn't a container object like an Organization Unit object. A Group object has a list of User object names.

Whenever electronic mail is sent to a Group object, or whenever a trustee assignment names a Group object, each user in the list is part of that action.

You can create Group objects based on who uses the same applications, printers, or print queues; who performs similar tasks; or who has similar needs for information.

You can use Group objects to simplify trustee assignments and login scripts. For example, instead of repeating a trustee assignment for each user, you can create a Group object that lists the users and use just one trustee assignment.

Related utilities: FILER"; "NETADMIN"; "NetWare Administrator"  
(*Utilities Reference*)

See also: "Managing Group Objects" (*Supervising the Network*); "Effective rights"; "Object."

---

## H

### **Handle**

A pointer used by a computer to identify a resource or feature.

For example, a directory handle identifies a volume and a directory, such as SYS:PUBLIC.

Other types of handles used to access NetWare include file handles, request handles, device handles, and volume handles.

### **Hard disk**

A high-capacity magnetic storage device that allows a user to write and read data. Hard disks can be network or local workstation disks.

See also: "Disk."

### **HCSS**

See "High Capacity Storage System (HCSS)."

### **Hexadecimal**

A base-16 numeric notation system the uses of which include specifying addresses in computer systems.

In hexadecimal notation, the decimal numbers 0 through 15 are represented by the decimal digits 0 through 9 and the alphabetic digits A through F (A = decimal 10, B = decimal 11, etc.).

### **Hidden attribute**

A DOS attribute that hides a directory or file from the DOS DIR command and prevents the directory or file from being deleted or copied.

See also: "Attributes."

### **High Capacity Storage System (HCSS)**

A system that increases data storage capacity by integrating an optical disk library into the NetWare file system.

HCSS is not supported in NetWare Services.

### **High Performance File System (HPFS)**

NetWare Services file systems support the following features of OS/2's High performance file system:

- Long filenames of up to 253 characters
- Extended attributes

Disk partitions and memory features are not supported.

See also: "Extended attribute."

### **Home directory**

A private network directory that the network supervisor can create for a user. The user's login script should contain a drive mapping to his or her home directory.

### **Hop count**

The number of network boards a message packet passes through on the way to its destination on an internetwork.

The destination network can be no more than 16 hops (NetWare server or router interface boards) from the source.

The NetWare Services server utilities TRACK ON and DROUTER show how many hops other recognized networks are away from the server router.

They also show the number of ticks (1/18 of a second) it takes for the message packet to reach its destination network.

See also "Partition, Directory Services."

## HPFS

See “High Performance File System (HPFS).”

## Hybrid user

A hybrid user is a user who can own or access the same files when logged in either as a NetWare user or as a HP-UX user.

On a NetWare Services server, three types of users are possible:

- NetWare users have a NetWare user account but do not have a HP-UX user account
- HP-UX users have a HP-UX user account but do not have a NetWare user account
- Hybrid users have both HP-UX user accounts and NetWare user accounts and can access the same files from either account

When NetWare users log in and create files, they maintain rights to the files because of trustee assignments. When HP-UX users log in and create files, they maintain rights to the files because they own the files.

However, when NetWare users create files, NetWare Services must assign a HP-UX owner to the file to store it on a HP-UX file system. When NetWare Services is installed, three HP-UX user accounts must be set up for this purpose:

- nwroot
- nwuser
- nobody

Two HP-UX groups must be set up to assign group permissions

- nwgroup
- nogroup

These accounts and groups must be set up manually from HP-UX.

NetWare Services runs as a HP-UX root process and sets the owner, group, and permission mask of all files created by NetWare users. If there are no hybrid users, these files are owned by nwuser, assigned to nwgroup, and assigned a permission mask as defined in the nwconfig file.

**H**

Users with both NetWare and HP-UX accounts can be made hybrid users by associating HP-UX login names with NetWare Directory Services common names. You can set up hybrid users with the graphical utility User Setup, or by modifying the `nwuser` file in the `/etc/netware4` directory. In this case the owner of files created by the NetWare server is the HP-UX user that the NetWare user maps to.

The value of hybrid mapping is that the HP-UX users retain permissions to the files they created as NetWare users, because the files are assigned the user's own UID and GID rather than the default NetWare UID and GID.

Parameters in `nwcm` control whether hybrid user mappings are allowed.

Related utility: “`nwcm`” (*Utilities Reference*).

See also: “Setting Up a Hybrid User” (*Supervising the Network*); “Attributes”; “NVT2”; “Rights.”



---

## I

### **Identifier variables**

Variables used in login scripts that allow you to enter a variable (such as LOGIN\_NAME) in a login script command, rather than a specific name (such as RICHARD).

See “Login scripts.”

### **Immediate Compress attribute**

A file system attribute that causes NetWare files to be compressed as soon as the operating system can do so, without waiting for a specific event to occur (such as a time delay).

NetWare Services does not support file compression or use this attribute.

See “Attributes.”

### **Indexed attribute**

A status flag set when a NetWare file exceeds a set size, indicating that the file is indexed for fast access.

NetWare Services does not support file indexing or use this attribute.

See “Attributes.”

### **Inherited Rights Filter, file system**

The method of controlling which rights users can inherit. An Inherited Rights Filter (IRF) is given to each file or directory when it is created. The IRF for any given file or directory is modified by revoking rights.

To change the IRF of a file or directory, you must have the Access Control right to that file or directory.

The directory's IRF controls which parent directory effective rights can be exercised in the current directory. The file's IRF controls which of the current directory's rights can be exercised in the file.

The following figure shows how a trustee assignment to a parent directory is inherited by a file or subdirectory. Compare Figure 24.

File system	directory and file rights
1. Nick's trustee assignment to directory PROJECTS:	[ RWCE F ]
2. IRF on file PLAN:	[ SR C F ]
3. Nick's effective rights to PLAN:	[ R C F ]
A right must be in both lines 1 and 2 to flow to line 3.	

Figure 1-23

**Inheritance of file system trustee assignment**

The Supervisor right cannot be blocked in the file system. A trustee who has the Supervisor right in the root directory of a volume has the Supervisor right to the entire volume: it can't be blocked with an IRF.

**Inherited Rights Filter, NDS object**

A list of rights that can be created for any object, which controls the rights a trustee can inherit from container object.

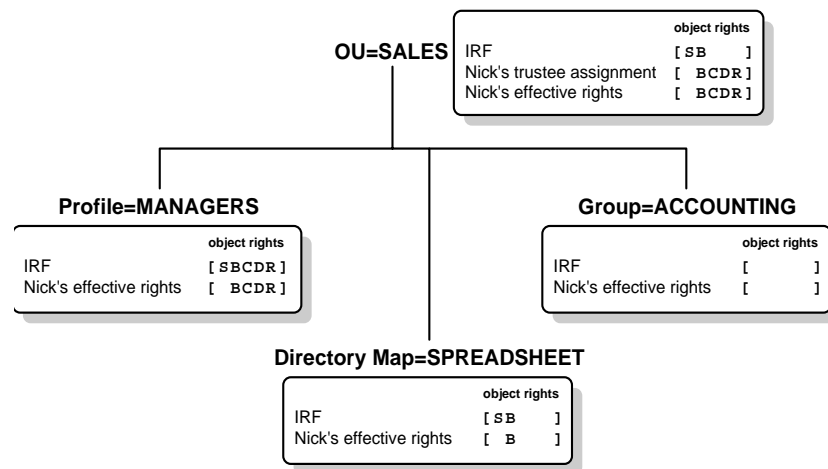
The Inherited Rights Filter (IRF) for any object is part of the access control information for that object.

To change the IRF of an object, you must have at least the Write and Read property rights to the ACL property of that object.

The IRF cannot grant rights, it can only revoke rights.

The effect of the IRF, for every object that doesn't have a trustee assignment to an object, is: "Whatever rights to this object you would have inherited, I am revoking all but these rights."

In the following figure, Nick's trustee assignment to Organizational Unit SALES grants him BCDR object rights.



**Figure 1-24**

#### **Inherited Rights Filter**

Because Nick does not have a trustee assignment to any of the three objects within that container, Nick's effective rights to those objects are inherited from SALES, and must pass through the IRF of each object.

The IRF for MANAGERS allows all rights to pass through, so Nick's rights to MANAGERS are the same as his rights to SALES. SPREADSHEET and ACCOUNTING block some or all of the rights that Nick was granted to SALES, so they aren't effective on those objects.

The IRF of an object and its properties can block the Supervisor right. This allows distributed management of the Directory tree.

NetWare utilities won't allow you to block the Supervisor right, however, unless a trustee is already granted the Supervisor right to that object. This prevents cutting off Supervisor-level access to a part of the Directory tree.

Because of the ability to block the Supervisor right to objects and properties, you should grant a trustee all rights that are appropriate.

For example, do not grant the Supervisor right only. Even though that right allows all actions on an object, if the Supervisor right is blocked, the trustee will be left with no rights.

Instead, grant the trustee all rights, so that if Supervisor is blocked by an IRF, the trustee will still have Browse, Rename, Create and Delete rights.

The following figure shows how a trustee assignment to a container object is inherited by an object lower in the structure.

The following figure shows how a trustee assignment to a container object is inherited by an object lower in the directory structure. Compare Figure 24.

Directory tree	object rights	all property rights
1. Nick's trustee assignment to SALES:	[ B ]	[ CRW ]
2. IRF on PROFILE:	[ SB R ]	[ CR ]
3. Nick's effective rights to PROFILE:	[ B ]	[ CR ]
A right must be in both lines 1 and 2 to flow to line 3.		

Figure 1-25

**Inheritance of NDS trustee assignment**

Notice that rights allowed in the IRF aren't granted to a trustee; instead, of the rights already granted to a trustee, only those rights listed in the IRF are allowed to pass through and be effective at the lower level.

Related utilities: "FILER"; "NETADMIN"; "NetWare Administrator"; "RIGHTS" (*Utilities Reference*).

See also: "Effective rights"; "Security."

**Internal network number**

A network number that uniquely identifies an individual NetWare 4 server; usually referred to as the IPX internal network number.

See "IPX internal network number."

**Internet Protocol (IP)**

An industry-standard suite of networking protocols, enabling dissimilar nodes in a heterogenous environment to communicate with one another.

See also: "TCP/IP."

### **Internetwork**

Two or more networks connected by a router.

Users on an internetwork can use the resources (files, printers, hard disks) of all connected networks, provided they have security clearance.

See also: "Router."

### **Internetwork Packet eXchange (IPX)**

A Novell communication protocol that sends data packets to requested destinations (for example, workstations and servers).

IPX addresses and routes outgoing data packets across a network. It reads the assigned addresses of returning data and directs the data to the proper area within the workstation's or NetWare server's operating system.

IPX is closely linked with other programs and routines that help in the network data-transmission process.

The NetWare DOS Requester prepares data packets in a form understandable to the intended destination before handing them to IPX.

The IPXODI.COM file then uses the services of a LAN driver routine to control the station's network board for data delivery.

IPX can route and accept data packets through physically different networks and workstations.

See also: "Communication protocol"; "IPXODI"; "LAN driver, client"; "LAN driver, client"; "Open Data-Link Interface (ODI)."

### **Internetwork Packet eXchange Open Data-Link Interface (IPXODI)**

A module that takes workstation requests that the DOS Requester has determined are for the network, packages them with transmission information (such as their destination), and hands them to the link support layer (LSL).

IPXODI attaches a header to each data packet. The header specifies information that targets network delivery, announcing where the packet came from, where it's going, and what happens after delivery.

Because IPXODI transmits data packets as datagrams (self-contained packages that move independently from source to destination), it can only deliver the packets on a best-effort basis. Delivery is assured by SPX.

See also: "IPX"; "Link Support Layer (LSL)"; "NetWare DOS Requester"; "SPXII."

### **Interoperability**

The ability to use products from different vendors within the same system.

For example, Novell's ODINSUP interface allows LAN Manager, LAN Server, or other NDIS protocols to co-exist with NetWare's ODI on a network.

Another example is Novell's network driver interface, the Open Data-Link Interface (ODI), on which any protocol, including the IPX protocol, can be used.

Communication protocols such as Internet Protocol (IP) or Appletalk Filing Protocol (AFP) can be used in ODI to process information from the network without the user having to know each protocol's required method of packet transmission.

Interoperability also means that an application running on different platforms can share files.

See also "Open Data-Link Interface (ODI)"; "ODINSUP."

### **Interprocess communication (IPC)**

The ability of two processes to communicate with each other or exchange data.

Shared memory, HP-UX sockets, and streams are different mechanisms that enable existing programs to communicate with processes on other systems.

For example, NetWare Services processes communicate using variables stored in shared memory. Access to the shared memory is synchronized using variable granularity locks.

NetWare Services also uses streams and message queues for IPC.

See “Synchronization services.”

## **IPX**

See “Internetwork Packet eXchange (IPX).”

### **IPX external network number**

A four-byte network address that identifies a specific logical network or LAN on an IPX internetwork. All NetWare servers, routers, and clients cabled to that segment and using a common frame type must use the same network address.

It is an arbitrary hexadecimal number, one to eight digits (1 to FFFFFFFE) assigned when the IPX protocol is bound to a network board in the server.

You can bind IPX with two different frame types (such as IEEE 802.2 and 802.3) to the same network board, but each frame type must be given a unique logical IPX external network number, even though both frame types are bound to the same network board and physical cable segment.

The terms network number and network address are sometimes used to refer to the IPX external network number.

See also: “IPX internal network number”; “IPX internetwork address”; “Network numbering.”

### **IPX internal network number**

A logical network number that identifies an individual NetWare server. Each server on a network must have a unique IPX internal network number.

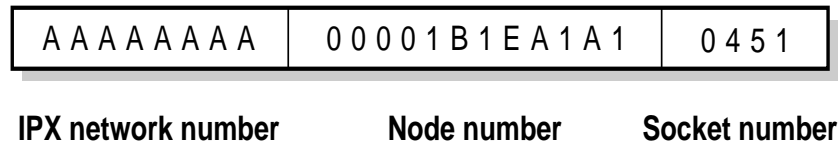
The IPX internal network number is a hexadecimal number, one to eight digits (1 to FFFFFFFE), and is assigned to the server during installation and specified in the configuration file.

In earlier versions of NetWare, the IPX internal network number was referred to as the internal network number.

See also: “IPX external network number.”

### **IPX internetwork address**

A 12-byte number (represented by 24 hexadecimal characters) divided into three parts, illustrated in the following figure.



**Figure 1-26**

#### **IPX internetwork address**

The first part is the 4-byte (8-characters) IPX network number. The second part is the 6-byte (12-character) node number. The third part is the 2-byte (4-character) socket number.

See also: “IPX external network number”; “Node number”; “Socket.”

### **IPXODI**

See “Internetwork Packet eXchange Open Data-Link Interface (IPXODI).”



---

**J**

**Jumper block**

A group of jumper pins that can be connected (jumped) or left unconnected to make hardware configuration settings on a circuit board.

---

## L

### LAN

See “Local Area Network (LAN).”

### LAN driver, client

A LAN driver serves as a link between a station's operating system and the physical network parts.

NetWare 4 clients are designed for LAN drivers written to the Open Data-Link Interface (ODI) specification.

ODI drivers connect directly to the ODI model's Link Support Layer (LSL), which serves as an intermediary between the drivers and the communication protocols.

See also: “Link Support Layer (LSL)”; “ODI.”

### Large Internet Packet (LIP)

Generally the size of packets that cross bridges or routers on NetWare networks has been limited to 576 total bytes. The 576 bytes include the following:

- 512 bytes for data
- 30 bytes for the IPX header
- 34 bytes for the NCP or SPX header

Some network architectures, such as Ethernet and token ring, allow larger packets to be sent over the network.

By allowing the NetWare packet size to be increased, LIP enhances the throughput over bridges and routers.

### **How LIP works**

In NetWare versions earlier than NetWare 4, the workstation initiated a negotiation with the NetWare server, during the connection process, to determine an acceptable packet size.

If, during this negotiation, the server detected a router between it and the station, the server made the maximum packet size 576 bytes.

In NetWare 4, the station still initiates acceptable packet size. However, the server does not default to 576 bytes when a router is detected.

Instead, the station checks the maximum size supported by the router, thereby allowing the station more flexibility in determining an acceptable packet size.

LIP functionality is implemented for DOS clients through the station's NET.CFG file.

On the NetWare Services server, LIP-related configuration parameters can be found in the nwconfig file.

For more information on LIP, see "Setting up Additional Protocol Support" in the *NetWare Client for DOS and MS Windows Technical Reference*.

### **Leaf objects**

Objects that don't contain any other objects, located at the end of a branch in the Directory tree.

Files are also leaf objects in the file system.

See also: "Object."

### **Link Support Layer (LSL)**

A software module that implements the interface between drivers and protocol stacks. It essentially acts like a switchboard, directing packets between the drivers and protocol stacks.

Any ODI LAN driver can communicate with any ODI stack protocol through the LSL. The LSL handles the communication between the protocol stack and MLIDs.

**L**

LSL is part of the ODI architecture. In NetWare Services, it is implemented only on the client portion of the network.

See also: “Open Data-Link Interface (ODI).”

**LIP**

See “Large Internet Packet (LIP).”

**Loadable module**

A program that can be loaded and unloaded from NetWare client memory while the system is running. In NetWare 4 there are two common types: NLMs (NetWare Loadable Modules) for NetWare servers and VLMs (Virtual Loadable Modules) for NetWare clients.

NetWare Services does not use NLMs.

See “Virtual Loadable Module (VLM).”

**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).”

**Lock manager**

See “Synchronization services.”

**Logical memory**

Memory that may not have contiguous addresses, but which appears contiguous to a process.

**Login**

The procedure that provides access to the network by using the LOGIN command.

When a user initiates a login request, the operating system looks for security rights; the user is then asked for a password.

All security information is placed into the NetWare server's connection list and the user is said to be logged in.

At this point, LOGIN executes one or more login scripts (which initialize environment variables, maps network drives, and so forth).

Related utility: "LOGIN" (*Utilities Reference*).

See also: "LOGIN directory"; "Login restrictions"; "Logout"; "Login scripts."

### **LOGIN directory**

The SYS:LOGIN directory, created during network installation, that contains the LOGIN and NLIST utilities. Users can use these utilities to log in and view a list of available NetWare servers.

Do not delete the LOGIN directory.

See also: "File Systems, NetWare Services"; "MAIL directory"; "PUBLIC directory"; "SYSTEM directory."

### **Login restrictions**

Limitations on a user account that control access to the network, including

- Requiring a password. If you require a password, you can specify its minimum length, whether it must be changed (and how often), whether it must be unique, and whether the user can change it.

You can also specify the number of times a user can log in using an expired password, and the number of incorrect login attempts allowed.

- Setting account limits. If you install Accounting, you can assign account limits, like an account balance or expiration date.
- Specifying the number of connections. You can limit the number of times a user can log in simultaneously. You can also specify, by node address, which workstations users can log in on.
- Setting time restrictions. You can restrict the hours during which users can log in. You can assign all users the same hours, or you can restrict users individually.

## L

When a user violates login restrictions, NetWare Services disables the account and no one can log in using that username. This prevents unauthorized users from logging in.

### **Login scripts**

A file containing commands that set up your users' DOS workstation environments whenever they log in. Login scripts are similar to configurable batch files and are executed by the LOGIN utility.

You can use login scripts to

- Map drives and search drives to directories.
- Display messages.
- Set environment variables.
- Execute programs or menus.

Login scripts work the same way for DOS and Window workstations.

### **Three types of login scripts**

When a user logs in, the LOGIN utility executes the appropriate login scripts. Three types of login scripts can be used together to specify a custom environment for your users. All three types of login scripts are optional.

- Container login scripts set general environments for all users in an Organization or Organizational Unit. These login scripts execute first.
- Profile login scripts set environments for multiple users. These login scripts execute after the container login script.
- User login scripts set environments specific to a single user, such as menu options or a username for electronic mail. These login scripts execute after system and profile login scripts.

The LOGIN utility contains a default login script. It contains only essential commands, such as a drive mapping to NetWare utilities. This default login script executes for any user who does not have an individual user login script.

If you don't want to create a user login script and you want to prevent the default login script from executing, you can disable the default script by including the NO\_DEFAULT command in the container login script.

**Deciding which login scripts to create**

Maintaining many user login scripts can be time-consuming. Therefore, include as much customizing information as possible in the container login scripts, which are fewer in number and easier to maintain.

For example, if all users need access to NetWare utilities in the same volume, put the search drive mapping to that volume in a single container login script rather than in every user login script.

Create profile login scripts if there are multiple users with identical login script needs.

Finally, in user login scripts, include only those individual items that can't be included in profile or container login scripts.

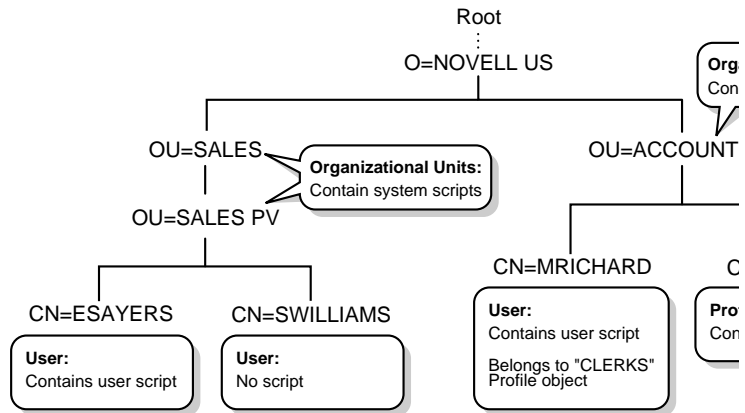
Since up to three login scripts can execute whenever a user logs in, conflicts can occur. If this happens, the last login script to execute (usually the user login script) overrides any conflicting commands in a previous login script.

Login scripts are properties of objects.

The following table shows which objects can contain login scripts.

<b>Object</b>	<b>Type of login script</b>
Organization	container
Organizational Unit	container
Profile	Profile
User	User

The following figure shows how the different types of login scripts can reside in a Directory tree and how they affect users.



**Figure 1-27**      **Where login scripts are located**

In the previous figure, there are three users, ESAYERS, SWILLIAMS, and MRICHARD. The following table shows which login scripts execute when each user logs in.

When this users logs in	Login scripts execute in this order
ESAYERS	1. SALES PV's container login script 2. ESAYERS's user login script
SWILLIAMS	1. SALES PV's container login script 2. Default user login script
MRICHARD	1. ACCOUNTING's container login script 2. CLERKS' profile login script 3. MRICHARD's user login script

Container login scripts only affect users immediately below the Organization or Organizational Unit that contains the login script.

For example, in the figure, although there are two levels of container objects above users ESAYERS and SWILLIAMS, only the container login script immediately above them (at OU=SALES PV) executes when they log in.



If SALES PV had no container login script defined, no container login script would execute for ESAYERS and SWILLIAMS, even though a container login script exists at a higher level.

Since SWILLIAMS has no user login script defined, the default login script executes after the container login script.

Since MRICHARD belongs to the profile CLERKS, the CLERKS' profile login script executes before MRICHARD's user login script. Users can belong to only one Profile, so only one profile login script can execute for any user.

### **Creating, modifying, and copying login scripts**

You can use either the NETADMIN utility or the NetWare Administrator utility to create or modify login scripts.

The main difference in creating container, profile, and user login scripts is the object you select to contain the scripts.

- Container login scripts are assigned to container objects (Organization or Organizational Unit objects).
- Profile login scripts are assigned to Profile objects. For a User object to use a profile login script, you must select that User object and assign it to the profile.
- User login scripts are assigned to User objects.

All three scripts use the same conventions, commands, and variables.

### **Login script commands**

Some of the commands you can use in login scripts are listed in the following table. (For a list of all login script commands, and a complete description of each, see "Login Script Commands and Variables" in *Supervising the Network*.)

**Table 1-4 Selected login script commands**

<b>Login script command</b>	<b>Description</b>
ATTACH	Connects to bindery-based NetWare servers (NetWare 2.x or 3.x) or to NetWare 4 servers using bindery services.
COMSPEC	If users run DOS from the network, this specifies the directory where the DOS command processor (COMMAND.COM) is loaded.
EXIT	Terminates execution of the LOGIN utility and executes an external program.
FIRE PHASERS	Emits a phaser sound when certain conditions exist.
IF...THEN	Performs an action only under certain conditions.
MAP	Maps drives and search drives to network directories.
PAUSE	Creates a pause in the execution of the login script.
SET	Sets a DOS environment variable.
WRITE	Displays messages on the workstation screen when a user logs in.

**Identifier variables**

With many login script commands, you can take advantage of identifier variables to make your login script more efficient and flexible.

Identifier variables allow you to enter a variable (such as LOGIN\_NAME) in a login script command, rather than a specific name (such as RICHARD). By using the variable, you can make the login script command applicable to many users.

When the login script executes, it substitutes real values for the identifier variables. Therefore, when Richard logs in, the command

```
WRITE "Hello, " ; LOGIN_NAME
```

displays the following message on Richard's workstation screen:

```
Hello, Richard
```

In the above example, when Richard logged in, the name he entered was substituted for the LOGIN\_NAME variable.

The following table lists identifier variables you can use in login scripts.

**Table 1-5 Identifier variables**

Category	Identifier variable	Function
Date	DAY DAY_OF_WEEK MONTH MONTH_NAME NDAY_OF_WEEK SHORT_YEAR YEAR	Day number (01 through 31). Day of week (Monday, Tuesday, etc.). Month number (01 through 12). Month name (January, February, etc.). Weekday number (1 through 7, with 1=Sunday). Last two digits of year (93, 94, 95, etc.). All four digits of year (1993, 1994, 1995, etc.).
Time	AM_PM GREETING_TIME HOUR HOUR2 MINUTE SECOND	Day or night (am or pm). Time of day (morning, afternoon, or evening). Hour (12-hour scale; 1 through 12). Hour (24-hour scale; 00 through 23). Minute (00 through 59). Second (00 through 59).
DOS Environment	<variable>	Any DOS environment variable can be used in angle brackets (<path>, etc.). To use a DOS environment variable in a MAP command, add a percent sign (%) in front of the variable, such as MAP S16:=%<path>.
Network	FILE_SERVER NETWORK_ADDRESS	NetWare server name. IPX external network number (8-digit hexadecimal number).
User	FULL_NAME LAST_NAME	User's complete name in the Directory context, or full name in bindery-based NetWare. User's last name (surname) in NDS, or full name in bindery-based NetWare.

**Table 1-5 Identifier variables**

<b>Category</b>	<b>Identifier variable</b>	<b>Function</b>
	LOGIN_NAME	User's unique login name. (Long names are truncated to eight characters.)
	MEMBER OF "group"	Group object that the user is assigned to.
	NOT MEMBER OF "group"	Group object that the user is not assigned to
	PASSWORD_EXPIRES	Number of days before password expires.
	USER_ID	Number assigned to each user.
Workstation	MACHINE	Type of computer (for example, IBM_PC). See your DOS manual for more information.
	OS	Type of DOS on the workstation (for example, DRDOS or MSDOS.).
	OS_VERSION	Version of DOS on the workstation (for example, 3.30).
	P_STATION	Workstation's node address (12-digit hex).
	SHELL_TYPE	Version of the workstation's DOS shell (for example, 1.02). Supports NetWare 2.x and 3.x shells and NetWare 4 Requester for DOS.
	SMACHINE	Short machine name (for example, IBM).
	STATION	Work station's connection number.
Miscellaneous	ACCESS_SERVER	Shows whether the access server is functional (TRUE=functional, FALSE=not functional).
	ERROR_LEVEL	An error number (0=No errors).
	%n	Replaced by parameters the user enters at the command line with the LOGIN utility.

**Table 1-5 Identifier variables**

Category	Identifier variable	Function
Object properties	property name	You can use any property of NDS objects as a variable. Use the property's name just as you do any other identifier variable. If the property name includes a space, enclose the name in quotation marks.

**Examples of login scripts**

The following examples of login scripts may help you plan your own container, profile, and user login scripts. Each example script is shown in a table. The left column shows the commands in the script. The right column explains the command.

Container login script. The container login script should contain as much information as possible that will apply to all users.

**Table 1-6 Sample container login script**

Login script commands	Purpose
MAP DISPLAY OFF	Prevents map commands from displaying on the screen.
MAP ERRORS OFF	Prevents mapping errors from displaying on the screen.
MAP *1:=SYS:	Maps the first drive to volume SYS:.
MAP *1:=SYS:%LOGIN_NAME	Maps the first drive to the user's home directory. If the user has no home directory, the first drive is still mapped to SYS:.
IF "%1"="ADMIN" THEN MAP *1:=SYS:SYSTEM	If the login name is ADMIN, it maps the first drive to SYS:SYSTEM instead of to the user's home directory.

**Table 1-6**      **Sample container login script**

Login script commands	Purpose
<pre>IF OS2 THEN MAP P:=SYS:PUBLIC ELSE MAP INS S1:=SYS:PUBLIC MAP INS S2:=SYS:PUBLIC\%MACHINE%\%OS\ %OS_VERSION END</pre>	<p>The first search drive is mapped to SYS:PUBLIC, where DOS-based NetWare utilities are stored. The second search drive is mapped to the directory where DOS is stored.</p> <p>For example, if all stations use DOS, use the following two commands instead of the IF...THEN command:</p> <pre>MAP INS S1:=SYS:PUBLIC MAP INS S2:=SYS:PUBLIC\ %MACHINE%\%OS%\%OS_VERSION</pre>
<pre>IF MEMBER OF "WIN31" THEN MAP INS *2:=SYS:USERS\%LOGIN_NAME\WIN31 MAP INS S16:=SYS:APPS\WINAPPS\WIN31 SET TEMP = "P:\USERS\%LOGIN_NAME\WIN31\TEMP" END</pre>	<p>If the user who logs in is a member of Group object WIN31, the next available drive is mapped to that user's Windows directory. Then the next available search drive is mapped to the Windows directory for the WIN31 group. Finally, the Windows TEMP directory is set to a subdirectory of the user's Windows directory.</p>
<pre>MAP INS S16:=VOL1:APPL\WP</pre>	<p>Maps the next available search drive to the directory that contains WordPerfect. Extra setup is required if users use different search drives for WordPerfect.</p>
<pre>MAP INS S16:=VOL1:APPL\LOTUS</pre>	<p>Maps the next available search drive to the directory that contains Lotus.</p>
<pre>MAP INS S16:=SYS:EMAIL</pre>	<p>Maps the next available search drive to the E-mail directory. Note: Some mail programs require all users to use the same search drive.</p>

**Table 1-6**                    **Sample container login script**

<b>Login script commands</b>	<b>Purpose</b>
MAP O:=SYS:DOC	Maps drive O: to a directory necessary for running the electronic NetWare documentation (DynaText).
IF MEMBER OF "MANAGERS" THEN MAP *3:=VOL1:PROJECTS\REPORTS END	If the user belongs to the MANAGERS Group object, the script maps the third network drive to the REPORTS directory.
IF MEMBER OF "TESTERS" THEN MAP *4:=INPUT:STATUS\UPDATES END	If the user belongs to the TESTERS Group object, the script maps the fourth network drive to the UPDATES directory.
COMSPEC = S2:COMMAND.COM	Sets COMSPEC to the DOS command processor, located in the DOS directory (in the second search drive).
SET PROMPT = "\$P\$G"	Sets the prompt to display the user's current directory path, followed by the > symbol.
MAP DISPLAY ON	Allows map commands to display.
MAP	Displays a list of all drive mappings.
WRITE	Displays a blank line between the list of mappings and following lines.
WRITE "Good %GREETING_TIME, %FULL_NAME."	Displays a greeting to the user. Example: "Good morning, MARY.SALES.HEWLETT-PACKARD."
WRITE "Your password expires in %PASSWORD_EXPIRES days."	Displays a message indicating the number of days before the user's password expires.
FIRE PHASERS 3 TIMES	Makes the phaser sound occur three times, to tell the user that the login process is complete.

## L

Profile login script. If you have groups of users with identical login script needs, you can create a Profile object, then create a login script for the Profile object. Then you can assign each user to be a member of that Profile object.

The following is an example of a profile login script you might create for users in the ACCOUNTING Profile object. The ACCOUNTING profile login script would execute after the container login script.

**Table 1-7**                    **Sample profile login script**

Login script commands	Purpose
MAP DISPLAY OFF	Prevents map commands from displaying on the screen as they are assigned.
MAP ERRORS OFF	Prevents mapping errors from displaying on the screen.
MAP INS S16:=VOL1:APPL\DB	Maps the first available search drive (after those assigned in the container login script) to the directory that contains the database program.
MAP *5:=VOL1:ACCOUNTS\NEW	Maps the fifth network drive (after those assigned in the container login script) to the NEW subdirectory.
MAP *6:=VOL1:ACCOUNTS\RECORDS	Maps the sixth network drive (after those assigned in the container login script) to the RECORDS subdirectory.
#WSUPDATE S1:IPXODI.COM /LOCAL	Executes WSUPDATE, which updates the IPXODI.COM file on the user's workstation with a new version of the file located in the first search drive.
MAP DISPLAY ON	Allows map commands to display.
MAP	Displays a list of all drive mappings.
WRITE	Displays a blank line between the list of mappings and following lines.



**Table 1-7**                    **Sample profile login script**

<b>Login script commands</b>	<b>Purpose</b>
IF DAY_OF_WEEK = "FRIDAY" THEN WRITE "Weekly progress report is due today." FIRE PHASERS 2 TIMES END	On Fridays, the phaser sound occurs twice to alert the user while the message "Weekly progress report is due today" displays on the screen.
PCCOMPATIBLE EXIT "NMENU WORK"	Stops the profile login script and sends the user into a menu program called WORK.  EXIT also prevents user login scripts from executing. If you want a user login script to execute after the profile script, put these lines at the end of the user login script instead.  DOS workstations with the machine name IBM_PC do not need the PCCOMPATIBLE line.

User login script. The following is an example of a user login script for MARY. The user login script executes after container and profile login scripts.

**Table 1-8**                    **Sample user login script**

<b>Login script commands</b>	<b>Purpose</b>
MAP DISPLAY OFF	Prevents map commands from displaying on the screen as they are assigned.
MAP ERRORS OFF	Prevents mapping errors from displaying on the screen.
MAP *7:=VOL1:MARY\PROJECTS\RESEARCH	Maps Mary's seventh network drive (after those assigned in the container and profile scripts) to the RESEARCH subdirectory in her home directory.

**Table 1-8**      **Sample user login script**

<b>Login script commands</b>	<b>Purpose</b>
MAP *8:=VOL1:FORMS	Maps Mary's eighth network drive (after those assigned in the container and profile scripts) to the FORMS directory.
REM Mary needs access to FORMS while she's on the REM troubleshooting team. Remove this drive mapping REM when she's reassigned.	This remark is a reminder to the person who created the login script. This remark won't display on the user's screen.  (Because the remark is several lines long, each line starts with the keyword REM.)
SET WP="/u-mjr/b-5"	Sets Mary's environment variables for WordPerfect (DOS version).
SET USR="mrichard"	Sets Mary's user name (mrichard) for the electronic mail program.
#CAPTURE Q=FAST_Q NB TI=10 NFF	Executes the CAPTURE utility so Mary can print from non-network applications.
PCCOMPATIBLE EXIT "NMENU TRAINING"	Stops the user login script and sends the user into a menu program called TRAINING.  DOS workstations with the machine name IBM_PC don't need the PCCOMPATIBLE line.

For information about creating, modifying, and copying login scripts, see "Creating, Modifying, and Copying Login Scripts" in *Supervising the Network*.

See also "Drive mapping."

**Logout**

A procedure that breaks the network connection and deletes drives mapped to the network.

If you log out without specifying a NetWare server name in the LOGOUT command, the station connections and drives mapped to all servers are terminated.

To log out of one server and remain logged in to the other servers, specify the server name in the LOGOUT command. To permit access to NetWare utilities, at least one of the remaining drives must be mapped to the PUBLIC directory of a NetWare server that you are still logged in to.

Related utilities: “LOGOUT”; “NETADMIN” (*Utilities Reference*).

**Long filename**

A filename that exceeds the 12-character limit (eight characters, a dot (.) delimiter, and a three-character extension) set by DOS.

In UNIX, filenames can be up to 1024 characters long. Embedded spaces and UNIX shell metacharacters (?, \*, [, ], and [!]) should be avoided.

UNIX filenames are case sensitive. For example, Data\_1 and data\_1 can exist as two separate files in the same directory.

In OS/2, a High Performance File system (HPFS), convention allows long, free-form filenames. OS/2 1.2 filenames can contain embedded spaces, mixed case, and multiple dot delimiters.

An HPFS volume in OS/2 allows filenames of up to 255 characters and path name components up to 260 characters.

**Long machine type**

A six-letter name representing a DOS workstation brand.

Use the long machine type in container login scripts (using the MACHINE identifier variable) to automatically map a drive to the correct version of DOS assigned to the station.

IBM computers use the long machine type IBM\_PC. If the station is not an IBM computer, create a long machine type for the station in a NET.CFG file.

**L**

Use the six-letter name for the long machine type as the subdirectory name when you use more than one brand of workstation. Example: COMPAQ.

Use the same six-letter name for DOS directories that you use for the long machine type.

If you use more than one version of DOS, you must have separate subdirectories for each DOS version used on each machine type.

See also: “DOS version”; ; “Login scripts”“Short machine type.”

**LPT1**

The primary parallel printer port of a personal computer.

See also: “Parallel port.”

**LSL (Link Support Layer)**

See “Link Support Layer (LSL).”

---

## M

### **MAIL directory**

The SYS:MAIL directory, created during network installation, used by mail programs that are compatible with NetWare.

In previous versions of NetWare, the MAIL directory held user login scripts. When you upgrade to NetWare 4, existing users still have subdirectories in the MAIL directory, but their login scripts become a property of the new User object.

New users that you create under NetWare 4 won't have subdirectories in the MAIL directory.

See also: "File Systems, NetWare Services"; "LOGIN directory"; "PUBLIC directory"; "SYSTEM directory."

### **Map**

For DOS clients, the MAP command is used to assign a drive letter to a directory path on a volume.

Example: If you map drive F: to the directory SYS:ACCTS/RECEIVE, you access that directory every time you change to drive F:.

See also: "Drive mapping."

### **Memory, DOS management**

The internal dynamic storage of a computer that can be addressed by the computer's operating system; referred to frequently as RAM (Random Access Memory).

Memory accepts and holds binary data. To be effective, a computer must store the data that will be operated on as well as the program that directs the operations to be performed.

Memory stores information and rapidly accesses any part of the information upon request.

### **Memory allocation**

The process of reserving specific memory locations in RAM for processes, instructions, and data.

When a computer system is installed, the installer may allocate memory for items such as disk caches, RAM disks, extended memory, and expanded memory.

Operating systems and application programs allocate memory to meet their requirements, but they can only use memory actually available to them.

Memory can be reallocated between resources to optimize performance. The proper memory allocation mix depends on the applications that are run.

For example, a large disk cache that speeds up one application may slow down others because there is less available conventional memory.

### **Memory board**

An add-on board that increases the amount of RAM within a personal computer.

See also: “Memory, DOS management”; “RAM.”

### **Message packet**

A unit of information used in network communication.

See also: “Packet.”

### **Message system**

A communications protocol that runs on top of IPX. It provides an engine (process) that allows a node on the network to send messages to other nodes.

A set of APIs (application programming interfaces) gives programs access to the message system.

See also: “IPX.”

**Migrated attribute**

A NetWare status flag that indicates a file is migrated.

NetWare Services does not support data migration or use this attribute.

See also: "Attributes."

**MLID**

See "Multiple Link Interface Driver (MLID)."

**Modify bit**

A bit set by NetWare, when a file is changed, to indicate that data has been modified.

**Modify right**

A directory or file right that grants the right to change the attributes or name of a directory or file.

See also: "Rights."

**Multiple-byte character**

A single character made up of more than one byte.

One byte allows 256 different characters. Since the number of ASCII characters equals 256, a computer can handle each ASCII character with one byte.

Asian character sets, however, include more than 256 characters; in this case, a computer must use two bytes for each character.

**Multiple Link Interface Driver (MLID)**

A device driver written to the ODI specification that handles the sending and receiving of packets to and from a physical or logical LAN medium.

See also: "Open Data-Link Interface (ODI)."

### **Multiple name space support**

The method that allows workstations running different operating systems to use their native naming conventions. The file system can present any given file using the client operating system's naming conventions.

The filename conversions require the following steps:

- Character mappings
- Collision detection
- Binding

#### **Character Mappings**

Characters that are legal in filenames of one client operating system may not be legal in another. The allowable length of filenames may vary among client file systems. Therefore filenames must be translated or "mapped" to appear properly on each client.

Mapping may

- Match a character with the same character when it is supported by the other file system's naming conventions
- Replace a character not supported by the other file system's naming conventions with a character that is supported
- Omit characters not supported by the other file system or that exceed the length of allowable names in the other file system

Character mapping results in a proposed filename that meets the naming conventions of the other file system.

#### **Collision Detection**

The collision detection routines check the proposed name with all filenames in the directory for that name space. If a match is found, a numbered suffix is added to the filename just before the extension.

A directory might contain the HP-UX file fredfilenew. If it were mapped to a DOS filename the proposed name would be FREDFILE. If collision detection found that there was already a DOS file called FREDFILE, the file would instead become FREDFIL0.



**Binding**

Binding is the action of generating valid filenames for the client types that did not create a file. In NPFS, the binding occurs when the volume is configured for the name space and mounted and as files are created.

See also: “Name space.”

**Multiserver network**

A single network that has two or more NetWare servers operating.

On a multiserver network, users can access files from any NetWare server they have access rights to.

A multiserver network isn't the same as an internetwork, where two or more networks are linked through a router.

See also: “Internetwork”; “Network numbering.”

**N**

**Name context**

The position of an object in the Directory tree.

See also: "Context."

**Name space**

A feature that allows you to store non-UNIX files on a NetWare Services server. Files appear in native mode to users at different workstations.

DOS and UNIX name spaces are always enabled on a volume and cannot be disabled. OS/2 name spaces can be added and removed.

NetWare Services supports the following lengths for filenames:

DOS8.3

UNIX255

See also: "Multiple name space support."

**NCP**

See "NetWare Core Protocol (NCP)."

**ncp\_engine**

A HP-UX daemon that is spawned by the NetWare daemon and runs as a root user. The NetWare daemon registers the ncp\_engine with NEMUX.

The ncp\_engine receives and processes NCP requests. It acts as the interface to the HP-UX file system to process the NCP requests and formulate responses to the NCP requests.

The HP-UX network supervisor controls how many ncp\_engine daemons are created and can dynamically increase or decrease the number. NetWare Services needs at least two and the number it needs increase with the number of users.

See also: “NEMUX”; “NetWare Core Protocol (NCP)”; “NetWare daemon”; “Lock manager.”

### **NCP Packet Signature**

A NetWare security feature that protects servers and clients using NetWare Core Protocol (NCP) by preventing packet forgery.

Without NCP Packet Signature, a client could pose as a more privileged client by sending a forged NCP request to a NetWare server. By forging the proper NCP request packet, an intruder could gain rights to all network resources.

NCP Packet Signature prevents packet forgery by requiring the server and the client to “sign” each NCP packet. The packet signature changes with every packet. NCP packets with incorrect signatures are discarded without breaking the client’s connection to the server.

However, an alert message about the invalid packet is sent to the error log and the server console. The alert message contains the login name and the station address of the affected client.

If NCP Packet Signature is installed on the server and all of its clients, it is virtually impossible to forge a valid NCP packet.

### **NDS**

See “NetWare Directory Services (NDS).”

### **NEMUX**

See “NetWare Engine Multiplexor (NEMUX).”

### **NETBIOS.EXE**

NetWare’s NetBIOS emulator program that allows DOS clients to run applications written for peer-to-peer communication or distributed processing.

The INT2F.COM file is used with NETBIOS.EXE.

### **NET.CFG**

A workstation boot file, similar to DOS CONFIG.SYS, that contains configuration values that are read and interpreted when your workstation starts up.

These configuration values adjust the operating parameters of the NetWare DOS Requester, IPX, and other workstation software.

Applications such as database, multitasking, or NetBIOS (involved in peer-to-peer communications or distributed processing) may require parameter values different from the default values to function properly on the network.

Some network problems such as printing and file retrieval might also be solved by adjusting workstation parameters.

NET.CFG is created with an ASCII text editor and needs to be included on the workstation boot diskette with other boot files. NET.CFG replaces SHELL.CFG, used in earlier NetWare versions.

### **NetWare daemon**

The process that starts and initializes NetWare Services.

The NetWare daemon

- Runs as a root user
- Links the protocol stacks to NEMUX by building the stream
- Initializes the NetWare Services environment by such operations as allocation of shared memory
- Starts NetWare Services ancillary processes
- Starts engine processes, in conjunction with NEMUX, and informs NEMUX which processes started are engines.
- Performs various types of asynchronous event handling, such as watchdog timeouts.
- Communicates stream events between the SNMP Agent and the protocol stacks

**NetWare Core Protocol (NCP)**

NetWare Core Protocol. Procedures that a server's NetWare operating system follows to accept and respond to workstation requests.

The process of requesting service from a NetWare server begins in the workstation's RAM where the NetWare DOS Requester or NetWare .

The Requester then hands the requests to the station's IPX communication protocol. IPX transmits the request to the server after attaching a header designating the source and destination.

Upon receiving the request, the server removes the IPX header and reads the request.

Because the NetWare Requester formed the request using the exact guidelines of a specific service protocol, the server handles the request according to the protocol rules, resulting in a proper response.

NetWare Core Protocols exist for every service a station might request from a server.

Common requests handled by NCP include creating or destroying a service connection, manipulating directories and files, opening semaphores, altering the Directory, and printing.

See also: "Communication protocol"; "IPX."

**NetWare Diagnostic Daemon (NWDIAGD)**

Provides IPX/SPX diagnostic services across an internetwork.

NWDIAG receives and answers diagnostics requests about a network in accordance with the network's configuration. The diagnostic services enable applications to

- Identify internetwork nodes
- Build an internetwork map
- Query one or more nodes on an internetwork
- Identify software components on different nodes
- Provide information about those components
- Perform point-to-point packet transmission efficiency tests

If a query is made that does not make sense in a particular NetWare Services environment, NWDIAG responds that the query is unsupported. For example, NetWare shell function calls would not be supported.

The NWDIAG startup is configurable using the nwcm utility.

See also: "nwcm."

### **NetWare Directory database**

The database (commonly referred to as the Directory) that organizes NetWare Directory Services objects in a hierarchical tree structure called the Directory tree.

See also: "NetWare Directory Services (NDS)."

### **NetWare Directory Services (NDS)**

A global, distributed, replicated database built into NetWare Services that maintains information about, and provides access to, every resource on the network.

NetWare Directory Services treats all network resources (users, groups, printers, volumes, computers, and so forth) as objects in a distributed database known as the NetWare Directory database (also referred to as the Directory).

The NetWare Directory database organizes objects, independent of their physical location, in a hierarchical tree structure called the Directory tree.

Users and network supervisors can access any network service without having to know the physical location of the server that stores the service.

NetWare Directory Services makes it possible to integrate a diverse network of resources into a single, easy-to-use environment.

The Directory replaces the bindery, which served as the system database for previous releases of NetWare. While the bindery supports the operation of a single NetWare server, NetWare Directory Services supports an entire network of servers.

So, instead of storing all information on one server, which can be a single point of failure, information is distributed over a global database.

Compatibility with previous versions of NetWare is provided through bindery services.

NetWare Directory Services helps you manage Directory resources such as NetWare servers and volumes, but it does not provide control over the file system (files and file directories). Graphical and text utilities are available to help you control the file system.

#### **Accessing NetWare Directory Services**

Instead of logging in or attaching to individual servers, NetWare Directory Services users log in to the network.

When a user accesses resources on the network, background authentication processes verify that the user has rights to use those resources.

Authentication allows a user (who has logged in) to access any server, volume, printer, etc., that the user has rights to. User trustee rights restrict the user's access within the network.

See also: "Authentication"; "Login scripts."

#### **Objects**

A NetWare Directory Services object consists of categories of information, called properties, and the data in those properties. The information is stored in the NetWare Directory database.

Some objects represent physical entities. For example, a User object represents a user, a Printer object represents a printer, etc.

Some objects represent logical entities, such as groups and print queues.

Other objects, such as the Organizational Unit object, help you organize and manage objects.

Remember: an NDS object is a structure where information is stored. It isn't the entity it represents.

For example, a Printer object stores information about a single specific printer and helps manage how the printer is used, but it isn't the physical device itself.

See also: "Object."

### Directory tree

NetWare Directory Services operates in a logical organization called the Directory tree. It is called a Directory tree because objects are stored in a hierarchical tree structure, starting with the root object and branching out.

Two types of objects make up the Directory tree: container objects and leaf objects.

A branch of the Directory tree consists of a container object and all the objects it holds, which can include other container objects.

Leaf objects are at the ends of branches and don't contain any other objects.

The following figure shows how container objects and leaf objects make up the Directory tree.

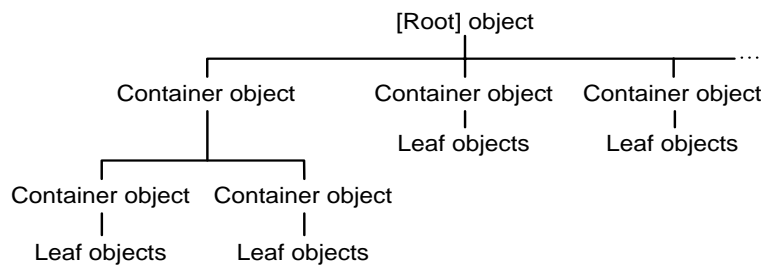


Figure 1-28

### Objects in a Directory tree

In a Directory tree, you can place container objects and leaf objects in different configurations, according to what best fits your company's needs.

The following figure shows some possible configurations for a Directory tree. (Country, Organization, and Organizational Unit are the three types of container objects.)



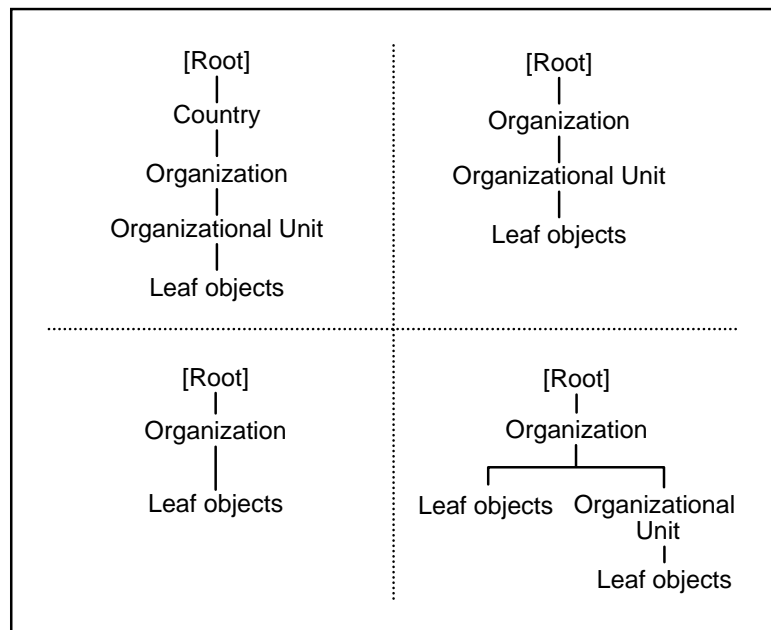


Figure 1-29

**Possible Directory tree configurations**

See also: “Directory tree”; “Object.”

**Object names**

The path from an object to the root of the Directory tree forms the object’s complete name (sometimes called the distinguished name), which is a unique name.

Most leaf objects have a common name. For User objects, the common name is their login name, displayed in the Directory tree.

Other leaf objects also have common names that are displayed in the Directory tree, such as a Printer object name or a NetWare Server object name. If you are referring to an object in the same container object as your User object, then you only need to refer to that object by its common name, instead of by its complete name.

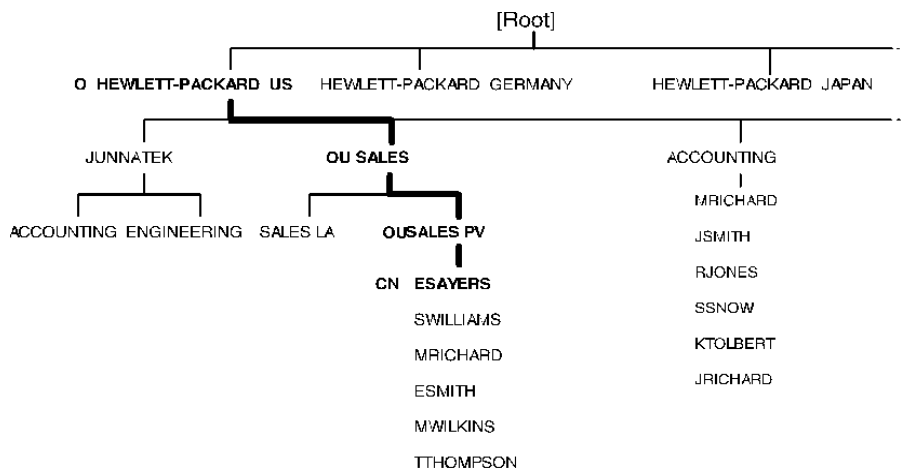
Container objects don’t have common names. They are referred to by their Organizational Unit name, Organization name, or Country name.

**N**

An object's complete name consists of its common name (if it has one), followed by a period (.); then the name of the container object, also followed by a period, and on up through succeeding container object names to the root of the tree.

For example, in the following figure, for User object ESAYERS (the common name), the complete name would be

**ESAYERS.SALES PV.SALES.HEWLETT-PACKARD US**



**Figure 1-30**

**Complete and common names**

When you

- Move from one container object to another
- Include the Country container object in your Directory tree

you must include the name type of the object in the complete name of the object.

Under either of these conditions, you would express ESAYERS as

**CN=ESAYERS.OU=SALES PV.OU=SALES.O=HEWLETT-PACKARD US**

where CN specifies the common name, OU specifies the Organizational Unit name, and O specifies the Organization name.

When querying the Directory, you can supply the complete name of an object; then receive information that describes that object. Or, you can supply a property value of an object and receive a list of objects that have that value.

For example, to find all users with a last name of Smith, then “Smith” is the value you want to find in the LAST NAME property of User objects.

### **Object context**

NetWare Directory Services allows you to refer to objects according to their positions within a tree. When you add an object (such as a server or user), you place that object in a container object in the Directory tree.

The position of the object within its container is its context. For example, in the previous figure, the context for the User object ESAYERS is SALES PV.SALES.HEWLETT-PACKARD US.

When you move from one container object to another, you change contexts. Whenever you change contexts, indicate the complete name of the object you are changing context to.

If you are referring to an object that is in the same container object as your User object, then you only need to refer to that object by its common name, instead of by its complete name.

For example, in the previous figure, if User object ESAYERS, located in SALES PV.SALES.HEWLETT-PACKARD US, wants information on User object ESMITH located in the same context, then ESAYERS need only refer to the User object as ESMITH.

### **User object ADMIN**

The first time the network supervisor logs in, he or she logs in as User ADMIN, created automatically during Directory Services installation.

When ADMIN is created, it is given a default trustee assignment to the root object. This assignment grants ADMIN all rights to all objects and all volumes (directories and files) in the entire Directory tree.

This means that ADMIN has rights to create and manage all objects in the tree.

**N**

ADMIN does not have significance like SUPERVISOR did in earlier versions of NetWare. It is only the first User object created and therefore must have the ability to create other objects.

As you create other User objects on the Directory tree, you can give some of them the Supervisor object right to create and manage other container objects and their leaf objects. Control of the network is as dispersed or centralized as you make it.

After you assign the Supervisor object right to the root object for another User object, you can rename ADMIN.

Part of the flexibility of NetWare Directory Services is the capability to have centralized or dispersed control of the network. Therefore, no single User object automatically has rights over any part of the network.

See also: "ADMIN object."

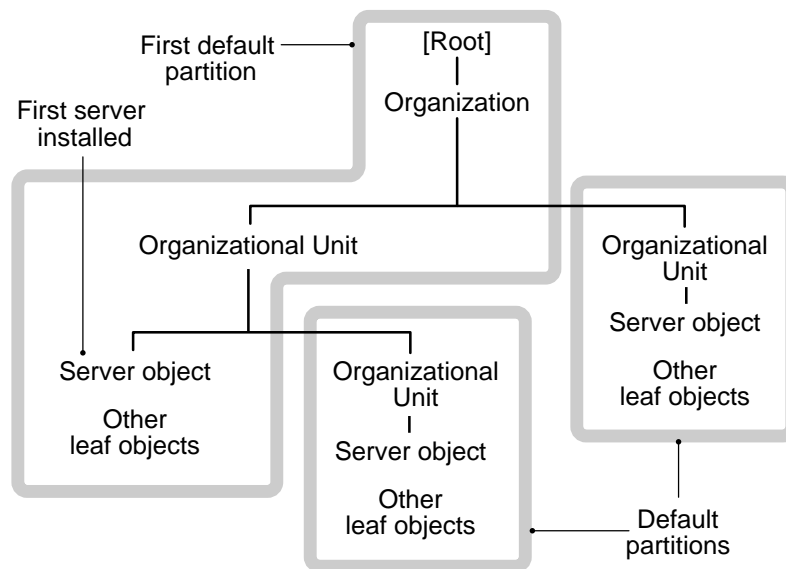
**Partitions**

To be more manageable, the NetWare Directory database is divided into smaller portions called partitions. Partitions are created by default when you install NetWare 4 on a server in a new context in the Directory tree.

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.

The root object (at the top of the tree) is included in the first partition created.

The following figure shows the default partition created for the first server installed and for all new Organizational Units created in which NetWare servers were installed.



**Figure 1-31 Example of default partitions**

The tree of partitions is transparent to Directory users (unless they are running Partition Manager); users usually see only a global tree of Directory objects.

To optimize access to different areas of the Directory, each partition can be replicated and stored at many locations.

Partition replication improves access and provides the Directory with fault tolerance. Since a partition can be replicated at several locations, damage to one of the replicas does not need to interrupt access to the partition information.

See also “Partition, Directory Services.”

### **Replicas**

For NetWare Directory Services to be distributed across a network, the database must be stored on many servers. Rather than have a copy of the whole database on each server, replicas of each partition are stored on many servers throughout the network.

A replica is a copy of a partition. You can create an unlimited number of replicas for each partition and store them on any server in the network.

Replicas serve two purposes:

- To eliminate any single point of failure.

For example, 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 that partition.

- To provide faster access to information for users across a WAN link.

For example, if users must use a WAN link to access information, decrease access time and network traffic by placing a replica containing the needed information on a server that users can access locally.

See also: “Replica.”

### **Bindery compatibility**

To provide compatibility with bindery-based versions of NetWare that may co-exist with NetWare Directory Services on the network, NetWare 4 features bindery services.

Objects in a bindery exist in a flat database instead of a hierarchical database like a Directory tree. Bindery services occur when NetWare Directory Services provides a flat structure for the objects within an Organization or Organizational Unit container object.

All objects within that container object can then be accessed both by NetWare Directory Services objects and by bindery-based clients and servers. Bindery services apply only to leaf objects in that Organizational Unit.

The Organizational Unit where bindery services are set is called the bindery context. You can change the bindery context by using the graphical “NetWare Setup” or the command line “nwcm” utility.

See also: “Bindery services.”

### **Time synchronization**

NetWare allows servers to synchronize their time with each other. NetWare Services gets its time from NTP, and when the NetWare Server is run in reference mode, it can advertise this clock value to the other NetWare servers on the network.

Time synchronization is critical to the operation of NetWare Directory Services because it establishes the order of events.

Whenever an event occurs in the Directory, such as when a password is changed, or an object is renamed, NetWare Directory Services assigns a time stamp.

A time stamp is a unique code that includes the time and identifies the event. The NetWare Directory Services event is assigned a time stamp so that the order in which replicas are updated is correct.

When you install NetWare Services, time synchronization is created by default, or you can customize synchronization by specifying a server as one of four types:

- Single Reference time server
- Reference time server
- Primary time server
- Secondary time server

Each time server performs a particular time synchronization function.

See also: “Time synchronization.”

### **NetWare Directory Services daemons**

HP-UX processes running in the background with little or no user input.

Because NetWare Services does not support NetWare loadable modules (NLMs), the function of some NLMs are performed by daemons. The daemons that perform NetWare Directory Services (NDS) functions are

- NetWare daemon, which installs and initializes NDS
- Server Advertiser Daemon, which keeps the server advertiser protocol (SAP)

**N**

table in the bindery partition current with all servers using SAP

- DSBackground daemon, which performs asynchronous background tasks, such as synchronization, for NDS
- DSJanitor daemon, which performs one of the asynchronous background tasks
- TimeSynch Daemon, which keeps the NetWare Services server's time synchronized with other NDS servers.

See also: "NetWare daemon"; "NetWare Directory Services (NDS)."

**NetWare DOS Requester**

The DOS client software portion of NetWare Services. The Requester can be called by applications or utilities in one of three ways:

- Before DOS, in the same way as the old NETX shell.
- By DOS, through the INT 2Fh redirector.
- Bypassing DOS, through a pipeline between the shell and post-DOS portions.

**VLMs**

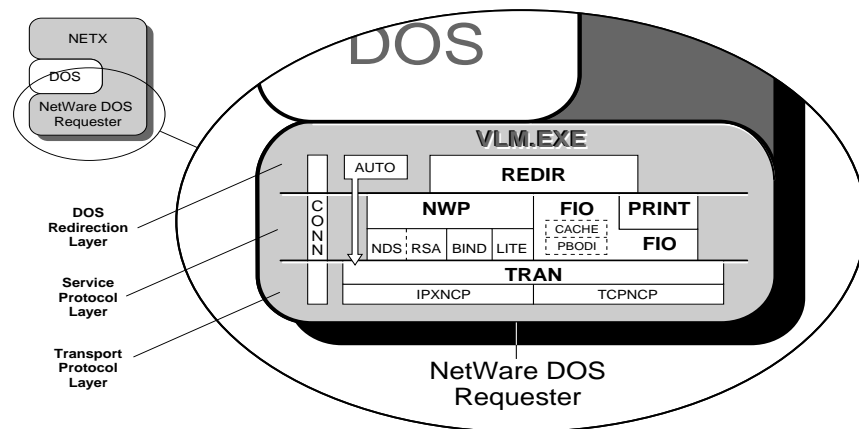
The NetWare DOS Requester is composed of a number of modules called Virtual Loadable Modules (VLMs). These VLMs are the key to the NetWare DOS Requester's modularity. (See "Virtual Loadable Module (VLM).")

The Requester contains categories of services and platforms in the following three layers:

- DOS Redirection Layer
- Service Protocol Layer
- Transport Protocol Layer

The following figure shows how these layers and modules fit together.





**Figure 1-32** NetWare DOS Requester layers and modules

### DOS redirection

The NetWare DOS Requester includes a redirector that, in contrast to the NetWare shell, is called by DOS.

Under the redirector, DOS makes specific requests for services from the redirector (such as for file and print services from the server) that DOS cannot provide. (These functions were previously performed by the NETX shell, without involving DOS.)

The NetWare DOS Requester also continues to provide network services for file and print redirection, as well as for connection maintenance and other NetWare-specific support.

For more information, see the NetWare Client for DOS and MS Windows User Guide and the NetWare Client for DOS and MS Windows Technical Reference.

### NetWare Engine Multiplexor (NEMUX)

An engine that connects the NetWare Services processes, which operate in HP-UX user space, with the NetWare protocol stack, which operates in the HP-UX kernel.

## N

The primary function of NEMUX is to schedule work for NetWare Services engines as messages arrive upstream and to route messages to the appropriate downstream protocol stack.

NEMUX passes messages between the NetWare Services processes and the NetWare protocol stack in NetWare core protocol (NCP).

NEMUX interacts with four types of NetWare Services processes—the NetWare daemon, engine processes, ancillary processes, and admin processes. All run as root users.

- NetWare daemon. Builds the stream that links NEMUX to the protocol stacks and starts the engine and ancillary processes.
- Engine processes. Daemons responsible for many of the NetWare Services functions. Clients typically interact with NetWare Services engine processes in a request-response scenario through NEMUX. Messages are received by NEMUX and passed to an engine which sends a response message to the NEMUX client  
  
NEMUX is not aware of the request response mechanism, but it does send information to the underlying protocol to allow it to properly manage the message context. NEMUX knows, however, when an engine has completed its work so that new work can be scheduled to that engine.
- Ancillary processes. Typically process work that would cause an engine process to block for an unknown period of time, or they perform background processing. They do not receive network messages but can receive messages through IPC (interprocess communications) from engines or other processes. Some ancillary processes can send messages to any client.
- Admin processes. Similar to ancillary processes but are not started by the NetWare daemon. They perform administrative functions that view or alter the state of NEMUX and NetWare Services processes.

### **NetWare inode**

A file—USInode—that stores all NetWare and name space information about data files in a NetWare volume on a HP-UX system. This information is specific to NetWare files and is not kept by the HP-UX operating system.

USInode contains such information as

- Filename
- Modification time
- Access time
- NetWare owner ID
- Name space information for DOS and OS/2

USInode can be located by looking at the NPFS volume control path in the voltab file.

See also: “Attributes”; “File Systems, NetWare Services”; “voltab.”

### **NetWare Loadable Module (NLM)**

A program in NetWare 4 that you can load and unload from server memory while the server is running. (Some NLMs are loaded automatically, because other NLMs can't run without them.)

NLMs link disk drivers, LAN drivers, name spaces, and other NetWare server management and enhancement utilities to the operating system.

NetWare Services does not run NLMs. Some NLMs, however, have been ported to NetWare Services as HP-UX daemons.

### **NetWare managed node**

A NetWare Services server that is available to network management consoles over IP/IPX. This server must have the management daemons enabled to register NetWare Services data to the Network management agent.

See also: “Network management”; “NetWare management agents.”

### **NetWare management agents**

The management agent for NetWare Services is the Simple Network Protocol Management (SNMP). It is a UNIX process that can gather data about the managed items in the definitions file.

## N

The SNMP management agent (daemon) communicates with many peer daemons. NetWare Services has two such peer daemons: nwumps and nwum.

The SNMP management agent can communicate with the management console using either IP or IPX.

See also: "Network management"; "NetWare managed node."

### **NetWare operating system**

The network operating system developed by Novell, Inc. NetWare runs on the server and provides several functions to the network and the applications running on it, including

- File and record locking
- Security
- Print spooling
- Interprocess communications

The NetWare operating system also determines performance and reliability of the network.

### **NetWare Protocol stack daemon (NPSD)**

A daemon that opens and links the drivers and modules in the NetWare Protocol stack. It also provides a clean shutdown of the stack.

The actions performed by the NPSD depend on nwcm configuration parameters.

The following scenario assumes all drivers and daemons are configured.

The startnps utility starts the NPSD and the protocol stack with these major steps:

- Reads the nwconfig file to determine which drivers it needs to open and link and

which daemons to spawn

- Opens the IPX driver
- Builds the lower multiplexor for IPX by
  - Configuring the LANs
  - Setting up Ethernet frame types
  - Initializing IPX with information from the nwconfig file
  - Setting up routing
  - Starting IPX
- Builds the upper multiplexor for IPX by linking the SPX and NetBIOS drivers
- Spawns daemons according to parameters in the nwconfig file. The first daemon spawned is SAPD followed by NWDIAG-IPX/SPX and NWUMPS
- Sets up the NetWare Virtual Terminal (NVT) according to the host machine's architecture

Once these steps are completed, the parent process exits, leaving NPSD as a daemon.

The stopnps utility initiates shut down, causing NPSD to:

- Send a “going down” message to SAPD, which then sends out a “services going down” packet
- Deconfigure NVT
- Send a “going down” message to RIP
- Bring down the protocol stack and associated daemons, causing “hang up” messages to be sent to all open sockets
- Exit

### **NetWare Server object**

A leaf object that represents a server running NetWare on your network.

The network address property identifies its location on the network. The NetWare Server object is referred to in several other objects to specify where to find items such as volumes.

## N

A NetWare Server object can represent a server running any version of NetWare.

See also: “Creating Leaf Objects”; “Cautions When Deleting Server Objects” (*Supervising the NetWork*); “Object.”

### **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.

A NetWare network consists of workstations, peripherals, and one or more NetWare servers.

NetWare network users can share the same files (both data and program files), send messages directly between workstations, and protect files with an extensive security system.

### **Network address**

A network number that uniquely identifies a network cable segment; usually referred to as the IPX external network number.

See “IPX external network number.”

### **Network backbone**

A cabling system that NetWare servers and routers are attached to. If your network has three or more NetWare servers, this may be an efficient way to improve network performance.

The central cable handles all network traffic, decreasing packet transmission time and traffic on the network.

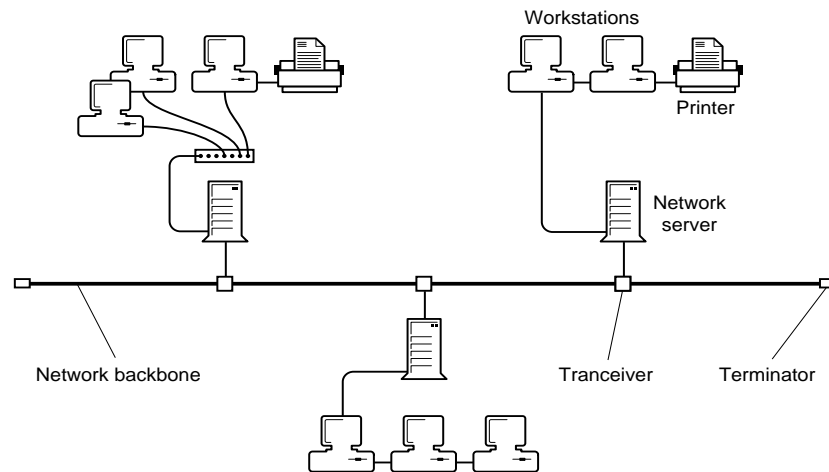


Figure 1-33

**Network backbone****Network board**

A circuit board installed in each workstation to allow stations to communicate with each other and with the NetWare server.

Some printers contain their own network board to allow them to attach directly to the network cabling.

**Network communication**

Data transmission between workstations. Requests for services and data pass from one workstation to another through a communication medium such as cabling.

**Network File System (NFS)**

NFS\* is a distributed UNIX file system that allows data to be shared by network users. Network users can share data regardless of operating system, workstation type, or protocols used.

NFS allows UNIX users to mount remote file systems so that they appear to be part of their local file system tree.

### **Network Interface Card (NIC)**

A circuit board installed in each workstation to allow stations to communicate with each other and with the NetWare server.

NetWare documentation uses the term “Network board” instead of “NIC.”

### **Network management**

Network management is the part of NetWare Services that offers client applications the capability to view and manage many of the resources on the network.

The network management on UNIX consists of

- Simple Network Management Protocol (SNMP)  
Maintains a list of requested peer daemons and knows which daemon or query for specific information.
- UNIX host peer daemon  
Manages the UNIX operating system. In HP-UX, for example, this is hostmibd
- Definition file  
Compiled from the MIB
- Network management console  
Presents management information to the user. Compiles MIBs using its own compiler.

NetWare Services adds

- NWUMPS daemon and associated MIB
- NWUM daemon and associated MIB
- Drivers

When the NetWare Services protocol stack is initialized, the NPS daemon checks the configuration parameters for network management. If the stack is configured for network management, the NPS daemon spawns the NWUMPS daemon, which allows the following to be managed: the protocol stack (IPX, RIP, SPX, SAP, Diagnostic).



When NetWare Services are initialized, the NetWare daemon checks the configuration parameters for network management. If NetWare Services are configured for network management, the NetWare daemon spawns the NWUM daemon, which allows the following to be managed: shared memory, NetWare queues, connection, NetWare Directory Services, the file system, NCP, and AFP.

The NWUM daemon uses shared memory, NCP calls, and ioctls to obtain statistics. The NWUMPS daemon uses shared memory and ioctls to obtain statistics.

### **Network node**

A personal computer or other device connected to a network by a network board and a communication medium.

A network node can be a server, workstation, router, printer, or fax machine.

### **Network number**

A number that uniquely identifies a network cable segment; usually referred to as the IPX external network number.

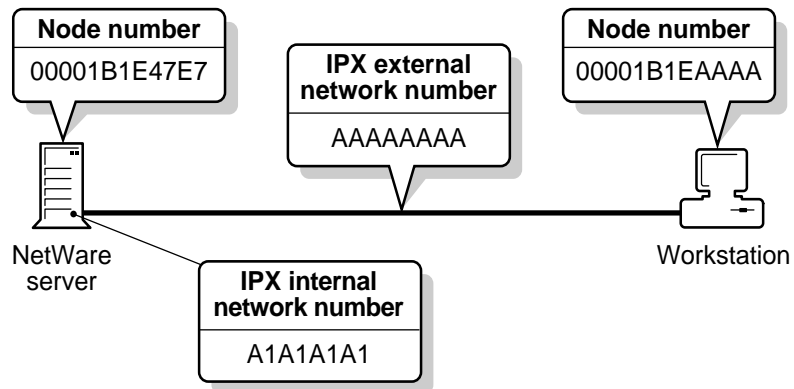
See "IPX external network number."

### **Network numbering**

The system of numbers that identifies servers, network boards, and cable segments. For IPX, these network numbers include the following:

- IPX external network number. A number that uniquely identifies a network cable segment.
- IPX internal network number. A number that identifies an individual NetWare 4 server.
- Node number. A number that identifies a network board (in a server, workstation, or router).

The relationship of these numbers is illustrated in the following figure.



**Figure 1-34** Network numbering

See also “IPX external network number”; “IPX internal network number”; “Node number.”

### **Network printer**

A printer shared in a network environment.

See also: “Printer.”

### **Network supervisor**

A generic term in NetWare 4 for the person responsible for configuring the NetWare 4 server, workstations, user access (security), printing, and so forth.

### **NETX**

A VLM (NETX.VLM) under the NetWare DOS Requester that provides backwards compatibility with NETX and other older versions of the NetWare shell.

See also “NetWare DOS Requester”; “Virtual Loadable Module (VLM).”

### **NFS**

See “Network File System (NFS).”

**NIC**

See “Network Interface Card (NIC).”

**NLM**

See “NetWare Loadable Module (NLM).”

**Node address**

A number that uniquely identifies a network board; usually referred to as the node number.

See “Node number.”

**Node number**

A number that uniquely identifies a network board.

Every node must have at least one network board, by which the node is connected to the network. Each network board must have a unique node number to distinguish it from all other network boards on that network.

Node numbers are assigned in several ways, depending on the network board type:

- Ethernet boards are factory-set (with no two Ethernet boards having the same number).
- ARCnet and Token-Ring board numbers are set with jumpers or switches.

Also known as station address, physical node address, and node address.

See also “IPX internetwork address.”

**Normal attribute**

A file system attribute that indicates that no NetWare attributes are set.

See “Attributes.”

**N**

**Novell Virtual Terminal (NVT)**

A method of allowing DOS clients to establish a virtual terminal sessions with a UNIX host running NetWare transport protocols.

See also: "NVT2."

**Novell Virtual Terminal 2 (NVT2)**

A method of allowing DOS clients to establish terminal sessions with a UNIX host running NetWare transport protocols. This feature gives clients access to UNIX applications.

NVT2 runs as a TSR program on the client. On the host, it uses UNIX features to listen for requests for connection from clients.

NVT2 is compatible with most third-party terminal emulation software that runs on PCs. The DOS clients load SPX, NVT2 and third-party terminal emulation software.

**NPS daemon**

See "NetWare Protocol stack daemon (NPSD)."

**NSE**

See "Network Support Encyclopedia Professional Volume (NSE Pro)."

**NVT**

See "Novell Virtual Terminal (NVT)."

**NVT2**

See "Novell Virtual Terminal 2 (NVT2)."

**nwcm**

The HP-UX command line interface to the NetWare run-time configuration library. You can use the `nwcm` (NetWare Configuration Manager) command to get or set values from the configuration database and get help messages for and descriptions of configuration parameters. It is also used to set parameters in the `nwconfig` file.

---

**NOTE:**

---

You can also use the graphical utility “NetWare Setup” to view or change `nwcm` values.

See also: “Configuration”; “`nwconfig`.”

**nwconfig**

A HP-UX file that contains configuration information for NetWare. The NetWare daemon reads `nwconfig` to establish the NetWare environment. The `nwconfig` file, usually found in the `/etc/netware4` directory, contains all NetWare Services parameters except volume parameters, which are stored in the configuration file `voltab` in the same directory.

Parameters in `nwconfig` control:

- Total connections allowed on the NetWare server
- Number of service engines the NetWare daemon spawns
- Whether to allow hybrid users
- SPX startup
- NetBIOS startup
- NVT2 startup
- External IPX network numbers for NetWare server boards
- Internal network number
- Loading the Ethernet drivers

The parameters in `nwconfig` are set using the NetWare configuration manager (`nwcm`) utility.

See also “Configuration”; “`nwcm`.”

NetWare Glossary

**N**

**NWDIAGD**

See “NetWare Diagnostic Daemon (NWDIAGD).”

---

## O

### **Object**

In NetWare Directory Services, a structure that stores information about a network resource (a user, group, printer, volume, etc.).

An object consists of categories of information, called properties, and the data in those properties. The information is stored in the NetWare Directory database.

Some objects represent physical entities. For example, a User object represents a user and a Printer object represents a printer.

Some objects represent logical entities, such as groups and print queues. Other objects, such as the Organizational Unit object, help you organize and manage objects.

Remember that an object is a structure where information about the entity is stored—it isn't the actual entity.

For example, a Printer object stores information about a printer and helps manage how the printer is used, but it isn't the printer itself.

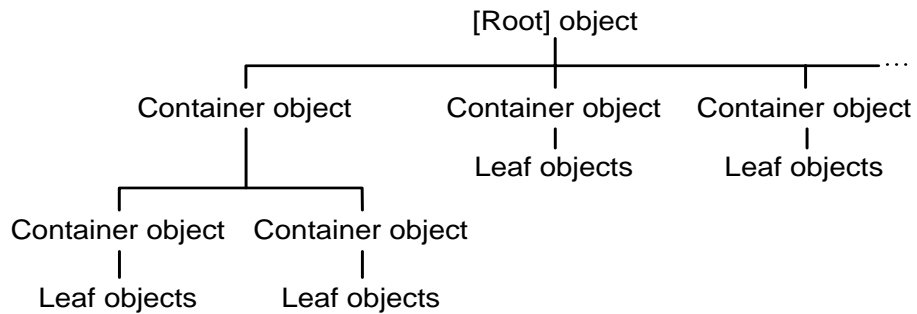
### **Objects and the Directory tree**

Two types of objects make up the Directory tree: container objects and leaf objects.

A branch of the Directory tree consists of a container object and all the objects it holds, which can include other container objects.

Leaf objects are at the ends of branches and don't contain other objects.

The following figure shows how container objects and leaf objects make up the Directory tree.



**Figure 1-35**      **Objects in a Directory tree**

Container objects. Container objects hold, or contain, other objects. Container objects are used as a way to logically organize all other objects in the Directory tree.

Container objects are like directories in a file system in that they group related information together. A container object is called a parent object if it has objects in it.

Types of container objects. There are three types of container objects, described in the following table.

**Table 1-9**      **Types of container objects**

Container object	Abbreviation	Description
Country	C	<p>Designates the countries where your network resides and organizes other Directory objects with the country.</p> <p>For example, you could use a Country object for the country where your organization headquarters reside or, if you have a multinational network, for each country that is a part of your network.</p> <p>To use a Country object, create it at installation. The Country object is not part of the default NetWare server installation. Using a Country object isn't required for interoperability with other X.500-compliant directory services.</p>



**Table 1-9**                    **Types of container objects**

<b>Container object</b>	<b>Abbreviation</b>	<b>Description</b>
Organization	O	<p>A level below the Country object (unless you don't use the Country object), the Organization object helps you organize other objects in the Directory and allows you to set template information for users created in this container.</p> <p>For example, you could use an Organization object to designate a company, or a university with various departments, or a department with several project teams.</p>
Organizational Unit	OU	<p>A level below the Organization object, the Organizational Unit object helps you to further organize other objects in the Directory and also allows you to set template information for users created in this container.</p> <p>For example, you could use an Organizational Unit object to designate a division, a business unit, a project team, or a college or department within a university.</p>

Country objects can contain Organization objects or Alias objects (a leaf object, described below).

Organization and Organizational Unit objects can contain Organizational Unit objects or leaf objects (described below).

#### **Leaf objects**

Leaf objects don't contain other objects. They represent network resources, such as users, computers, printers, and lists. The following table lists and describes leaf objects.

**Table 1-10**      **Types of leaf objects**

<b>Leaf object</b>	<b>Description</b>
Alias	<p>Points to the original location of an object in the Directory. Aliases can make NetWare Directory Services easier to use. Any Directory object located in one place in the Directory can also appear to be in another place in the Directory by using aliases.</p> <p>When you create an alias, name it in a way that you can recognize it as an alias. The name is the only way you can recognize it as an alias once it is created.</p> <p>When you add aliases to a list—for example, to add an alias of a user to a group, the name of the User object appears in the list, not the alias that points to the user.</p> <p>To access the alias and the properties of the object it refers to, you need the Read right to the alias name and the Read right to the properties of the object it refers to.</p> <p>You can set an option in NETADMIN to see the alias as a reference to another object, and then assign rights or modify the properties of the actual object.</p> <p>If you don't set the option to see aliases as references, you cannot work with the properties or rights of the object the alias refers to.</p>
Bindery	Represents an object placed in the Directory tree by an upgrade or migration utility, but that NetWare Directory Services can't identify. This object provides backward compatibility for bindery-oriented utilities.
Bindery Queue	Represents a queue placed in the Directory tree by an upgrade or migration utility, but that NetWare Directory Services can't identify. This object provides backward compatibility for bindery-oriented utilities.
Computer	<p>Represents a computer on the network.</p> <p>In the Computer object's properties, you can store information such as the computer's serial number or the person the computer is assigned to.</p>

**Table 1-10**      **Types of leaf objects**

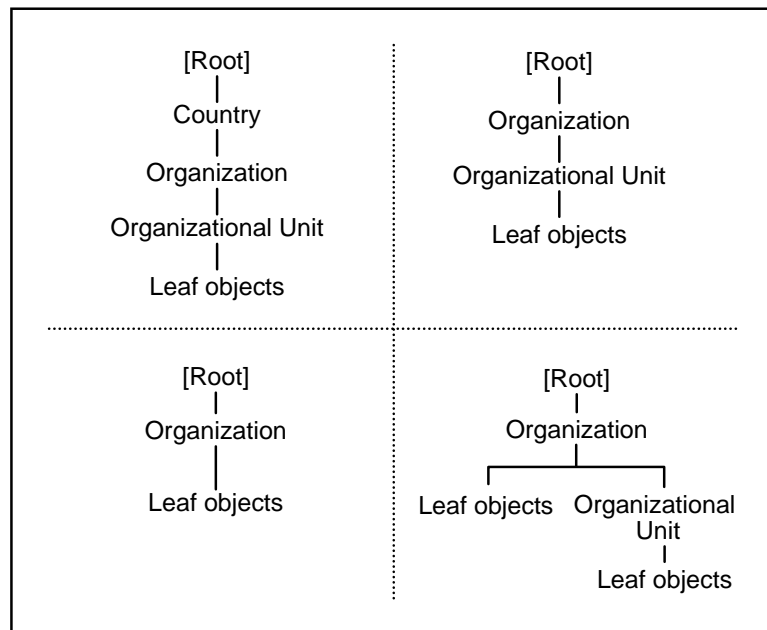
<b>Leaf object</b>	<b>Description</b>
Directory Map	Refers to a directory on a volume.  You cannot look at the file structure on the volume from the Directory Map object, but login scripts can use the MAP command with a Directory Map object to record the location of frequently used applications. If the application moves, you change only the directory map; all login scripts remain unchanged.
Group	Assigns a name to a list of User objects in the Directory. That way you can assign rights to the group instead of to each user—the rights transfer to each user in the group.
NetWare Server	Represents a server running any version of NetWare. The Network Address property identifies the server's location on the network. The NetWare Server object is referred to in several other objects to specify where to find items such as volumes.
Organizational Role	Defines a position or role within an organization. Use the Organizational Role object to specify a position that can be filled by different people, such as Team Leader or Vice President.
Print Queue	Represents a network print queue.
Print Server	Represents a network print server.
Printer	Represents a network printing device.
Profile	Represents a login script used by a group of users who need to share common login script commands but who are not necessarily located under the same container in the Directory tree, or who are a subset of users in the same container.
User	Represents the people who use your network.  In the User object's properties, you can store information about the user, such as a telephone number, an address, or group membership. You can also manage users by storing information about a user's print job configuration, account and password restrictions, or rights to files and directories.
Unknown	Represents a NetWare Directory Services object that has been corrupted and can't be identified as belonging to any of the other object classes.

**Table 1-10**      **Types of leaf objects**

Leaf object	Description
Volume	<p>Represents a physical volume on the network.</p> <p>In the Volume object's properties, you store information about which server the volume is located on, the volume's name (for example, SYS:), the volume's owner, space restrictions for users, and so forth.</p>

**Objects in the Directory tree**

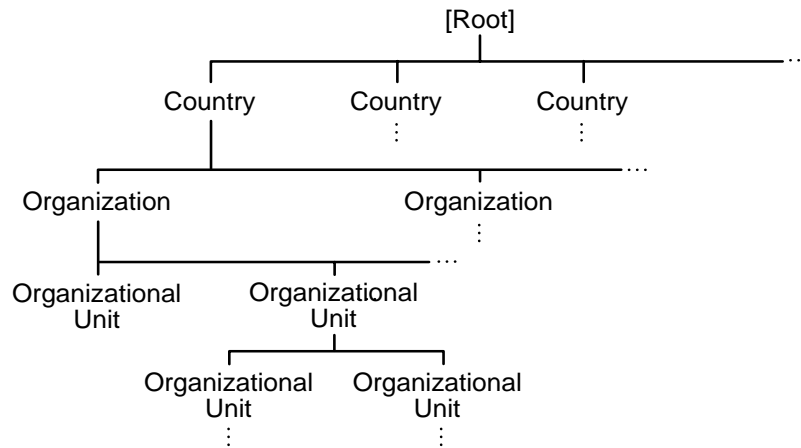
In a Directory tree, you can place container objects and leaf objects in different configurations, according to your company's needs. The following figure shows possible configurations.



**Figure 1-36**      **Possible configurations for a Directory tree**

The Country and Organizational Unit objects are optional, but you must include at least one Organization object in your Directory tree.

You are not limited to using only one container object in a tree; you can use many at each level. The following figure shows an example Directory tree that has several container objects at each level of a tree.



**Figure 1-37**

**Example Directory tree with several container objects**

**Object names**

The path from an object to the root of the Directory tree forms the object's complete name, which is a unique name. Most leaf objects have a common name. For User objects, the common name is their login name, displayed in the Directory tree.

Other leaf objects also have common names displayed in the Directory tree, such as a Printer object name or a Server object name.

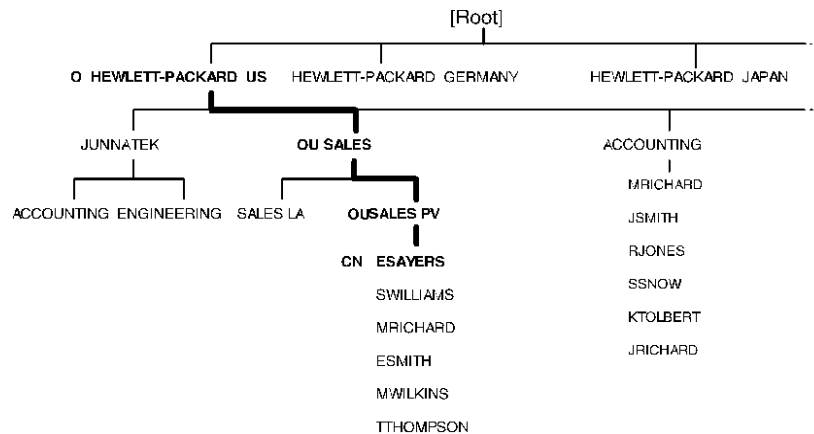
Container objects do not have common names. They are referred to by their Organizational Unit name, or Organization name, or Country name.

An object's complete name consists of its common name (if it has one), followed by a period (.), then the name of the container object, also followed by a period, and on up through succeeding container object names through the root of the tree.

For example, in the following figure, for User object ESAYERS (the common name), the complete name would be

ESAYERS.SALES PV.SALES.HEWLETT-PACKARD US

O



**Figure 1-38 Complete and common names**

At times, such as when you move from one container object to another, you must include the object's name type in the complete object name.

For example, in the previous figure you would express ESAYERS as

**CN=ESAYERS.OU=SALES PV.OU=SALES.O=HEWLETT-PACKARD US**

(where CN is the common name, OU is the Organizational Unit name, and O is the Organization name).

---

**NOTE:** When you include the Country container object in a Directory tree, always designate the object's name type (CN or OU or O), even when you refer to objects located in the same container object.

---

When querying the Directory, you can supply the object's complete name; then receive information that describes that object.

Or, you can supply an object's property value and receive a list of object names that have that value.

For example, to find all users with a last name of Smith, then "Smith" is the value you want to find in the last name property of User objects.

**Object contexts**

NetWare Directory Services allows you to refer to objects according to their positions within a tree. When you add an object (such as a server or user) to the network, you place that object in a container object in the Directory tree.

The position of the object within its container is its context. For example, in the previous figure, the context for User object ESAYERS is SALES.PV.SALES.HEWLETT-PACKARD.US.

When you move from one container object to another, you change contexts. Whenever you change contexts, indicate the complete name of the object you are changing contexts to.

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

For example, in the previous figure, if ESAYERS located in SALES.PV.SALES.HEWLETT-PACKARD.US wants information on ESMITH located in the same context, then ESAYERS only needs to refer to the User object as ESMITH.

**Object properties**

Each type of object has certain properties that hold information about the object.

For example, some User object properties include the login name, password restrictions, and group memberships. Some Profile object properties are the profile name, login script, and volume.

The only properties required for objects are those you enter when you create a new object. You must enter a value in each field.

Properties you must specify when you create an object are

- Properties that name the object.
- Properties required to create the object but that don't name it.

For example, when you create a Volume object, you must specify the host server that the volume is attached to.

## O

Many of an object's properties can contain multiple values. For example, the telephone number property, found in many object types, can contain several different telephone numbers.

The NETADMIN and NetWare Administrator utilities allow you to see and change properties for any object that you have sufficient rights to.

See also: "Managing Directory Service Objects" (*Supervising the Network*); "NetWare Directory Services (NDS)."

### **Object rights**

Qualities assigned to an object that controls what the object can do with directories, files, or other objects.

See also: "Rights."

### **ODI**

See "Open Data-Link Interface (ODI)."

### **ODINSUP**

See "Open Data-Link Interface Network driver interface specification SUPport (ODINSUP)."

### **Open Data-Link Interface (ODI)**

An architecture that allows multiple LAN drivers and protocols to coexist on network systems.

The ODI specification describes the set of interface and software modules used to decouple device drivers from protocol stacks and to enable multiple protocol stacks to share the network hardware and media transparently.

In a NetWare Services system, ODI runs only on clients.

The following figure illustrates the components of the ODI model in the client environment.



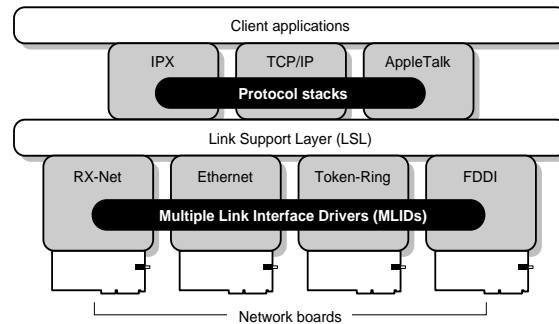


Figure 1-39

**ODI model**

The major components of the ODI architecture are described in the following sections.

**Multiple Link Interface Driver (MLID)**

The MLID is a device driver written to the ODI specification that handles the sending and receiving of packets to and from a physical or logical LAN medium.

Each driver is unique due to the adapter hardware and media, but the ODI eliminates the need to write separate drivers for each protocol stack.

ODI allows LAN drivers to function with protocol stacks independently of the media frame type and protocol stack details.

MLIDs interface with a network board and handle media frame header appending and stripping.

They also help demultiplex the incoming packets by determining their frame format.

A set of support modules provide all tools necessary to interface a LAN driver to the LSL. These are the Media Support Module (MSM), which contains general functions common to all drivers, and the Topology Specific Modules (TSMs), which provide support for the standardized media types of Ethernet, Token-Ring, RX-Net, and FDDI.

These modules and the Hardware Specific Module (HSM) are described later in this section.

## O

In DOS client environments, the MSM and TSMs are linked in with the HSM so that only one module is loaded.

### **Link Support Layer (LSL)**

The LSL is a software module that implements the interface between drivers and protocol stacks. It essentially acts like a switchboard, directing packets between the drivers and protocol stacks.

Any ODI LAN driver can communicate with any ODI protocol stack through the LSL. The LSL handles the communication between protocol stacks and MLIDs.

Because the ODI allows the physical LAN medium to support many different types of protocols (for example, IPX, TCP/IP, AppleTalk and LAT might all be supported on one Ethernet network adapter), the MLID receives packets destined for different protocol stacks that might be present in the system.

The LSL then determines which protocol stack the packet is to be delivered to. Next, the protocol stack determines what should be done with the packet.

When the protocol stack must transmit a packet, the protocol stack hands the packet to the LSL, which then routes the packet to the appropriate MLID.

The LSL enables the protocol stacks to handle sending and receiving.

The LSL also tracks the various protocols and MLIDs that are loaded in the system and provides a consistent method of finding and using each of the loaded modules.

### **Media Support Module (MSM)**

The MSM standardizes and manages primary details of interfacing ODI MLIDs to the LSL and operating system.

The MSM handles generic initialization and run-time issues common to all drivers.

### **Topology Specific Module (TSM)**

The TSM manages operations unique to a specific media type, such as Ethernet, or Token-Ring. Multiple frame support is implemented in the TSM so that all frame types for a given media are supported.

**Hardware Specific Module (HSM)**

The HSM is created for a specific network board. The HSM handles all hardware interactions. Its primary functions include adapter initialization, reset, shutdown, and removal.

It also handles packet reception and transmission. Additional procedures may also provide support for timeout detection, multicast addressing, and promiscuous mode reception.

**Open Data-Link Interface Network driver interface specification Support (ODINSUP)**

An interface that allows the coexistence of two network driver interfaces: the Network Driver Interface Specification (NDIS) and the Open Data-Link Interface (ODI) specification (see “Open Data-Link Interface (ODI)”).

ODINSUP allows you to connect to dissimilar networks from your workstation and use them as if they were one network.

In a NetWare Services system, ODINSUP relates only to clients.

ODINSUP also allows NDIS protocol stacks to communicate through the ODI's LSL and MLID. This way, NDIS and ODI protocol stacks can coexist in the same system, making use of a single ODI MLID.

For example, after you load ODINSUP on your workstation, you can log in to 3Com's 3+Share, Microsoft's LAN Manager, or IBM's LAN Server network, and also log in to a NetWare network, using the same network board in the workstation.

You can then copy files and run applications between the two networks as if they were one.

When ODINSUP is loaded, you can use a wider variety of programs without having to worry about compatibility and without reconfiguring or rebooting your workstation to switch from one type of network to another.

### How ODINSUP Works

ODINSUP functions as a default protocol stack. As a default protocol stack, it accepts packets from the ODI Link Support Layer (LSL) that aren't specifically marked with a protocol identifier (PID) for registered ODI protocol stacks (such as IPX or TCP/IP).

When it receives a packet, ODINSUP provides the packet to the NDIS Protocol Manager and passes it on to the NDIS protocol stack.

ODINSUP allows the NDIS protocol stack to communicate with a network board.

The NDIS protocol stack acts as though it is communicating with the network through an NDIS 2.0 MAC (Media Access Control) driver, and isn't aware of the ODINSUP protocol stack.

The details of the packet's transmission are handled by the Multiple Link Interface Driver (MLID), which is the ODI driver. This is illustrated in the following figure.

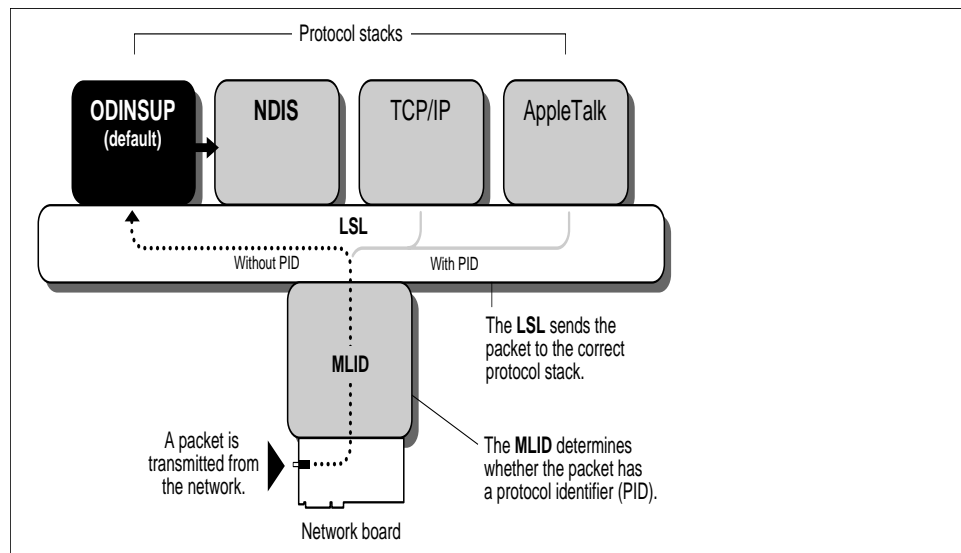


Figure 1-40

### ODINSUP

**Organization object**

A container object that helps you organize other objects in the Directory and allows you to set template information for users created in this container.

For example, you could use an Organization object to represent a company, or a university with various departments, or a department with several project teams.

The Organization object is a level below the Country object (if used), and a level above the Organizational Unit object (if used).

See also: “Creating Container Objects” (*Supervising the Network*); “Object.”

**Organizational Role object**

A leaf object that defines a position or role within an organization. Use the Organizational Role object to specify a position that can be filled by different people, such as Team Leader or Vice President.

See also: “Managing Organizational Role Objects” (*Supervising the Network*); “Object.”

**Organizational Unit object**

A container object, a level below the Organization object, that helps you to further organize other objects in the Directory and also allows you to set template information for users created in this container.

For example, you could use an Organizational Unit object to designate a division, a business unit, a project team, or a college or department within a university.

See also: “Creating Container Objects” (*Supervising the Network*); “Object.”

---

## P

### **Packet**

A unit of information used in network communication.

Messages sent between network devices (for example, workstations and NetWare server.) are formed into packets at the source device.

The packets are reassembled, if necessary, into complete messages when they reach their destination.

A packet might contain a request for service, information on how to handle the request, and the data that will be serviced.

An individual packet consists of headers and a data portion. Different headers are appended to the data portion as the packet travels through the communication layers.

A message that exceeds the maximum size is partitioned and carried as several packets. When the packet arrives at its destination, the headers are stripped off in reverse order and the request is serviced.

For example, the NetWare Core Protocol (NCP) attaches a write request header and an IPX header to a piece of data to be written.

Then the workstation's IPX communication protocol fills in the IPX header, designating, among other things, the source of the request and the packet length.

Finally, the Multiple Link Interface Driver (MLID) adds a hardware or MAC (Media Access Control) frame header.

See also: "Communication protocol"; "Ethernet configuration"; "Large Internet Packet (LIP)"; "NetWare daemon."

### **Packet Burst protocol**

A protocol built on top of IPX that speeds multiple-packet NCP (NetWare Core Protocol) reads and writes. of files.

The Packet Burst protocol speeds the transfer of NCP data between a workstation and a NetWare server by eliminating the need to sequence and acknowledge each packet.

Packet Burst protocol is more efficient than the one-request/one-response protocol in earlier NetWare versions. With Packet Burst protocol, the server or workstation can send a whole set (burst) of packets before it requires an acknowledgment.

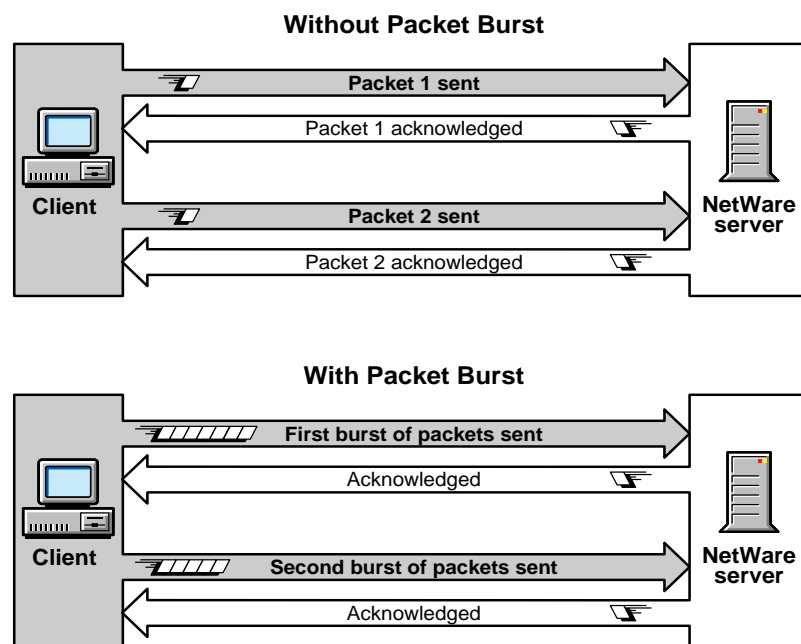


Figure 1-41

#### Packet Burst protocol

By allowing multiple packets to be acknowledged, Packet Burst protocol reduces network traffic.

Packet Burst protocol also monitors dropped packets and retransmits only the missing packets.

NetWare Services requires the following configuration parameters to be set in nwconfig to enable Packet Burst at the server:

- burst\_mode\_protocol=(on or off; on=default)
- burst\_mode\_clients=(number of clients allowed: 0-1,000)

Other parameters can be used to tune the Packet Burst protocol.

For NETX workstations to send and receive Packet Burst data, you must enable Packet Burst under the NetWare DOS Requester. On the VLM client, burst-mode is enabled by default.

When Packet Burst-enabled servers or workstations transfer data to servers or workstations that don't have Packet Burst enabled, the data defaults to normal NCP mode (one-request/one-response).

### **Parallel port**

A printer interface that allows data to be transmitted a byte at a time, all eight bits moving in parallel.

See also "LPT1."

### **Parent directory**

The directory immediately above any subdirectory. For example, SYS:ACCTS would be the parent directory of the subdirectory SYS:ACCTS/RECEIVE.

See also: "Directory structure, file system"; "Directory tree."

### **Parent objects**

Container objects that contain other objects.

See also: "Object."

### **Parity**

A method of checking for errors in transmitted data.

See "Serial port."



## Partition, Directory Services

In NetWare Directory Services, a logical division of the Directory's global database. A partition forms a distinct unit of data in the Directory tree that you use to store and replicate Directory information.

Each partition consists of a container object, all objects contained in it, and data about those objects. Partitions don't include any information about the file system or the directories and files contained there.

The following figure shows the default partition created for the first server installed and for all the new Organizational Units created in which NetWare Services servers were installed.

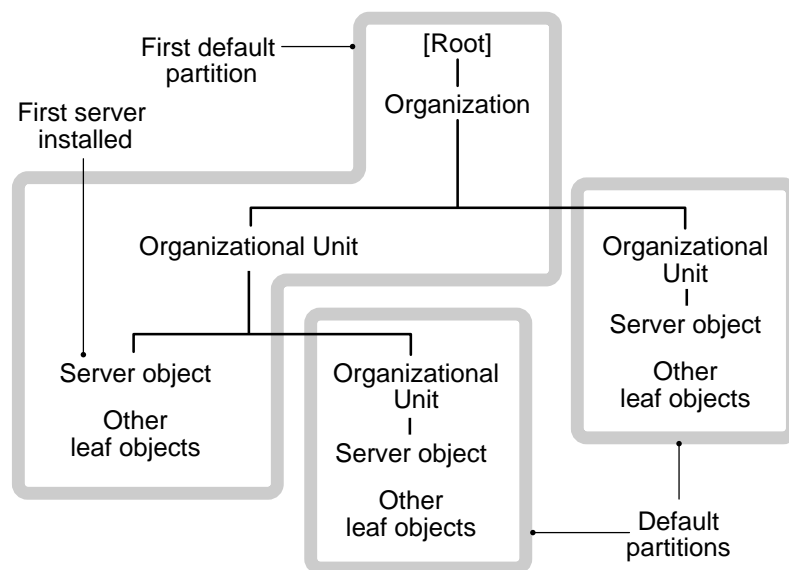


Figure 1-42

### Example of default partitions

Under NetWare Directory Services, an object resides in only one partition, but through distributed operations, the object can be accessed from any point on the network.

To optimize access to different areas of the Directory, each partition can be replicated and stored at many locations.

Partition replication improves access and provides the Directory with fault tolerance. Since a partition can be replicated at several locations, damage to one of the replicas does not need to interrupt access to the partition information.

Furthermore, the Directory allows replicas to be designated as read-write or read-only, thus controlling the introduction of updates into the system.

The tree of partitions is transparent to Directory users (unless they are running Partition Manager); users usually see only a global tree of Directory objects.

Partition replicas are stored on NetWare servers. Multiple partitions can be stored on the same NetWare server; none of the partitions need to be contiguous to each other.

Related utility: “PARTMGR” in (*Utilities Reference*).

See also: “NetWare Directory Services (NDS)”; “Replica.”

### **Partition, file system**

See “File system, HP-UX.”

### **Partition management**

The method of managing NetWare Directory database partitions and replicas.

Partition management divides the directory into partitions and makes and manages various replicas of these partitions.

Partition management allows you to

- Create, delete, rebuild, and synchronize partitions
- Display partitions and partition details
- Add, delete, and display replicas

Partition management won't allow you to modify the structure of existing partitions; partitions can't be split, combined, or moved.

## Password

The characters a user must type to log in with. NetWare allows the network supervisor to specify whether passwords are required and, if so, to assign a login password to each user on the network.

The network supervisor can also specify whether passwords must be unique and whether they must be changed periodically.

In NetWare, login passwords are encrypted at the workstation and put into a format that only the NetWare server can decode. This format helps prevent intruders from accessing network files.

Hybrid users have two passwords, one for their HP-UX account and one for their NetWare account.

See also “Authentication”; “Security” (Login security); “User object”, “High Performance File System (HPFS).”

## Path

The location of a file or directory in the file system.

For example, the path for file REPORT.FIL in subdirectory ACCTG in directory CORP on vol SYS: of server ADMIN is

```
ADMIN\SYS:CORP\ACCTG\REPORT.FIL
```

In HP-UX, the path for REPORT.HL in subdirectory ACCTG in subdirectory CORP of the root (/) directory is

```
/CORP/ACCTG/REPORT.HL
```

See also

## Permissions

Access levels that can be set for HP-UX files and directories: read (r), write (w), and execute (x). Each HP-UX file or directory also has an owner who, along with the system administrator (root user), controls these permissions.

Each of the HP-UX permissions can be set individually for three different sets of users: the file or directory owner, the group to which the owner belongs, and all other users.

See also “HPFS”; “Rights.”

### **Port, hardware**

A connecting component that allows a microprocessor to communicate with a compatible peripheral.

See also “Parallel port”; “Serial port.”

### **Port, software**

A memory address that identifies the physical circuit used to transfer information between a client LAN driver and a peripheral.

### **Power conditioning**

Methods of protecting sensitive network hardware components against power disturbances.

Power disturbances can be categorized in several ways:

- A transient (sometimes called a spike or surge). A very short, but extreme, burst of voltage.
- Noise or static. A smaller change in voltage.
- Blackouts and brownouts. The temporary drop in or loss of electrical power.

### **Protection against power disturbances**

Three types of protection are available:

- Suppression. Protects against transients. The most common suppression devices are surge protectors that usually include circuitry to prevent excess voltage.
- Isolation. Protects against noise, using ferro-resonant isolation transformers to control voltage irregularities.
- Regulation. Protects against brownouts and blackouts. The Uninterruptible Power Supply (UPS) is the most commonly used form of regulation.

Proper use of power conditioning devices greatly reduces network maintenance costs.

Make sure the proper amperage is available for each system; dedicated power lines should provide ample amperage.

Also, make sure all outlets are grounded. Power conditioning devices connected to poorly grounded outlets offer very little protection.

### **Primary time server**

In NetWare Directory Services, a 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 workstations.

See “NetWare Directory Services (NDS)”; “Time synchronization.”

### **Print device**

A printer, plotter, or other peripheral that prints from the network.

Print devices require print jobs to be formatted with the correct control sequences that set and reset the printer, produce bolding, underlining, and so forth.

Most applications provide you with a printer definition to correctly format print jobs. If you use a DOS application that can't format the print job, NetWare provides over 50 printer definitions in SYS:PUBLIC.

Each printer definition has a .PDF extension and must be imported into NetWare print services using the NetWare Administrator or PRINTDEF.

If your printer does not match the predefined print devices, use your printer manual to create a print device by defining the correct control sequences in the NetWare Administrator or PRINTDEF.

This involves setting up print device functions that set and reset the printer, control bold, emphasis, italics, print size, font selection, colors, etc., depending on the printer.

You specify modes (groups of one or more functions) for use in print job configurations. Modes can prepare the printer for a print job, combine functions, reset the printer back to default settings, etc.

## P

You specify modes using the NetWare Administrator or PRINTCON. Then NPRINT, CAPTURE, or PCONSOLE use the print job configuration option to send print jobs to your printer with the correct control sequences.

Related utilities: “CAPTURE”; “NPRINT”; “PCONSOLE”; “PRINTCON”; “PRINTDEF.” (*Utilities Reference*)

See also: “Working with Print Device Definitions and Printer Forms” (*Print Services*); “Printing.”

### **Print device mode**

A sequence of print functions (also called printer commands, control sequences, or escape sequences) that determines the appearance of the printed file.

A print device mode can define the style, size, boldness, and orientation of the typeface.

Print device modes are designated using the NetWare Administrator or PRINTDEF.

Related utilities: “NetWare Administrator”; “PRINTDEF” (*Utilities Reference*).

### **Print job**

A file stored in a print queue directory waiting to be printed. As soon as a print server sends a print job to the printer, the print job is deleted from the queue directory.

Each print job is assigned a filename with a variation of the first four digits of the print queue directory ID, four more numerals, and a .Q extension.

For example, if the print queue directory is named 4B020057.QDR, the first print job to enter the empty print queue would be named 024B0001.Q.

If more print jobs entered the print queue before the current print job was printed, they would be named 024B0002.Q, 024B0003.Q, 024B0004.Q, etc.

As soon as print job 024B0001.Q is printed, the next print job to enter the print queue is named 024B0001.Q. Like all print jobs, it would follow the first-in, first-out basis, unless the print job was held or a print queue operator changed the order of service.

### **Print job configuration**

A group of characteristics that determine how a job is printed. The characteristics may include the following:

- Printer the print job will be printed on
- Print queue the print job is sent through
- Number of copies to print
- Use of a banner page
- Printer form
- Print device mode

You can create print job configurations using either the “NetWare Administrator” or “PRINTCON” utility.

See also: “Creating and Managing Print Job Configuration” (*Print Services*).

### **Print queue**

A network directory that stores print jobs. When the printer assigned to a print queue is ready, the print server takes the print job out of the print queue and sends it to the printer.

The print queue can hold as many print jobs as disk space allows.

When you create a print queue using either the “NetWare Administrator” or “PCONSOLE” utility, a corresponding directory is created.

---

**NOTE:**

You can also create print queues using the graphical utility Printer Setup. For more information, see “Configuring and Managing Print Services with Printer Setup” in *Print Services*.

In NetWare Directory Services mode, the print queue directory resides in the QUEUES directory on the volume specified.

In bindery mode, the print queue directory resides in the SYSTEM directory on volume SYS: of the current server.

The “Quick Setup” option in PCONSOLE automatically creates a print queue for each printer.

### **Print queue name**

The print queue directory is assigned a random name. This name is the print queue ID seen in NetWare Administrator or PCONSOLE with a .QDR extension.

For example, print queue LETTERHEAD\_Q might be directory LEGAL/SYS:SYSTEM/4B020057.QDR if configured on NetWare server LEGAL. The print queue ID would then be 4B020057.

All print queue directories have the extension .QDR and contain hidden and system files with a .SYS and .SRV extension. These files are visible with the NDIR utility.

These files begin with “Q\_” and use a variation of the first four digits of the print queue directory ID.

The print queue directory in the previous example would have hidden and system files Q\_024B.SYS and Q\_024B.SRV.

See also “Print job.”

### **Additional information**

To simplify your printing setup, create one queue for each printer.

Also, if you name the queue according to the type or location of the printer, it is easier to remember which queue is serviced by which printer.

When you create a print queue, user ADMIN is assigned as a print queue operator, and all users in the same context are assigned as print queue users.

To change these default assignments, use NetWare Administrator, or PCONSOLE.

Related utilities: “NetWare Administrator”; “PCONSOLE” (*Utilities Reference*)



See also: “Configuring and Managing Print Services with Printer Setup”; “Configuring and Managing Print Services with NetWare Administrator”; “Configuring and Managing Print Services with PCONSOLE” (*Print Services*); “Print queue operator.”

### **Print queue operator**

A user who can edit other users' print jobs, delete print jobs from the print queue, or modify the print queue status by changing the operator flags.

Print queue operators can also change the order in which print jobs are serviced. They can also change the service mode.

User ADMIN can assign users to be print queue operators as necessary.

Related utilities: “NetWare Administrator”; “PCONSOLE” (*Utilities Reference*).

See also: “Configuring and Managing Print Services with Printer Setup”; “Configuring and Managing Print Services with NetWare Administrator”; “Configuring and Managing Print Services with PCONSOLE” (*Print Services*); “Print Server Operator.”

### **Print queue polling time**

The time interval the print server waits between checking the print queues for jobs ready and waiting to be printed. Users can specify the time period.

See also: “Print server.”

### **Print server**

The agent that monitors print queues and transfers pending print jobs from each queue to their associated printers.

The NetWare Services print server is a HP-UX daemon (PServer) that can send jobs to local HP-UX printers, remote UNIX printers, remote NetWare printers, and remote AppleTalk printers.

The PServer daemon lets any client on the network print to any printer on the network.

See also: “Configuring and Managing Print Services with Printer Setup”; “Configuring and Managing Print Services with NetWare Administrator”; “Configuring and Managing Print Services with PCONSOLE” (*Print Services*); “Printer”; “Printing.”

### **Print Server object**

A leaf object that represents a network print server.

See also “Object.”

### **Print Server Operator**

A user or member of a group delegated rights by ADMIN to manage the print server.

A Print Server Operator has rights to control notify lists, printers, and queue assignments.

Related utility: “PCONSOLE” (*Utilities Reference*).

See also: “Configuring and Managing Print Services with Printer Setup”; “Configuring and Managing Print Services with NetWare Administrator”; “Configuring and Managing Print Services with PCONSOLE” (*Print Services*); “Printer”; “Printing.”

### **Printer**

Computer equipment used to produce printed material.

Network printers can be attached in the following ways:

- Directly to the network
- To the printer port of a NetWare server
- To the printer port of a PC workstation

In NetWare Services, users can specify printer names as the destination of their print jobs. In previous versions, users had to specify the print queue. (Users can still specify print queues.)

**Network printer drivers**

Every network printer requires a network printer driver to pass a print job from the network to the printer. The type of driver depends on how the printer is attached to the network.

Printers attached to the network store their own printer driver.

Printers attached to the workstation need NPRINTER.EXE loaded on the workstation.

Printers attached to HP-UX can use the printer driver in either the NPrinter daemon or the PServer daemon.

**Bindery and NetWare Directory Services differences**

In bindery-based NetWare, printers are a subset of the print server. For that reason, a print server must exist before you can define printers.

In NetWare Directory Services, printers are objects used in conjunction with Print Server and Print Queue objects. They can be added, modified, or removed independently.

Each network printer must be defined using NetWare Administrator, or PCONSOLE.

See “Print Services” for detailed information on NetWare print services, including step-by-step instructions for configuring and managing print servers, print queues, and printers.

Related utilities: “NetWare Administrator”; “PCONSOLE” (*Utilities Reference*).

See also “Remote Reset.”

**Printer definition**

A set of printer control characters used to interpret commands to bold, italicize, and center text. Printer definitions are specific to a printer brand and model.

If your printer or plotter does not function with one of the available print device definitions, you can create your own set of control characters and specify what forms your print device accepts. See “Working with Print Device Definitions and Printer Forms” in Print Services.

### **Printer form**

A print option designed to prevent print jobs from being printed on the wrong paper.

NetWare print services allows you to send print jobs that will not print until you make sure that the correct paper is in the printer.

For example, suppose your printer uses regular, letterhead, and bond paper. For each type of paper, you can create a printer form. Each form has a unique name and number (between 0 and 255).

If you specify this form in a print job configuration or in NPRINT or CAPTURE with the form option, the print job won't print unless the mounted form matches the number required by the print job.

After you check to make sure the proper paper is in the printer, you can change the number of the currently mounted form at the print server console or using the NetWare Administrator or PCONSOLE.

Printer forms are defined using the NetWare Administrator or PRINTDEF.

Related utilities: “CAPTURE”; “NetWare Administrator”; “NPRINT”; “PCONSOLE”; “PRINTDEF” (*Utilities Reference*).

### **Printer object**

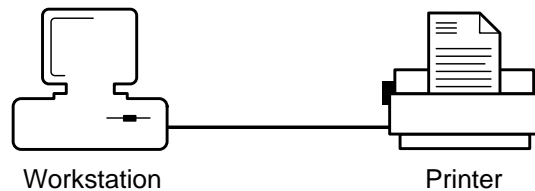
A leaf object that represents a physical printing device on the network.

See also “Object.”

### **Printing**

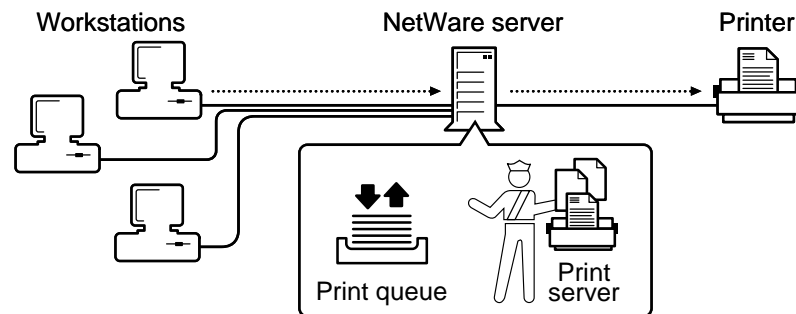
The ability to transfer data from computer files to paper.

NetWare Services allows users to share printing hardware, where previously each personal computer had to have a printer attached to one of its printer ports.



**Figure 1-43 Non-network printing**

NetWare Services uses a print queue and print server to allow workstations to print to a printer. The print server takes print jobs from the print queue and sends them to the printer.



**Figure 1-44 Network printing**

### Process

A systematic sequence of operations that transforms raw data into useful information. Each time a program is run on HP-UX, it runs as a process and is assigned a process ID. Because HP-UX is a multitasking system, more than one process can run at the same time. Some processes run in the background without the user being aware of them.

NetWare Services has three main processes or daemons

- NetWare Protocol Stack daemon (NSPD)
- NetWare daemon
- NetWare Engines

See also “Client”; “Engine”; “NetWare daemon”; “NPS daemon”; “ncp\_engine.”

### **Profile login script**

A type of login script that sets environments for a group of users. Use profile login scripts if there are groups of users with identical login script needs.

Profile login scripts are optional; if used, they execute after the container login script and before the user login script.

See also “Login scripts.”

### **Profile object**

A leaf object that represents a login script used by a group of users who need to share common login script commands.

The group of users may not necessarily be located under the same container in the Directory tree, or they may be a subset of users in the same container.

You can grant trustee assignments to a profile object. See “Managing Profile Objects” in *Supervising the Network*.

See also “Object.”

### **Prompt**

A character or message that appears on the display screen and requires a response (such as a command or a utility name) from the user.

Standard types of prompts include

- The DOS prompt, which, by default, displays the current drive letter followed by a > symbol: F>
- The UNIX prompt, which displays a # for root users and a \$ for other users.

The client DOS prompt is a DOS environment setting. You can change the DOS prompt using the SET PROMPT command in a batch file (such as AUTOEXEC.BAT), in the login script, or at the command line.

For example, to change your DOS prompt at the command line so that the prompt displays the current drive mapping followed by the > symbol, you would type

```
SET PROMPT=$P$G <Enter>
```

See your respective DOS manual for further details on changing prompts, including prompt variables. (The NetWare server console prompt can't be changed.)

See also “Drive mapping”; “Login scripts.”

## Property

A characteristic of a NetWare Directory Services object. 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 login name, E-mail address, password restrictions, group memberships, etc.

As another example, a Profile object's properties include profile name, login script, volume, etc.

The only properties required for objects are those you enter when you create a new object. You must enter a value in each field.

Properties you must enter when you create an object can be properties that name the object, or properties required to create the object but that don't name it.

For example, when you create a volume object, you must specify the host server that the volume is attached to.

Many of an object's properties can contain multiple values. For example, the telephone number property, found in many object types, can contain several different telephone numbers.

The NETADMIN and NetWare Administrator utilities allow you to see and change properties for any object that you have sufficient rights to.

Related utilities: “NETADMIN”; “NetWare Administrator” (*Utilities Reference*).

See also “Object.”

### **Property rights**

Rights that apply to the properties of a NetWare Directory Services object.

See “Rights.”

### **Protected mode**

The mode that 80286, 80386, and 80486 processors run in by default. When running in protected mode, these processors are not subject to the same memory constraints as 8086 processors.

The 80286 processor uses a 24-bit address bus, and can address up to 16 MB of memory. The 80386 and 80486 processors use a 32-bit address bus, and can address up to 4 GB of memory.

Protected mode operating provides the capability of multitasking (running more than one application or process at a time).

Protected mode allocates memory to various processes running concurrently so that memory used by one process does not overlap memory used by another process.

By contrast, 8086 processors can address only 1 MB of memory, and can run only one application or process at a time.

80286, 80386, and 80486 processors can be set to run in real mode, in which case they emulate an 8086 processor (and are subject to its memory constraints).

See also: “Real mode.”

### **Protocol**

Conventions or rules used by a program or operating system to communicate between two or more endpoints.

See also: “Communication protocol.”



**Protocol stack**

The protocols that allow a UNIX host to communicate with the rest of the NetWare Services network. The protocol stack contains IPX, SPX, RIP, and SAP.

See “IPX”; “SAP”; “SPXII”; “RIP.”

**PUBLIC directory**

The SYS:PUBLIC directory, created during network installation, that allows general access to the network and contains NetWare utilities and programs for network users.

NetWare users running DOS have a search drive mapped to SYS:PUBLIC through the container login script and are assigned Read and File Scan rights to this directory.

NetWare users running access NetWare utilities from the SYS:PUBLIC/OS2 directory.

Do not delete the PUBLIC directory or any files in it.

See also: “Directory structure, file system”; “LOGIN directory”; “MAIL directory”; “SYSTEM directory.”

**Public files**

Files that must be accessed by all NetWare users, including NetWare utilities, help files, and some message and data files.

By convention, the files are located in SYS:PUBLIC for DOS users.

All NetWare users have Read and File Scan rights to the files.

**Public trustee**

A special trustee that can be added to any object, directory, or file.

[Public] is only used in trustee assignments and must always be entered within square brackets. [Public] can be added or deleted like any other trustee. An Inherited Rights Filter will block inherited rights for [Public].

Whatever rights are granted to [Public] are effective for any object in NetWare Directory Services that does not have any other effective rights.

This is similar to granting trustee rights to user GUEST or group EVERYONE in previous NetWare versions.

Make [Public] a trustee of areas that every object should have access to. (A user does not have to log in to access areas where [Public] is granted rights.)

In most cases, it is better to make a container object a trustee, rather than making [Public] a trustee. This grants rights only to the objects within the container, improving security control.

During NetWare installation, [Public] is granted the Browse object right at the root object of the Directory tree.

See also “Inherited Rights Filter, NDS object”; “Trustee.”

### **Purge attribute**

A file system attribute that causes NetWare to purge the directory or file when it is deleted.

Because file salvage is not supported, all files are immediately purged upon deletion.

NetWare Services does not support purging and does not use this attribute.

See “Attributes.”

---

## Q

### **Queue**

A list of jobs to which new jobs are added at the end and jobs are accessed from the beginning. There are various types of queues, including the NetWare printer queue.

The list of print jobs is stored in a network directory. When the printer assigned to the print queue is ready, the print server takes the print job out of the print queue and sends it to the printer.

See “Print queue.”

### **Queue polling time**

The time interval the accessing agent waits between checking the queue for jobs ready to process.

---

## **R**

### **RAM**

(Random Access Memory) The internal dynamic storage of a computer that can be addressed by the computer's operating system.

See "Memory, DOS management."

### **Read-ahead cache**

A data cache that can be used to enhance client reads for clients that do not use packet burst mode protocol. While a client is processing the first response to a read request, the NetWare engine reads the rest of the file into read-ahead cache so that subsequent requests can be served from memory.

See "Cache memory"; "Packet Burst protocol."

### **Read Only attribute**

A file system attribute that indicates that no one can write to the file.

See "Attributes."

### **Read right**

A file system right that grants the right to open and read files.

Also, a Directory Services property right that grants the right to read the values of the property.

See "Rights."

### **Real mode**

The mode that allows 80286, 80386, and 80486 processors to emulate an 8086 processor and run as though they actually were an 8086 processor.

The 8086 processor uses a 20-bit address bus, and can address up to 1 MB of memory. The 8086 processor is also limited to running only one application or process at a time.

When running in protected mode, the 80286, 80386, and 80486 processors are capable of multitasking and addressing much more than 1 MB of memory.

When running in real mode, these processors are subject to the same 1MB memory constraint as the 8086 processor, and they can run only one application or process at a time.

However, the 80286, 80386, and 80486 processors running in real mode perform more efficiently than the 8086 processor, because they operate at a faster clock rate.

See also “Protected mode.”

### **Record locking**

A feature that prevents different users from gaining simultaneous access to the same record in a shared file, preventing overlapping disk writes and ensuring data integrity. A physical record lock—also called a byte-range lock—locks a specific byte range within a file; a logical record lock locks a record name associated with one or more files and/or records.

NetWare uses a physical record lock to control data access by multiple users. When a client requests access to the data, the application requests a physical lock on a byte range in the file. The Lock Manager records this byte range in its lock table and enforces the lock. NetWare Services enforces physical locks for all NetWare clients.

An application or client uses a logical record lock to control access by multiple users. The application assigns a name to each resource that needs to be locked so that only one user can access the resource. A logical record name is also referred to as a synchronization string. Logical record locking does not lock the data represented by the string; it only locks the string itself.

See also “File locking”; “nwcm”; “Synchronization services”; “UNIX host locking.”

### **Recursive copying**

The process of copying a specified source directory to a destination directory until all files and subdirectories in and below the specified source directory are copied.

Recursive copying copies all directories and files of a logical drive to the destination, keeping them exactly as they were in the source directory.

The DOS XCOPY and BACKUP utilities use recursive copying, as does the NetWare NCOPY command.

Related utility: “NCOPY” (*Utilities Reference*).

### **Reference time server**

A server that specifies the time for all other time servers and workstations to synchronize to.

See “Time synchronization.”

### **Registered resources**

Network resources that report information to the SNMP (Simple Network Management Protocol) agent.

The SNMP agent maintains a table of ID numbers that are assigned to the registered resources and is aware of which management daemon monitors each resource.

In NetWare Services the principal management daemons are the NWUMPS daemon and the NWUM daemon.

Each daemon keeps a hierarchical listing of registered resources but does not maintain a permanent record of the resources' attributes, status, and so forth. Instead of maintaining records of the actual data, the daemon regularly, or on request, monitors the resources registered with it.

The system manages registered resources through requests to the SNMP agent. The SNMP agent checks the table of ID numbers to determine which daemon is monitoring that registered resource and the request for services or

information is then transmitted to the resource through the appropriate daemon and then back through the daemon and the SNMP agent to the source of the request.

See also “NetWare management agents”; “Network management.”

### **Remote administration**

The use terminal emulation by network supervisors to run configuration (for example, nwcmm) and system management utilities from a client rather than from the host machine or other servers.

Remote administration can provide greater server security since servers can be locked in a safe place without keyboards and monitors.

Under NetWare Services all users logged into the system as HP-UX users are capable of administering the system, provided they have the appropriate privileges. Administration is accomplished by running the appropriate utilities. This varies from native NetWare, which has a set of special remote utilities, such as rconsole.

### **Remote boot**

See “Remote Reset.”

### **Remote connection**

A connection between a LAN on one end and a workstation or network on the other, often using telephone lines and modems.

A remote connection allows data to be sent and received across greater distances than those allowed by normal cabling.

### **Remote Reset**

The procedure that allows you to boot a DOS workstation (including a diskless workstation) from a remote boot image file on a NetWare server, rather than from a boot diskette in the workstation's local drive.

For a workstation to boot remotely, the remote program load (RPL) ROM must be installed on the workstation's network board.

## R

BOOTNCP.COM and RPL.COM, a DOS terminate-and-stay-resident (TSR) module need to be loaded in that order on a workstation, or RPL.NLM on a native NetWare 4 server. When RPL.COM is loaded on a DOS client it function as an “RPL Server.” An ODI driver must also be loaded at the RPL server.

A NET.CFG file is used by various ODI modules—including the LSL, NAN drivers, and protocol stacks— to get network configuration information at initialization. RPL.COM reads this file.

BOOTCONF.SYS is an ASCII text file in which the user specifies a unique Disk Image File for each workstation that needs access to different files.

Each line in BOOTCONF.SYS that contains a node address may specify more than one Disk Image filename, separated by one or more spaces. The user will be prompted at boot time to select one of these Disk Image files.

The remote boot image file includes the station's AUTOEXEC.BAT file, used by the station as if the file were present on a local boot diskette.

Copy the workstation's AUTOEXEC.BAT file to the remote boot image file, to the LOGIN directory, and to any default directory named in the workstation's login script.

### **Remote workstation**

A terminal or personal computer connected to the LAN by a router or through a remote asynchronous connection.

A remote workstation can be either a standalone computer or a workstation on another network.

### **Rename Inhibit attribute**

A file system attribute that prevents any user from renaming the directory or file.

See “Attributes.”



**Rename right**

A Directory Services object right that grants the right to change the name of an object, in effect changing the naming property.

See “Rights.”

**Replica**

A copy of a Directory partition.

For a NetWare Directory Services database to be distributed across a network, the database must be stored on many servers. Rather than have a copy of the whole database on each server, replicas of each partition are stored on many servers throughout the network.

You can create an unlimited number of replicas for each partition and store them on any server.

**Purpose**

Replicas serve two purposes:

- To eliminate any single point of failure.

For example, if a disk crashes or a server goes down, a replica on another server can still authenticate users to use the network and can provide information on objects in that partition.

With the same information distributed on several servers, you aren't dependent on any single server to authenticate who can use the network.

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

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

- To provide faster access to information for users across a WAN link.

For example, if a WAN link is used to access information, you can decrease access time and network traffic by placing a replica containing the needed information on a server that users can access locally.

Distributing replicas among servers lets you access information more quickly and reliably because the information comes from the nearest available server.

### **Types of replicas**

- Master replica. Although many replicas can exist in the Directory, only one is the master replica. Use it to create a new partition in the Directory database, or to read and update Directory information, such as adding or deleting objects.
- Read-write replica. Use to read or update Directory information (such as adding or deleting objects).
- Read-only replica. Use to view, but not to modify, Directory information.

### **Synchronization**

To maintain their fault tolerance, a partition's replicas must be periodically updated, or synchronized.

When changes are made in one replica, synchronization ensures that those changes are made in all other replicas of that partition, so that each partition's replica contains the same data as the other replicas.

Network supervisors control how often replica synchronization occurs. The more frequently synchronization occurs, the more consistent the replicas are. See "Managing the NetWare Directory Services Database" in *Supervising the Network*.

Use the Partition Manager utility to manage partitions, replicas, and synchronization.

Related utilities: "PARTMGR" (*Utilities Reference*).

See also "Partition, Directory Services."

### **Resources**

The manageable components of a network, including

- Networking components—cabling, hubs, concentrators, adapters, and network boards.
- Hardware components—servers, workstations, hard disks, printers, etc.
- Major software components the NetWare operating system and resulting network services such as file, mail, queue, communication, and so forth.
- Minor software components that are controlled by the operating system of its

subsystems—protocols, gateways, LAN and disk drivers, etc.

- Data structures and other network resources that do not easily fit into one of the above categories, or are created by a combination of network components—volumes, queues, users, processes, security, and so forth.

### **Restore**

A retrieval of data previously copied and backed up to a storage media. Perform a restore if data has been lost or corrupted since the backup.

See also “Backup.”

### **Rights**

Qualities assigned to a NetWare Directory Services (NDS) object that control the access the NDS object has to directories, files, or other NDS objects. Creating, reading, and other operations can be done only if an object has rights to perform them.

Rights are granted to a specific directory, file, or NDS object by trustee assignments. An object with a trustee assignment to a file, directory, or NDS object is a trustee of that file, directory, or object.

Within each NDS object is an Access Control List (ACL) that defines who has rights to the object. Files and directories contain similar information, but not in the form of ACLs.

An NDS object's ACL defines what has rights to the object but does not define what the object has rights to. For example, a printer object's ACL defines which users have rights to the printer, but a user object's ACL does not define which printers the user has rights to.

This differs from the file system. Rights to a file can be designated in the user object. However, rights to a file can also be granted in the file itself.

#### **Directory, file, object, and property rights**

Directory rights apply to the directory in the NetWare file system that they are assigned to, as well as to all files and subdirectories in that directory (unless redefined at the file or subdirectory level).

Directory rights are a part of the file system. They aren't assigned to NetWare Directory Services objects. But a User object can be granted Directory rights to a directory on a volume.

The following table describes directory rights.

**Table 1-11**      **Directory rights**

<b>Right</b>	<b>Description</b>
Supervisor	Grants all rights to the directory, its files, and subdirectories. The Supervisor right can't be blocked by an Inherited Rights Filter. Users with this right can grant other users rights to the directory, its files, and subdirectories.
Read	Grants the right to open files in the directory and read their contents or run the programs.
Write	Grants the right to open and change the contents of files in the directory.
Create	Grants the right to create new files and subdirectories in the directory. If Create is the only right granted to a trustee for the directory, and no other rights are granted below the directory, a drop box directory is created.  In a drop box directory, you can create a file and write to it. Once the file is closed, however, only a trustee with more rights than Create can see or update the file. You can copy files or subdirectories into the directory and assume ownership of them, but other users' rights are revoked.
Erase	Grants the right to delete the directory, its files, and subdirectories.
Modify	Grants the right to change the attributes or name of the directory and of its files and subdirectories—but does not grant the right to change the contents of them. (That requires the Write right.)
File Scan	Grants the right to see the directory and its files with the DIR or NDIR command, including the directory structure from that file to the root directory.
Access Control	Grants the right to change the trustee assignments and Inherited Rights Filter of the directory and of its files and subdirectories.

File rights apply only to the file they are assigned to. A trustee can also inherit rights to a file from the directory above the file.

The following table describes file rights.

**Table 1-12 File rights**

<b>Right</b>	<b>Description</b>
Supervisor	Grants all rights to the file. The file Supervisor right can't be blocked with an Inherited Rights Filter. Users who have this right can also grant other users any rights to the file, and can change the file's Inherited Rights Filter.
Read	Grants the right to open and read the file.
Create	Grants the right to salvage the file after it has been deleted. This right is not applicable to NetWare Services, which does not support salvageable files.
Write	Grants the right to open and write to an existing file.
Erase	Grants the right to erase (delete) the file.
Modify	Grants the right to change the attributes and name of the file—but does not grant the right to change its contents. (That requires the Write right.)
File Scan	Grants the right to see the file with the DIR or NDIR directory command, including the directory structure from that file to the root directory.
Access Control	Grants the right to change the trustee assignments and Inherited Rights Filter of the file.

Object rights apply to NetWare Directory Services objects. Object rights don't affect the properties of an object (see property rights below). A trustee can inherit rights to an object from the object above it.

The following table describes object rights.

**Table 1-13 Object Rights**

<b>Right</b>	<b>Description</b>
Supervisor	Grants all access privileges. A trustee with the Supervisor object right also has unrestricted access to all properties. The Supervisor object right can be blocked by the Inherited Rights Filter below the object where the Supervisor right is granted.
Browse	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.

**Table 1-13 Object Rights**

<b>Right</b>	<b>Description</b>
Create	Grants the right to create a new object below this object in the Directory tree. Rights are not defined for the new object. This right is only available on container objects, because non-container objects can't have subordinates.
Delete	Grants the right to delete the object from the Directory tree. Objects that have subordinates can't be deleted (unless subordinates are deleted first).
Rename	Grants the right to change the name of the object, in effect changing the naming property. This changes what the object is called in complete names.

Property rights apply to the properties of a NetWare Directory Services object. They can be assigned to each property, and a default set can apply to properties without specific rights set.

The following table describes property rights.

**Table 1-14 Property rights**

<b>Right</b>	<b>Description</b>
Supervisor	Grants all rights to the property. The property Supervisor right can be blocked by an object's Inherited Rights Filter.
Compare	Grants the right to compare any value to a value of the property. With the compare right, an operation can return True or False, but you can't see the value of the property. The Read right includes the Compare right.
Read	Grants the right to read the values of the property. Compare is a subset of Read. If the Read right is given, Compare operations are also allowed.
Write	Grants the right to add, change, or remove any values of the property. The Write right includes the Add or Delete Self right.
Add or Delete Self	Grants a trustee the right to add or remove itself as a value of the property. The trustee can't affect any other values of the property. This right is only meaningful for properties that contain object names as values, such as group membership lists or mailing lists. The Write right includes Add or Delete Self.

To grant directory or file rights to other objects, a trustee must have the Access Control right to a directory or file.

To grant object or property rights to other objects, a trustee must have the Write, Add or Delete, Self, or Supervisor right to the ACL property of the object.

Rights are granted and revoked by creating trustee assignments with the “RIGHTS”, “NETADMIN”, or “NetWare Administrator” utilities.

### **File Access Control**

NetWare Services allows a volume to enforce file access based on NetWare rights, HP-UX permissions, or a combination as follows:

- None. All users have access to all files and directories.
- NetWare. Netware-only enforcement. All file access is controlled by NetWare rights.
- UNIX. HP-UX-only enforcement. All file access is controlled by HP-UX permissions.
- Both. All file access is controlled by a combination of NetWare and HP-UX enforcement. In each case, the more restrictive of the two applies.

### **NetWare Rights**

NetWare rights refer to trustee assignments. Users may be granted rights to a file or a directory depending on the trustee rights they inherit or are granted. Whether NetWare rights are enforced depends on the file access control for the volume as specified in the voltab file. The enforcement of NetWare rights is the default.

### **HP-UX Permissions**

HP-UX permissions refer to the access modes on HP-UX files or directories. Under HP-UX, a user may be granted read, write, or execute permission to a file or a directory depending on their user ID, group ID, and the parent directory's access mode. NetWare Services optionally allows a volume to grant rights to a file based on HP-UX permissions. This is done by setting the file access control for a volume in the voltab file.

Related utilities: “NETADMIN”; “NetWare Administrator”; “RIGHTS”  
(*Utilities Reference*).

**R**

“HPFS” See also “Access Control List (ACL)”; “File Systems, NetWare Services”; “File system, HP-UX”; “Security”; “Trustee database.”

**RIP**

See “Router Information Protocol (RIP).”

**Root directory**

The highest directory level in a hierarchical directory structure.

With NetWare, the root directory is the volume; all other directories are subdirectories of the volume.

In HP-UX, all mounted files systems are presented to the user in a single hierarchical structure which has only one root directory (/).

See also “Directory structure, file system”; “Directory tree.”

**Root file system**

The directory structure that contains the programs necessary to boot the operating system and create a hierarchy of file systems. The root file system contains most of the programs required to administer HP-UX.

DOS file systems contain one root directory per volume. HP-UX file systems contain a single root directory which allows partitions to be spliced in.

**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.

The root object is a place holder; it contains no information.

See also “Directory tree”; “Object.”



### Root user

A special user in the HP-UX operating system who has unrestricted access to files and system resources.

See also “Root Directory.”

### Router

A workstation or NetWare server running software that manages the exchange of information (in the form of data packets) between network cabling systems.

A NetWare router runs as part of a NetWare server. It connects separate network cabling topologies or separate networks by way of the server’s NetWare operating system.

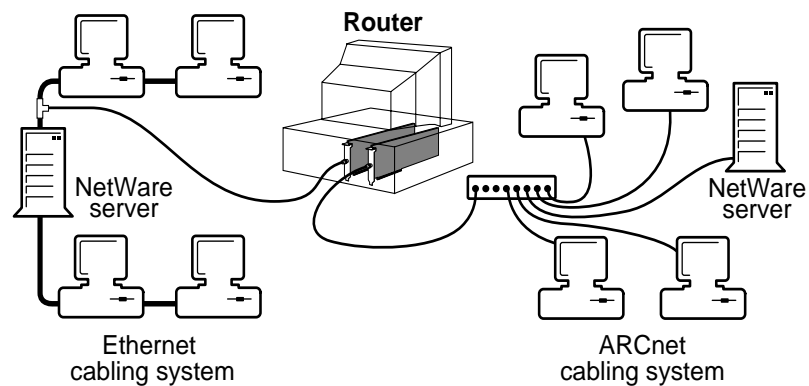


Figure 1-45

### Router

#### NetWare router vs. traditional bridge

A NetWare router, unlike a traditional bridge, does more than just transfer data packets between networks that use the same communications protocol.

A NetWare router is intelligent. It not only passes packets of data between different cabling systems, but also routes the packets through the most efficient path.

A NetWare router can also connect cabling systems that use different kinds of transmission media and different addressing systems.

For example, a NetWare router can connect a network using the Ethernet addressing structure and RG/58 coaxial cable to another network using the ARCnet addressing structure and RG/62 coaxial cable.

**Local vs. remote**

When a router is used within the cable length limitations for its LAN drivers, it is a local router. If the router is connected beyond its driver limitations or through a modem, it is a remote router.

**Router Information Protocol (RIP)**

A protocol that provides a way for routers to exchange routing information on a NetWare internetwork.

RIP allows NetWare routers to create and maintain a database (or router table) of current internetwork routing information.

Workstations can query the nearest router to find the fastest route to a distant network by broadcasting a RIP request packet.

Routers send periodic RIP broadcast packets containing current routing information to keep all routers on the internetwork synchronized. Routers also send RIP update broadcasts whenever they detect a change in the internetwork configuration.

By default, a NetWare router sends RIP packets to each of its connected network segments every 60 seconds.

Routes that do not appear in these periodic broadcasts (because a router has failed) are aged. After a certain period of time (default: 3 minutes), routers delete the aged routes from their router tables.

To reduce traffic on lower bandwidth (X.25 and asynchronous) segments, network supervisors can configure routers to send only RIP updates rather than periodic RIP broadcasts over those segments.

However, turning off the periodic RIP broadcasts can cause inconsistencies on the internetwork. For example, if an unreliable segment loses a RIP update packet, routers on that segment will broadcast old information.

The nwcm utility allows network supervisors to configure RIP broadcasts for each network segment.

To avoid inconsistencies in broadcast and aging intervals, all routers on the same network segment must have the same RIP configuration.

See also “Router”; “Service Advertising Protocol (SAP).”

---

## S

### **Salvageable files**

Files saved by NetWare, after being deleted by users, that can be salvaged (recovered). NetWare Services does not support salvageable files.

### **SAP**

See “Service Advertising Protocol (SAP).”

### **SAP daemon**

See “Service Advertising Protocol daemon (SAPD).”

### **SCSI**

See “Small Computer Systems Interface (SCSI).”

### **SCSI bus**

See also: “Small Computer Systems Interface (SCSI); “Hard disk.”

### **Search drive**

A drive that is searched by the operating system when a requested file is not found in the current directory.

Search drives are supported only from DOS workstations and are similar to the drives specified in the DOS path command.

A search drive allows a user working in one directory to access an application or data file located in another directory.

See also: “Drive mapping.”

## Search modes

Methods of operation that specify how a program will use search drives when looking for a data file.

When a DOS .EXE or .COM file requires an auxiliary file, it makes an open request through the operating system. The request may or may not specify the path to that file.

If a path is specified, the operating system searches that path. Otherwise, it only searches the default directory.

If the file is not found, the NetWare requester uses the search mode of the executable file to determine if it should continue looking for the file in the search drives.

FLAG allows you to set the search mode for executable files individually, or you can set the requester's search mode in the NET.CFG file for the majority of your files.

The following table describes the type of search modes.

**Table 1-15**

**Search modes**

<b>Mode</b>	<b>Description</b>
0	The default setting for all executable files. The executable file looks for instructions in the NET.CFG file.
1	The executable file searches the path specified in the file itself. If there is no path, the file searches the default directory, and then all search drives.
2	The executable file searches the path specified in the file itself. If there is no path, the file searches only the default directory.
3	The executable file searches the path specified in the file itself. If there is no path, the file searches the default directory; then if the open request is read only, the file searches the search drives.
4	Reserved.
5	The executable file searches the path specified first, and then all search drives. If there is no path, the file searches the default directory, and then all search drives.

Table 1-15

**Search modes**

Mode	Description
6	Reserved.
7	The executable file searches the path specified first. If the open request is read only, the file searches the search drives. If there is no path, the file searches the default directory, and then all search drives.

For example, if you assign an executable file mode 2, it won't use search drives.

If you assign mode 5, the executable file can use search drives to find a data file if the file isn't found in the first directory the executable file looks in.

Related utility: "FLAG" (*Utilities Reference*).

**Secondary time server**

A server that obtains the time from a Single Reference, Primary, or Reference time server and provides the time to workstations.

See "Time synchronization."

**Security**

The control of access to the network or to specific information on the network. Security under NetWare Services requires security of both the NetWare and the HP-UX portions of the system.

Categories of NetWare security features are

NetWare Login Security. Controls who can access the network through NetWare. See "NetWare login security."

Trustees Security. Designates who can access directories, files, or objects. See "Trustee security."

Rights Security. Determines the level of access for each trustee. See "Rights security."

**Inheritance Security.** Passes rights from higher to lower levels. See “Inheritance security.”

**Attributes Security.** Describes characteristics of directories and files. See “Attributes security.”

**Effective rights Security.** Determines a user’s actual calculated rights to a directory, file, or object. See “Effective rights security.”

Two categories of HP-UX security are

**HP-UX Login Security.** Controls who can access the network through HP-UX. See “HP-UX login security.”

**HP-UX Permissions Security.** Determines the level of access to files and directories of each user and group. See “HP-UX permissions security.”

Access control determines whether HP-UX permissions, NetWare rights, both, or neither control the access to files and directories. The various security features can be chosen per volume.

### **NetWare login security**

The LOGIN command controls who can access the network by determining if a valid user is attempting to log in.

A person must know the User object’s name and the correct password (if required) to log in.

The network supervisor establishes this login security by creating a User object in NetWare Directory Services and by then assigning values to the properties of that user. Those values determine how the user can access the network.

A User object’s properties affect when a user can log in, which workstations can be used, when the user’s account is disabled, and information about the user’s password, among other things.

Passwords aren’t required, but they should be used. Without one, an intruder can access the network with only a user’s name. Do not use family or pet names as passwords; they are easily guessed by an intruder.

Passwords are encrypted and are never displayed on the monitor or transmitted across the network. The password authenticates every action of a user.

## S

You can assign and change passwords, or assign initial passwords and allow users to change them. To increase login security, consider requiring these password options:

- Minimum password length. Prevents the use of passwords that might be easily guessed. Default: 5 characters.
- Periodic change in the password. Prevents keeping a password indefinitely. Default: every 90 days.
- Unique password. Prevents alternating between a few favorite passwords. The server remembers and rejects the use of the eight passwords most recently used for one day or longer.

### **Trustee security**

A trustee is a User or Group object that has been granted access to a directory, file, or object.

Access is granted through a trustee assignment. A trustee assignment says, in effect, “This user can access this directory, file, or object in these ways.”

Any object with sufficient rights can make trustee assignments with the “RIGHTS”, “NETADMIN”, or “NetWare Administrator” utilities.

- Trustee List. Each directory, file, and object has a list of trustee assignments, called a trustee list, that specifies who can access that directory, file, or object. An object’s trustee list is stored in the object’s ACL property. A directory or files trustee list is stored in the trustee data base of each NetWare volume.
- Trustees of groups. For several users to access a directory, file, or object, a trustee assignment is required for each user. Rather than make trustee assignments for each user, create a Group object, include the users in the group, and then grant access for the group with one trustee assignment.
- [Public] trustee. [Public] is a special trustee that can be added to an object, directory or file. The rights assigned to [Public] are effective for anyone who has no rights to the file, directory, or object.

### **Rights security**

Rights assigned to a NetWare Directory Services (NDS) object control the access the NDS object has to directories, files, or other NDS objects.

Creating, reading, and other operations can be done only if an object has rights to perform them.



Rights are granted to a specific directory, file, or NDS object by trustee assignments. An object with a trustee assignment to a file, directory, or NDS object is a trustee of that file, directory, or object.

Within each NDS object is an Access Control List (ACL) that defines who has rights to the object. Files and directories contain similar information, but not in the form of ACLs.

An NDS object's ACL defines what has rights to the object but does not define what the object has rights to. For example, a printer object's ACL defines which users have rights to the printer, but a user object's ACL does not define which printers the user has rights to.

This differs from the file system. Rights to a file can be designated in the user object. However, rights to a file can also be granted in the file itself.

There are four kinds of rights in NetWare:

- Directory rights. Control what a trustee can do with a directory. Directory rights also apply to files in the directory if file rights aren't granted and if the file's Inherited Rights Filter does not block the directory rights.
- File rights. Control what a trustee can do with a file.
- Object rights. Control what a trustee can do with an object. These rights control the object as a single piece in the directory tree, but don't allow access to information stored within that object (unless the Supervisor object right is granted).
- Property rights. Control a trustee's access to information stored within the object—that is, the information stored in the object's properties. Each object has several properties.

A default set of property rights can be granted to a trustee for all properties in the object. Rights can also be granted that override the default property rights of that trustee.

To grant directory or file rights to other objects, an object must have the Access Control right to that directory or file. To grant object or property rights, an object must have the Write right to the object's ACL property.

For a list and description of all rights, see "Rights."

Related utilities: "RIGHTS"; "NETADMIN"; "NetWare Administrator" (*Utilities Reference*).

### **Inheritance security**

By inheritance, rights granted by a trustee assignment apply to everything below the point where the assignment is made, unless another trustee assignment is made or unless the rights are blocked by an Inherited Rights Filter (IRF).

Inheritance applies both to directories and files on a volume, and to objects in the Directory tree.

For directories and files, all access rights are inherited. For objects, only object and property rights are inherited. When property rights are inherited all property right are inherited. Individual properties of an object can't be inherited.

Rights are also not inherited from a volume object in the Directory tree to the root directory of a volume, and directory rights must be assigned separately from object rights.

Inherited Rights Filter. If you were to create a file, but didn't want everyone who has rights in the directory to have rights to your file, you could create a filter that stops those rights from being inherited.

An Inherited Rights Filter has the same set of possible rights as a trustee assignment, but instead of granting rights, it revokes rights. Its effect is this: "Whatever rights to this file, directory, or object you would have inherited, I am revoking all but these rights."

Every directory, file, and NDS object has an Inherited Rights Filter. With this filter, you can grant access more freely at the top of the object tree or volume; then filter out rights in sensitive areas.

When all rights in a sensitive area are blocked by an Inherited Rights Filter, no one can inherit rights. Only users with a trustee assignment in that area have access.

The Supervisor right is unique.

If a trustee is granted the Supervisor right to a directory, that trustee inherits the Supervisor right for all subdirectories and files—the rights can't be blocked by another trustee assignment or an Inherited Rights Filter.

But if the Supervisor right is granted for objects and properties, it can be blocked by an Inherited Rights Filter, like any other right can.

Be careful not to block everyone's rights to an object with an Inherited Rights Filter, leaving no one with access to part of the Directory tree.

The utilities don't allow you to block the Supervisor object right unless a trustee already has the Supervisor object right at that point. But you could still delete the trustee object, making the trustee assignment invalid and cutting off access to that part of the Directory tree.

### **Attributes security**

Attributes (also called flags) describe the characteristics of a directory or file and tell NetWare what actions are allowed, and in a few cases, what actions have been performed. They aren't used for objects.

NetWare reads the attributes the user sets (for example, Delete Inhibit) and sets other attributes to show what has been done (for example, Archive Needed).

Attributes are separate from rights. Attributes aren't inherited, and if an attribute indicates that a file can't be deleted, not even a network supervisor can delete it without first changing the attribute.

To change the attributes of a directory or file, an object must be granted the Modify right in a trustee assignment for the directory or file.

For a list of attributes, see "Attributes."

Related utilities: "FLAG"; "FILER"; "NETADMIN"; "NetWare Administrator" (*Utilities Reference*).

### **Effective rights security**

Effective rights are the rights that a user actually has to a directory, file, or NetWare Directory Services (NDS) object.

NetWare calculates your effective rights to a directory, file, or NDS object whenever you initiate an action.

Effective rights to a file, directory, or NDS object are determined by

- An object's trustee assignments to the directory or file.
- Inherited rights from an object's trustee assignments to parent directory. (Because trustee assignments in the current directory or file override inherited rights from the parent directory, this only applies if the object does not have

## S

trustee assignments to the current directory or file.)

- Trustee assignments of Group objects that a User object belongs to.
- Trustee assignments of objects listed in a User object's security equivalences list.

Trustee assignments to a group are added to individual user trustee assignments.

No rights are granted by default. They must be granted by a trustee assignment at some point.

The Supervisor right can be masked for object and property rights, but can't be masked for file system rights.

### **HP-UX login security**

The first line of HP-UX security is the login and password. To access the HP-UX system, you must have a HP-UX login name and password and enter them when prompted.

### **HP-UX permissions security**

For each file and directory in the HP-UX system one of three permission levels can be set: read (r), write (w), and execute (x). Each HP-UX file or directory also has an owner, who along with the HP-UX system supervisor (root user) controls these permissions.

Each of the HP-UX permissions can be set individually for three different sets of users: the file or directory owner, the group to which the owner belongs, and all other users.

See also "Attributes"; "Effective rights"; "HPFS"; "Inherited Rights Filter, file system"; "Inherited Rights Filter, NDS object"; "Rights"; "Security equivalence"; "Trustee."

### **Security equivalence**

A property of every User object that lists other objects. The user is granted all rights that any object (User, Group, Printer, etc.) in that list is granted, both to objects and to files and directories.

Use security equivalence 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 equivalence.

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

Use security equivalence with caution. If users have rights to add to their own security equivalence list, they could add the name of a network supervisor and change anything on the network. Be careful when granting the Write or Supervisor property right to this property, and consider blocking it in the Inherited Rights Filter of each User object. This way, only network supervisors and those granted specific rights to this property can add to the list.

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 complete name. Because of this, you can make a container a trustee.

Every object in that container will have the rights that are granted to the container, through security equivalence. None of these containers are listed in a users security equivalence list, however.

Security equivalence 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. Security equivalence only grants Tom those rights that Jill is explicitly granted.

To add an object to a user's security equivalence list, you must have at least the Write property right to the ACL property of the object you want to add to the list. You don't need rights to the security equivalence property of the user; only the Browse object right.

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

Related utility: "NETADMIN" (*Utilities Reference*).

See also "User object."

## **Semaphore**

A flag that coordinates activities of both programs and processes to prevent data corruption in multiprocess or multiuser environments.

Semaphores are similar to logical locks in that they lock a certain string. However, unlike logical locks, semaphores allow more than one user to control the lock at one time.

For example, semaphores can allow a specified number of users access to a resource, such as to network applications with limited-user licenses. When the specified number is reached, the semaphore denies access to additional users.

See also “File locking”; “Record locking”; “Synchronization services.”

## **Sequenced Packet Exchange (SPX)**

A NetWare DOS Requester module that enhances the IPX protocol by supervising data sent out across the network.

See also: “Sequenced Packet eXchange II (SPXII).”

## **Sequenced Packet eXchange II (SPXII)**

An update to the SPX protocol.

Two characteristics differentiate SPXII from SPX:

- Larger packet sizes

Applications that used SPX were responsible for determining the size of the packet that could be used. In contrast, when an application uses SPXII, the SPXII driver is responsible for packet size negotiation.

- A windowing protocol.

The SPXII windowing protocol uses the existing header fields and allows the transmitter to send multiple packets before requesting an acknowledgment.

The receiver determines and maintains the window size for its half of the session.

The receiving endpoint is passive, that is, it only sends acknowledgments when the transmitting endpoint sets the ACK bit in the header.

SPXII maintains full backward compatibility with the existing SPX. The interface is the same, but taking full advantage of SPXII requires minor changes to user applications. TLI (Transport Level Interface) is the only supported application interface to SPXII.

SPXII verifies and acknowledges successful packet delivery to any network destination by requesting a verification from the destination that the data was received.

The SPXII verification must include a value that matches the value calculated from the data before transmission. By comparing these values, SPXII ensures not only that the data packet made it to the destination, but that it arrived intact.

SPXII can track data transmissions consisting of a series of separate packets. If an acknowledgment request brings no response within a specified time, SPXII retransmits it.

If a reasonable number of retransmissions does not get a response, SPXII notifies the application that the connection has failed.

SPXII is derived from Novell's SPX using the Xerox Sequenced Packet Protocol.

See also: "NetWare DOS Requester."

### **Serial port**

A port that allows data to be transmitted asynchronously, one bit at a time. Typically, serial ports are used for modems or serial printers.

On IBM PC-compatible computers, COM1 and COM2 are asynchronous serial ports.

### **Server console**

Under NetWare Services, a device (/dev/osm), which is the default device on which system messages are displayed. The output of these messages may be changed to another device or file by changing the nwcm parameter console\_device.

If you want to have the messages appear on your current terminal when the server is started, set console\_device to /def/fd/1 as follows:

```
nwcm -s console_device=/dev/fd/1
```

Since NetWare Services operates as a set of daemons, input to the system is not possible through the console device. All maintenance and server commands are executed through appropriate utilities (for example, “NetWare Server Status” and “Directory Services Repair”).

Related utility: “nwcm” (*Utilities Reference*).

### **Server protocol**

Procedures that a NetWare server follows to accept and respond to workstation requests.

See “NetWare daemon.”

### **Service Advertising Protocol (SAP)**

A protocol that provides a way for service nodes, such as NetWare servers, print servers, NVT2 servers and gateway servers, to register their services and addresses on the IPX internetwork. Clients can then query these tables for available services and their IPX address.

Routers and servers maintain a server information table and can be referred to as SAP agents. The Server Information Table contains, among other fields:

- Server name
- Server type
- IPX internet address
- Hops to server

Servers advertise their services with SAP, allowing SAP agents (SAs) to create and maintain a database of current internetwork server information.

SAs send periodic SAP broadcasts to keep all SAs on the internetwork synchronized. SAs also send SAP update broadcasts whenever they detect a change in the internetwork configuration.

Workstations can query the network to find a server by broadcasting SAP request packets. When a workstation logs in to a network, it broadcasts a “Get Nearest Server” SAP request and attaches to the first server that replies.



To keep workstations from attaching to a server, network supervisors can turn off the “Get Nearest Server” SAP option.

By default, an SA sends SAP packets to each of its connected network segments every 60 seconds.

Related utilities: “nwsapinfo(1M)”; “nwsaputil(1M)”.

See also: “Router”; “Router Information Protocol (RIP)”; “Service Advertising Protocol daemon (SAPD).”

### **Service Advertising Protocol daemon (SAPD)**

A daemon that performs the functions of the SAP agent that are independent of NetWare services.

When created by the NPS daemon, the SAP daemon opens a stream to IPX and sends a request to all SAP agents to send their server information.

From the response packets, the SAP daemon forms a Server Information Table. The SAP daemon then maintains that table, adding, deleting, and modifying server information as services are added, deleted, or rerouted on the network.

UNIX system services can advertise their availability by sending the SAP daemon a periodic broadcast packet. The SAP daemon adds these services to its Server Information Table, informs other SAP agents that a new service is available, and then responds to all queries about these services.

The services remain in the table as long as the services send the SAP daemon a broadcast every 60 seconds. When a service informs the SAP daemon that it is going down, the SAP daemon removes the service from its table, informs other SAP agents that the service is going down, and then quits responding to queries about the service.

See also “Service Advertising Protocol (SAP).”

### **Shareable attribute**

A file system attribute that allows a file to be accessed by more than one user at a time.

See “Attributes.”

### **Shared memory**

A pool of memory that HP-UX processes have access to. Shared memory is divided into pools and controlled by internal locking mechanisms.

NetWare Services stores the following in shared memory:

- Connection table
- Synchronization data structures
- Trustee database
- NetWare server information
- Volume table

### **Short machine type**

A four-letter (or less) name representing a DOS workstation brand. The short machine type is similar to the long machine type, except the short machine type is used specifically with overlay files.

Files using the short machine type include the IBM\$RUN.OVL file for windowing utilities and the CMPQ\$RUN.OVL file that uses a default black-and-white color palette for NetWare menus.

The short machine type is set in the NET.CFG file, using the SHORT MACHINE TYPE parameter. The default is IBM.

The short machine type can be accessed in login scripts, using the %SMACHINE identifier variable.

See also “Login scripts”; “Long machine type.”

### **Single Reference time server**

A server that provides time to Secondary time servers and to workstations. The Single Reference time server is the sole source of time on the network.

See “Time synchronization.”

### **Small Computer Systems Interface (SCSI)**

Commonly pronounced scuzzy. An industry standard that sets guidelines for connecting peripheral devices and their controllers to a microprocessor.

The SCSI interface defines both hardware and software standards for communication between a host computer and a peripheral.

Computers and peripheral devices designed to meet SCSI specifications have a large degree of compatibility.

### **Socket**

The part of an IPX internetwork address, within a network node, that represents the destination of an IPX packet.

Some sockets are reserved by Hewlett-Packard for specific applications. For example, IPX delivers all NCP request packets to socket 451h.

Third-party developers can also reserve socket numbers for specific purposes by registering those numbers with Hewlett-Packard.

Some key socket numbers reserved by Hewlett-Packard:

<b>Socket</b>	<b>Process</b>
451h	NCP
452h	SAP
453h	RIP
455h	NetBIOS
456h	Diagnostics
8063h	NVT2 (Novell Virtual Terminal 2)
4000h-7FFFh	Temporary sockets used for interaction with NetWare servers and other network communications

See also "IPX internetwork address."

**S**

**SPX**

See “Sequenced Packet Exchange (SPX).”

**SPX2**

See “Sequenced Packet eXchange II (SPXII).”

**SPXII**

See “Sequenced Packet eXchange II (SPXII).”

**Station**

Usually a shortened form for workstation, but can also be a server, router, printer, fax machine, or any computer device connected to a network by a network board and a communication medium.

**Station address**

A number that uniquely identifies a network board; usually referred to as the node number.

See “Node number.”

**STREAMS**

A general, flexible facility and a set of tools for development of UNIX communication services. It supports the implementation of services ranging from complete networking protocol suites to individual device drivers. STREAMS defines standard interfaces for character input/output within the kernel, and between the kernel and the rest of the UNIX system.

NetWare Services drivers are STREAMS drivers.

**Subdirectory**

A directory below another in the file system structure. For example, in SYS:ACCTS\RECEIVE, RECEIVE is a subdirectory of SYS:ACCTS.

See also “Directory structure, file system.”

**Supervisor right**

A NetWare file system right that grants all rights to the respective directory and files.

Also, an object right that grants all privileges to all objects in that container.

Also, a property right that grants all rights to the property.

See “Rights.”

**Switch block**

A set of switches mounted to form a single component.

In some computers, a switch block is used to control system configuration data, such as type of monitor, amount of memory, and number of drives.

Network boards often use switch blocks to set system addresses (such as station, base I/O, and base memory addresses).

**Synchronization**

Replica synchronization. A means of ensuring that replicas of a Directory partition contain the same information as other replicas of that partition. See “Replica.”

Time synchronization. A method of ensuring that all servers in a Directory tree report the same time. See “Time synchronization.”

**Synchronization services**

Allows concurrent access to files and data structures and regulates access to resources on the NetWare Services server. These services are provided to DOS and Windows clients.

---

**NOTE:**

---

Normally, only a client can invoke Synchronization Services. However, the NetWare server will occasionally issue synchronization services calls to perform internal synchronization on critical data.

NetWare Services provides a simple synchronization mechanism through the mode in which the file is opened. For example, opening a file in deny-write mode would prevent other clients from modifying the file, but would still allow them to read the file. This method of file synchronization is probably adequate for most simple network applications.

Finer-grained synchronization is provided through four kinds of synchronization services:

- Whole file synchronization (file locks)
- File segment synchronization (physical or byte-range locks)
- Logical entity synchronization (logical locks)
- Semaphore synchronization (semaphore locks)

**Whole file synchronization.** When a client issues a file lock on a file, synchronization occurs on the entire file. While a file lock is held, no other clients can access the file for read or write.

**File segment synchronization.** This occurs when a client issues a “physical lock” on a file. Physical locks allow a client to place a shared or exclusive lock on a particular range of bytes within the file. This range may extend beyond the current end of file.

**Logical entity synchronization.** Logical entity synchronization occurs when a client issues a “logical lock” on a server entity. A logical lock is a label in the form of an ASCII string that represents network data and indirectly controls access to that network data.

**Semaphore synchronization.** Semaphores, like logical records, are labels in the form of ASCII strings that individually control network activity. A value is set when the semaphore is open that indicates how many client can access the semaphore’s resource at one time.

The NCP engines handle initial lock requests, which completes if there is no contention for the resource. If the lock is successful, the engine signals the client. If the lock is not successful, the engine hands it off to one of four lock

daemons: the file lock daemon, the physical lock daemon, the logical lock daemon, and the semaphore daemon. An additional timer daemon keeps accurate time for lock aging.

See “File locking”; “Record locking”; “Semaphore.”

### **System attribute**

A file system attribute that marks directories or files for use only by the operating system.

See “Attributes.”

### **SYSTEM directory**

The SYS:SYSTEM directory, created during NetWare installation, that contains NetWare utilities for managing the network.

---

**WARNING:**

---

**Do not delete the SYSTEM directory.**

See also “Directory structure, file system”; “LOGIN directory”; “MAIL directory”; “PUBLIC directory.”

### **System login script**

A type of login script that sets general environments for all users in an Organization or Organizational Unit.

System login scripts are optional; if used, they execute first, before profile and user login scripts.

See also “Login scripts.”

## T

### **Tape backup unit**

Typically, an external tape drive that backs up data from hard disks.

### **TCP/IP**

See “Transmission Control Protocol/Internet Protocol (TCP/IP).”

### **Terminal emulation software**

Software that duplicates the communication protocol of a dedicated terminal, connecting workstation users to the UNIX environment.

A workstation running terminal emulation software acts as if it were wired directly to the UNIX terminal ports. NetWare's terminal emulator, NVT2, sends and receives data from NetWare workstations

Intelligent workstations using terminal emulation must be connected to the host. With NetWare Services, workstations using terminal emulation are connected via the network, greatly increasing the performance of terminal emulation as well as simplifying the installation of terminal lines.

NVT2 supports a variety of third-party terminal emulation programs.

See also “NVT2.”

### **Time synchronization**

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 and provides a timestamp to order NetWare Directory Services events.

Whenever an event occurs in the Directory, such as when a password is changed, or an object is renamed, NetWare Directory Services requests a time stamp.



NetWare Directory Services uses time stamps to

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

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

Replication allows partition updates to originate from many locations. As various users update the partition’s replicas, some updates will inevitably pertain to the same data.

For example, a user might delete an object and then recreate it. But without a method of recording the order of these events, the Directory could try to create the object and then delete it.

Time stamps allow the Directory to reproduce the “real world” order of events.

By default NetWare Services, using the NetWare Time Synch daemon (ntsd), handles time synchronization for the HP-UX system. If your HP-UX system is time synchronizing by another method, such as Network Time Protocol (NTP), you should disable the NetWare Services synchronization by configuring it as a Reference Time server (nwcm parameter `ts_type` set to “reference”).

You can change time synchronization parameters with the graphical NetWare Setup” utility or with `nwcm`.

### **Time server types**

NetWare servers can be designated as Single Reference, Primary, Reference, or Secondary.

If a NetWare Services server contains the root of the NetWare Directory Services tree, the server defaults to a Single Reference time server.

If a NetWare Services server does not contain the root of the NetWare Directory tree, the server defaults to a Secondary time server.

Each time server type performs a particular time synchronization function:

- Single Reference time server. Provides time to Secondary time servers and to

workstations.

This server determines the time for the entire network and is the only source of time on the network. The network supervisor sets the time on the Single Reference time server.

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 workstations. The Secondary time servers, in turn, provide time to their workstations.

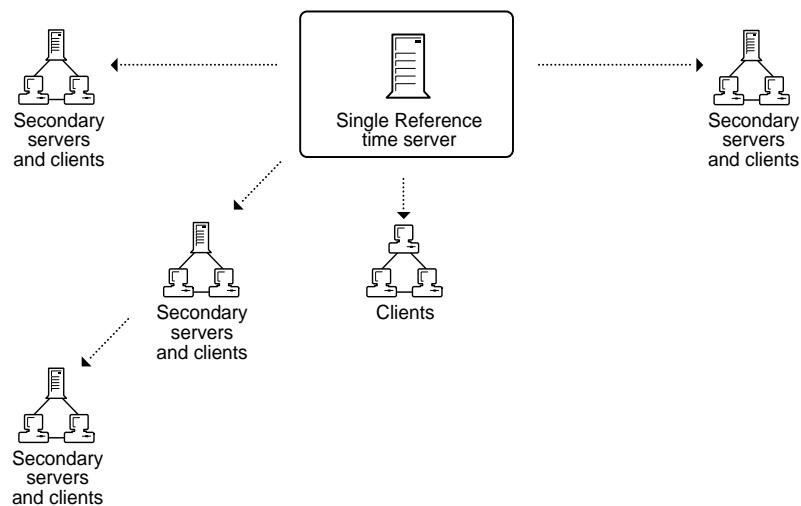


Figure 1-46

**Single Reference time server**

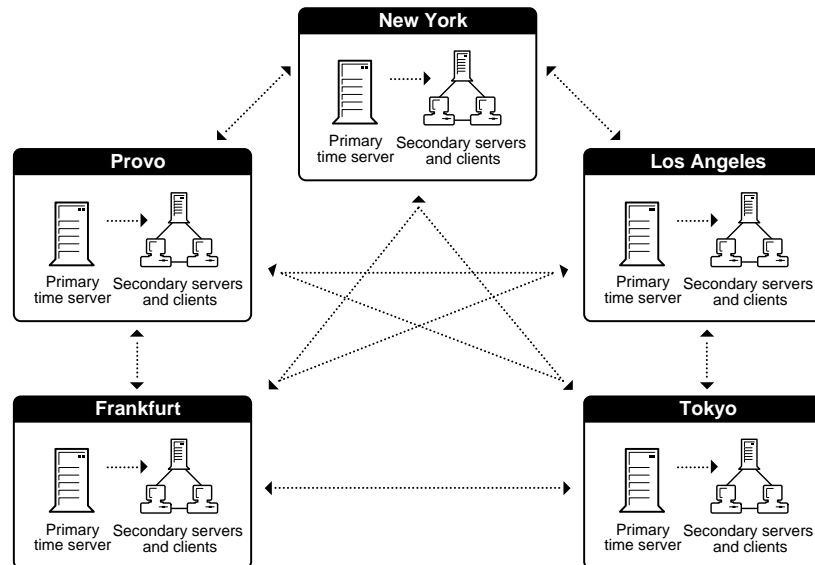
The Single Reference time server works on networks of any size, but this configuration is used primarily for small networks that are not WANs.

If you use a Single Reference time server, do not use any Primary or Reference time servers on the network.

- Primary time server. Synchronizes the time with at least one other Primary or Reference time server, and provides the time to Secondary time servers and to workstations.

Primary time servers also “vote” with other Primary or Reference time servers to determine what the common network time should be.

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 workstations.



**Figure 1-47**

### Primary time servers

Use the Primary time server 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.

Place a Primary time server in each geographically distinct area so that secondary servers and workstations can access them without using WAN links.

You must have at least one other Primary or Reference time server that the 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.

However, Primary servers do adjust their internal clocks to synchronize with that common network time. Because all Primary servers adjust their clocks, network time may drift slightly.

- Reference time server. Provides a time to which all other Primary time servers and workstations synchronize.

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

Reference time servers “vote” with other Primary or Reference time servers to determine what the common network time should be.

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

Therefore, a Reference time server acts as a central point to set network time. Eventually, all Primary time servers will adjust their clocks to agree with a 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 workstations, as well as to the Primary time server at another location.

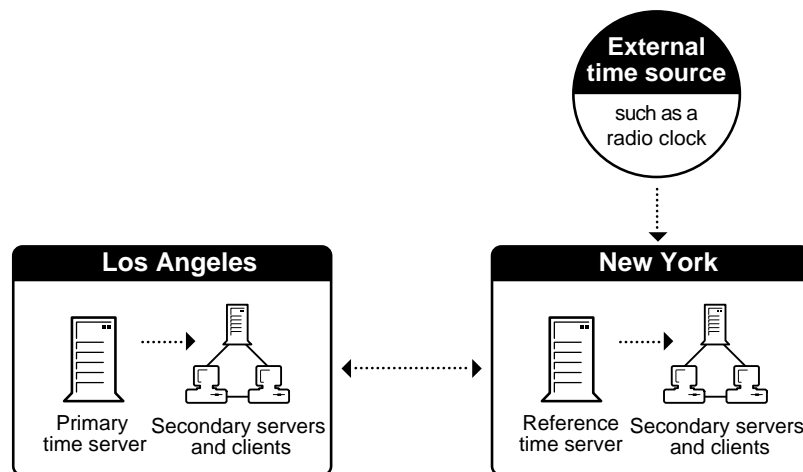


Figure 1-48

**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.

You can use more than one Reference time server on a network, but you must synchronize each Reference time server with an external time source, such as a radio clock.

You must have at least one other Primary time server that the Reference time server can contact.

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

- Secondary time server. 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 workstations.

A Secondary time server does not participate in determining 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, connect Secondary time servers to Primary or Reference time servers that are physically nearby.

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.

### **SAP and custom configuration**

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

- SAP (Service Advertising Protocol). By default, Primary, Reference, and Single Reference servers use SAP to announce their presence on the network.

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

Secondary time servers use SAP information to pick a time server to follow.

An advantage of SAP is that it allows 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.

A disadvantage of the SAP method is that a small amount of additional network traffic is generated.

## T

Custom configuration. You can list the specific time servers that a particular server should contact.

You can also specify that a server shouldn't listen for SAP information from other time sources, and that it is not to advertise its presence using SAP

An advantage of custom configuration is that the network supervisor maintains complete control of the time synchronization environment.

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

A disadvantage of custom configuration is the increased time required for planning and installation.

Also, it is more difficult to install or remove Primary, Reference, or Single Reference time servers. You must manually change the approved server list maintained on each server.

### **Which time synchronization method to use**

On small networks where it is unlikely that servers will be added or reconfigured after initial installation, we recommend that you use a Single Reference time server using SAP (the installation default).

On larger networks, or on networks subject to frequent accidental reconfiguration when servers are added or removed, custom configuration is recommended.

Related utilities: "NetWare Setup"; "nwcm"; "SYSTIME" (*Utilities Reference*).

### **Topology**

The physical layout of network components (cables, stations, gateways, hubs, etc.). There are three basic topologies:

- Star network. Workstations are connected directly to a NetWare server but not to each other.
- Ring network. The NetWare server and workstations are cabled in a ring; a workstation's messages may have to pass through several other workstations before reaching the NetWare server.
- Bus network. All workstations and the NetWare server are connected to a central cable (called a trunk or bus).

**Transmission Control Protocol/Internet Protocol (TCP/IP)**

An industry-standard suite of networking protocols, enabling dissimilar nodes in a heterogenous environment to communicate with one another.

TCP/IP is built upon four layers that roughly correspond to the seven-layer OSI model. The TCP/IP layers are

- Process/application
- Host to host
- Internet
- Network access

**Transaction Tracking System (TTS)**

A system that protects database applications from corruption by backing out incomplete transactions that result from a failure in a network component.

TTS is not supported in NetWare Services, but is emulated for the Directory Services database.

**Transactional attribute**

A NetWare file system attribute that indicates the file is protected by TTS.

NetWare Services does not support TTS or use this attribute.

See "Attributes."

**Transmission Control Protocol (TCP)**

An industry-standard suite of networking protocols, enabling dissimilar nodes in a heterogenous environment to communicate with one another.

**Trustee**

A user or group granted rights to work with a directory, file, or object; the object is called a trustee of that directory, file, or object.

Rights are granted to objects (making them trustees) by trustee assignments. Trustee assignments are part of the directory, file, or object to which they grant access.

Trustee assignments are stored in a trustee list. An object's trustee list is stored in the object's ACL property. In NetWare Services a directory or file's trustee list is stored in the volume's trustee database.

For example, to make group WRITERS a trustee of directory PROJECTS, go to PROJECTS and make a trustee assignment with the name of group WRITERS.

[Public] is a special trustee. [Public] can be specified as the trustee of any file, directory, or object.

Anyone who tries to access a file, directory, or object without any other rights is allowed the rights granted to the [Public] trustee.

Trustee assignments for objects and for directories and files are made in the same way, but the rights granted by a trustee assignment are different for objects and for directories and files.

Rights flow down the tree structure for objects and for directories and files, but rights from objects never affect directories and files, or vice versa.

Exception: a trustee of a Server object with the Supervisor right is automatically granted the Supervisor right to the root directory of every volume attached to that server.

When you make a trustee assignment to a directory, file, or object, the trustee has access to the directory, its files, and its subdirectories (unless rights are redefined at the file or subdirectory level) or to the subordinate objects. This is called inheritance.

Through inheritance, rights granted to a trustee flow down through the directory structure unless one of the following is true:

- Other trustee assignments are granted for the same object at a lower level of the directory structure, or
- The Inherited Rights Filter of a subordinate object revokes rights granted in a trustee assignment above that point.

An explicit trustee of a directory, file, or object is an object that has a trustee assignment to that directory, file, or object.



A trustee through inheritance is an object that has a trustee assignment to a directory, file, or object higher in the structure and inherits rights for the current directory, file, or object.

Hole in the tree. A trustee assignment for a file or directory always allows the user to see the path to the root directory of the volume. A trustee assignment for an object, however, does not automatically show the user the directory tree to the root.

This security feature prevents a user who has rights in one branch of the tree from jumping over a place where the user does not have rights and browsing through the entire tree.

The Supervisor right, which can't be blocked by an Inherited Rights Filter on a file or directory, can be blocked by the Inherited Rights Filter for an object, either for object or property rights.

Use caution when blocking the Supervisor right to objects. If you delete the only object that has the Supervisor right to part of the tree and the Inherited Rights Filter blocks others from inheriting the Supervisor object right, that part of the tree is cut off.

To make trustee assignments for a directory or file, an object must be a trustee of that directory or file with the Access Control right.

To make a trustee assignment for an object, an object must be a trustee of that object with the Write or Add Self right to the object's ACL property.

To grant or modify a trustee assignment for either directories, files, or objects, use "FILER", "NETADMIN", "NetWare Administrator," or "RIGHTS" (*Utilities Reference*).

See also "Inherited Rights Filter, file system"; "Inherited Rights Filter, NDS object"; "Security"; "Trustee database."

### **Trustee database**

A database that contains information about trustee assignments and inherited rights filters (IRFs) in NetWare. Since NetWare checks trustee rights every time a user accesses a file or directory, quick access to the trustee information is essential. The trustee database stores the following information:

- Trustee assignments granted to directories or files
- Modified IRFs (each NetWare file and directory has an IRF which controls how rights granted with trustee assignments are enforced on descendent nodes)

In NetWare Services, each volume maintains its own database of trustee assignments in the volume's control directory in a hidden file (.trustee.sys).

When NetWare Services is initialized, these volume databases are loaded into shared memory in a per-volume memory structure. Since each volume's database is separately maintained both on disk and in memory, a volume can be dismounted and its database removed from memory without affecting the other volumes.

See also "Inherited Rights Filter, file system"; "Inherited Rights Filter, NDS object"; "Security"; "Trustee."

### **TTS**

See "Transaction Tracking System (TTS)."

---

## U

### **Unbinding**

The process of removing a communication protocol from network boards and LAN drivers.

See “Binding and unbinding.”

### **Unknown object**

A leaf object that represents a NetWare Directory Services object that has been corrupted and can't be identified as belonging to any of the other object classes.

See also “Object.”

### **Unicode**

A 16-bit character representation, defined by the Unicode Consortium, that supports up to 65,536 unique characters. Unicode allows you to represent the characters for multiple languages using a single Unicode representation.

All objects and their attributes in the NetWare Directory database are stored in their Unicode representation.

However, clients (including DOS) use 256-character code pages (using 8-bit characters). Not every character created using a given code page will display correctly on a workstation using a different code page.

(For more information on code pages, see your DOS manual.)

When you change code pages, you need a different set of Unicode translation tables in order to run NetWare utilities and manage the NetWare Directory database.

## U

For example, to use code page 850 (Europe) with country information for France (for which the international telephone country code is 33), you need the following Unicode files:

- 850\_UNI.033 — Translates code page 850 to Unicode
- UNI\_850.033 — Translates Unicode to code page 850
- UNI\_MON.033 — Handles monocasing (the proper alphabetization of upper- and lower-case letters)
- UNI\_COL.033 — Collation, sorted lists

For different code pages and locales, you need Unicode tables with corresponding code page numbers and country codes.

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

### **UNIX client**

A computer running the UNIX operating system that is connected to the network.

The UNIX client stores and retrieves data from the NetWare server and runs executable network files. The UNIX client provides multiple NetWare client multitasking on a single station.

UNIX clients use IPX/NCP protocols to access NetWare resources and NetWare clients use Novell Virtual Terminal 2 (NVT2) to access UNIX resources.

See also “Client.”

### **UNIX host locking**

The ability of the NetWare server to map NetWare client byte-range locks (physical locks) onto the UNIX operating system. This helps avoid conflicts when a NetWare application and a UNIX application are concurrently operating on the same file.

Host locking is activated using the configuration manager (nwcm).

When a NetWare client tries to lock a file, NetWare first makes sure that the lock request does not conflict with other NetWare locks. If it does not, NetWare attempts to lock the entire file under UNIX.

If any UNIX locks are on the file, the lock request fails. Many NetWare clients can set locks on the file or many UNIX processes can set locks on the file, but not both concurrently.

NetWare allows shareable and exclusive locks on files opened in read-only or read-write mode, but UNIX allows shareable and exclusive locks only when a file is opened in read-write mode.

If the UNIX file is opened in read-only mode, only shareable UNIX locks can be applied. If the UNIX file is opened write-only, only exclusive UNIX locks can be applied.

Therefore, to allow maximum interoperability between NetWare and UNIX, NetWare always opens files in read-write mode.

UNIX locks are only advisory; UNIX does not enforce them. Applications are responsible for checking locks and enforcing them.

### **UNIX operating system**

A computer operating system originally developed at Bell Labs in 1969. UNIX features include

- Multiuser functionality
- Multitasking
- Platform independence
- Built-in communication tools
- Application portability
- Password and file access security

See also: "HP-UX."

## **HP-UX**

The operating system developed by the UNIX Systems Group at Hewlett-Packard., that provides all the power and reliability of a traditional UNIX operating system in an easy-to-use, graphical user interface (GUI) environment.

### **User login script**

A type of login script that sets environments specific to a user. Use user login scripts to contain items that cannot be included in system or profile login scripts.

User login scripts are optional; if used, they execute after system and profile login scripts.

See also “Login scripts.”

### **User object**

A leaf object in NetWare Directory Services that represents a person with access to the network. The following items are important to managing User objects.

#### **Login name**

The login name is the name the user logs in with. A login name is mandatory when creating a User object.

You can have a login name of up to 127 characters; however, for efficiency, use one of the following conventions when creating login names:

- Given name (for example, user JANE).
- Surname (for example, user DOE).
- Initials and surname (for example, user JDOE).

Do not use special characters or control characters in a login name. Spaces can be used but aren't recommended.

When you use a login name with spaces in a login script, enclose the entire login name within quotation marks. (To avoid this, use underscores in the login name rather than spaces.)

**Group membership**

You can assign a user to Group objects. When added to a group, a user inherits the rights assigned to that group.

**Home directories**

A home directory serves as a user's personal workspace.

If you create home directories, plan a parent directory (such as SYS:HOME or SYS:USERS) for them. Or, for a large system, set aside a separate volume for users' home directories.

To simplify system administration, make each user's home directory name the same as that user's login name (for example, SYS:USER\JANE, SYS:HOME\RDSMITH, and so forth).

If you grant all trustee rights to users in their own directories, users can control access to files in their directories. This allows users to work on projects in their home directory and prevent others from accessing their work.

Once the work is completed, the files can be copied to a work or project directory (group work space) where other users can access the information.

**Trustee rights**

If users need to access specific directories and files (other than those assigned by the system), you must grant users trustee rights to these directories.

**Security equivalences**

Security equivalences allow users to exercise rights equivalent to those of another user.

Assigning security equivalences is convenient when you need to give a user access to the same information that another user already has access to.

In networks that contain confidential data for selected users, make sure that you don't inadvertently give a user access to restricted information.

Use particular caution when granting a user security equivalence to a user who has Supervisor rights.

## U

### **User login scripts**

These configurable batch files customize the network environment for users by initializing environment variables, mapping drives, and executing other commands.

Up to three login scripts are used at login, executed in the following order:

- The login script of the user's immediate container.
- The login script in a profile object specified for that user.
- The user's individual login script.

Where there are conflicting or contradicting commands in login scripts, the commands executed in the last login script are effective.

If no login scripts exists, a default login script is executed that maps a drive to NetWare utilities.

### **Print job configurations**

Each user can use printing defaults, or you can create print job configurations for a container or User object.

### **Account management**

Any user who has the Supervisor object right to another User object manages that User object and can modify information about that user.

Users who don't have the Supervisor object right can be granted rights to other User objects to fulfill specific responsibilities. For example, the network supervisor may want only the phone book manager to be granted rights to each user's phone number property.

### **User account restrictions**

Every user account can be restricted to prevent unauthorized users from accessing the network. Some restrictions can even disable the account so that no one can log in as that user.

The network supervisor can restrict logins in the following ways:

- Account balance restrictions. If you install Accounting to monitor or limit network resources, you can assign account balances for users and specify credit limits. When the balance is depleted, the account is disabled.



If accounting hasn't been installed, this option isn't available.

- Expiration restrictions. You can specify an expiration date for a user account. The account expires at 12:01 a.m. the following day.

Any attempt to log in after the account expires disables the account. (Default: no expiration.)

- Password restrictions. You can require passwords.

If you require passwords, you can also specify the minimum length (default: 5 characters), how often the password must be changed (default: 40 days), whether the user can change the password (default: Yes), and whether the password must be unique (default: No).

For a password to be unique, it must be different from the previous eight passwords used by the account.

You can also specify the number of times a user can log in with an expired password (grace logins) or the number of incorrect login attempts (default: 7 times).

When either number is exceeded, the account is disabled.

See also "Security" (Login security).

- Connection restrictions. You can limit the number of workstations (connections) that a user can be logged in from at any one time.
- Time restrictions. You can restrict the hours and days during which users can log in. Times are specified in half-hour blocks. You can assign all users the same times, or you can restrict users individually. Default: no time restriction.
- Network address restrictions. You can restrict the physical locations that a user can log in from by specifying the network and node addresses of the workstation the user can log in from.

Workstation restrictions can't be set with system default restrictions; they must be assigned individually. If no network address restrictions are listed, no station restrictions are in effect.

Related utilities: "NETADMIN"; "NetWare Administrator" (*Utilities Reference*).

See also: "Creating Leaf Objects"; "Managing Groups of User Objects" and "Cautions When Deleting User Objects" (*Supervising the Network*); "Accounting"; "Group object"; "High Performance File System (HPFS)"; "Login scripts"; "Security equivalence."

## U

### User template

A file containing default information you can apply to new User objects to give them default property values. This helps if you are creating many users who need the same property values.

You create user templates in Organization or Organizational Unit objects.

When you create a User object, you can specify that you want to use a user template. In this case, the property values entered in the user template for that container (or the container above, if no user template exists in the current container) are copied into the new User object as it is created.

The user template saves you from re-entering information—a fax number, login time restrictions, addresses, password restrictions, language, etc.—that is common to every User object in a container.

When you create a user template in a container, you can copy information from the parent container's user template.

For example, if you create a template in OU=SALES.O=HEWLETT-PACKARD, you are asked whether you want to start by copying the user template (if one exists) from O=HEWLETT-PACKARD.

Using the parent container's user template saves re-entering similar information for lower-level containers.

A user template is actually a User object named USER\_TEMPLATE. You enter information in this object just as you do any other User object, although not all properties of a User object can be copied from a user template.

Information assigned to a new User object from a template can be changed after the User object is created.

However, you can't log in as USER\_TEMPLATE, grant rights with a user template, apply a user template to existing User objects, nor apply user template updates to User objects created with that template.

### Utilities

Programs that add functionality to the NetWare operating system. NetWare utilities support HP-UX, Windows, and DOS environments. See the *Utilities Reference* for detailed information about the NetWare Services utilities.

**Server utilities**

NetWare administrators use server utilities to maintain the network. These utilities are available from the server console or from a Remote Console™.

Tasks you can perform with server utilities include the following:

- Configuring the server
- Starting and stopping the server
- Checking server status
- Installing and deleting Directory Services (DS)
- Installing and deleting NetWare server user licenses
- Creating and deleting NetWare volumes
- Displaying statistical information

**Workstation utilities**

NetWare workstation utilities can be run from a DOS or Windows workstation.

Graphical utilities, like NetWare Administrator, allow network supervisors to manage the network through Microsoft Windows 3.0 and 3.1 Presentation Manager 2.0.

Text utilities for DOS support both bindery and NetWare Directory Services.

---

## V

### **Value-added server**

A separate, specialized, dedicated computer (such as a print server or a database server) that fulfills a specific function for network users.

### **Virtual Loadable Module (VLM)**

A modular executable program that runs at each DOS workstation and enables communication with the NetWare server.

A VLM file has a .VLM filename extension. For example, the IPX VLM file is IPXNCP.VLM.

The NetWare DOS Requester is composed of several VLMs. These VLMs replace, and provide backward compatibility with, NetWare shells used in previous NetWare versions.

There are two types of VLMs: child VLMs and multiplexor VLMs.

#### **Child VLMs**

Child VLMs handle a particular implementation of a logical grouping of functionality. For example, each NetWare server type has its own child VLM:

- NDS.VLM, for NetWare Directory Services-based (NetWare 4) servers.
- BIND.VLM, for bindery-based servers (prior to NetWare 4).
- PNW.VLM, for NetWare desktop-based servers.

Various implementations of transport protocols also have their individual child VLMs. For example, IPXNCP.VLM handles IPX services, and TCPNCP.VLM handles TCP functions.

#### **Multiplexor VLMs**

A multiplexor is a VLM that routes calls to the proper child VLM. Requester multiplexors can be considered parent VLMs, ensuring that requests to child VLMs reach the appropriate VLM module.

See also “NetWare DOS Requester.”

## **VLM**

See “Virtual Loadable Module (VLM).”

## **voltab**

The NetWare Services configuration file that contains all volume parameters. All other configuration parameters are stored in `nwconfig`. Parameters in `voltab` are set using the graphical utility NetWare Volume Setup.

Related utilities: “NetWare Volume Setup” (*Utilities Reference*).

## **Volume**

The highest point in a NetWare file system. Volumes contain directories, subdirectories, and files. In NetWare Services, each NetWare volume is a path to a point in the HP-UX file system. These paths are specified in the `voltab` file. The specified point in the HP-UX file system is the NetWare volume root node—the point at which the NetWare volume begins. See “Directory structure, file system.”

A volume is a logical unit. It behaves much as a hard disk behaves in a stand-alone system. NetWare users cannot see the HP-UX directory structure above the NetWare volume level. To HP-UX users, NetWare volumes appear as HP-UX subdirectories.

NetWare supports a maximum of 64 volumes.

### **Volume Names**

The first NetWare volume must be named `SYS:`. Other volumes may be given the same name as the UNIX directory or may be given different names. Volume names are 2 to 15 characters long. NetWare convention adds the server name in front of the volume name (`SERVER/SYS:`). The volume name is followed by a colon (:), as in `SERVER/SYS:PUBLIC`.

### **Volume Configuration**

To configure NetWare volumes, use the “NetWare Volume Setup” graphical utility to set the following volume properties:

- **Volume Name.** Specifies the name NetWare clients use to access the volume. The first volume is always SYS. Other volumes may have the same name as the HP-UX directory, or may be different. Names must be from 2 to 15 characters long.
- **Type.** Specifies whether the volume is Standard (portable to non-HP-UX systems), Extended (a kernel implementation of the NetWare file system that exists on an Extended type partition), or CDROM (volume is a CDROM and cannot be modified by NetWare or HP-UX users).
- **Name Spaces.** Specifies the client types the volume will support. DOS and UNIX name spaces are always supported.
- **Mount Point.** Specifies the path from the UNIX root to the directory where the volume begins.
- **Control Directory.** Specifies the location of the volume’s trustee database files and the desktop database. For Standard volumes, this specifies the location of the volume’s inodes, the extended file names, and the last mount log. Once modified, the NetWare inodes are moved to the new control path.
- **Attributes.** Specifies if the volume is Read and Write or Read Only.
- **File Access Control.** Determines which users have access to the files and directories. Can be NetWare (trustee assignments control a NetWare user’s access), UNIX (HP-UX permissions control a NetWare user’s access), Both (both NetWare trustee assignments and HP-UX permissions control access), None (all NetWare users can access the files and directories as if they had Supervisor rights.).

Related utilities: “NetWare Volume”; “nwcm” (*Utilities Reference*).

See also: “Configuration”; “Directory structure, file system”; “Directory tree”; “Directory path”; “File Systems, NetWare Services”; “File system, HP-UX”; “Multiple name space support”; “Partition management”; “Root user”; “Root directory”; “voltab.”

### **Volume object**

A leaf object that represents a physical volume on the network.

In the Volume object's properties, you can store information about which NetWare server the physical volume is located on and the volume name recorded when the volume was initialized at the server (for example, SYS:).

You can also store information such as the volume's owner, space use restrictions for users, or a description of its use.

See "Creating Leaf Objects" in *Supervising the Network*.

See also "Object"; "Volume."

## W

### **Wait state**

A period of time when the processor does nothing; it simply waits. A wait state is used to synchronize circuitry or devices operating at different speeds.

For example, wait states used in memory access slow down the CPU so all components seem to be running at the same speed.

### **WAN**

See “Wide Area Network (WAN).”

### **Watchdog**

Packets used to make sure workstations are still connected to the NetWare server.

If the server has not received a packet from a station in a certain time, a watchdog packet is sent to the station. If the station does not respond within a certain time, another watchdog packet is sent.

If the station still does not respond to a certain number of watchdog packets, the server assumes that the station is no longer connected and clears the station's connection.

The time period between watchdog packets and the number of watchdog packets sent is configurable using the NetWare Setup graphical utility or the nwcm command line utility.

Related utilities: “NetWare Setup”; “nwcm” (*Utilities Reference*).

### **Wide Area Network (WAN)**

A network that communicates over a long distance, such as across a city or around the world.



A local area network (LAN) becomes a part of a WAN when it's linked (by modems, remote routers, phone lines, satellites, or microwave) to a mainframe, public data network, or another LAN.

See also "Local Area Network (LAN)."

### **Windows client**

A workstation that boots with DOS and accesses the network through either

- The NetWare DOS Requester and its VLMs (for NetWare 4)
- A NetWare shell (for NetWare versions earlier than NetWare 4)

It also runs MS Windows and, with client software, performs such tasks as mapping drives, capturing printer ports, sending messages, and changing contexts in the Windows environment.

See also "Client."

### **Workstation**

A personal computer connected to a NetWare network and used to perform tasks through application programs or utilities. Also referred to as a client or shortened to station.

See also "Client"; "Station."

### **Write right**

A file system right that grants the right to open and write to files.

Also, a property right that grants the right to add, change, or remove any values of the property.

See "Rights."

NetWare Glossary

**W**

---

## Index

---

- A**
- Access Control List, explained, 1-2
  - Access Control right, explained, 1-2
  - Access privileges, explained, 1-3
  - Accounting
    - assigning account balances, 1-5
    - charge rate formula, 1-4
    - charging for network services and resources, 1-3
    - explained, 1-3
    - sample charge rate, 1-4
  - ACL. See Access Control List
  - Add Self right, explained, 1-5
  - Add-on board, explained, 1-5
  - Address Resolution Protocol (ARP), explained, 1-6
  - Address. See Address Resolution Protocol (ARP); IPX external network number; station address
  - ADMIN object, explained, 1-6
  - Administration, remote, 1-169
  - Alias object, explained, 1-7
  - Application, explained, 1-7
  - Archive bit. See Modify bit
  - Archive Needed attribute, explained, 1-8
  - Archive, explained, 1-8
  - ARP. See Address Resolution Protocol
  - Attaching, explained, 1-8
  - Attribute
    - execute only, 1-53
    - explained, 1-9, 1-189
    - extended, 1-53
    - hidden, 1-63
    - immediate compress, 1-67
    - indexed, 1-67
    - migrated, 1-97
    - normal, 1-125
    - purge, 1-164
    - read only, 1-166
    - rename inhibit, 1-170
    - system, 1-201
    - transactional, 1-209
- B**
- Backup, explained, 1-14
  - Bindery
    - compared with NetWare Directory database, 1-14
    - compatibility, 1-112
    - emulation. See Bindery services
    - explained, 1-14
  - Bindery object, explained, 1-15
  - Bindery Queue object, explained, 1-15
  - Bindery services
    - explained, 1-15
  - Binding
    - communication protocols to boards and drivers, 1-18
    - explained, 1-18
  - Board. See Network, board
  - Boot
    - files, explained, 1-19
    - remote, 1-169
  - Boot record. See DOS boot record
  - BOOTCONF.SYS file, explained, 1-19
  - Bridge, explained, 1-20
  - Browse right, explained, 1-20
  - Browsing
    - explained, 1-20
    - illustrated, 1-20
  - Buffer, explained, 1-21
- C**
- Cabling system, explained, 1-22, 1-120
  - Cache
    - memory, 1-22
    - read-ahead, 1-166
  - Can't Compress attribute, explained, 1-22
  - CD-ROM, explained, 1-41
  - Character, multiple-byte, explained, 1-97
  - Child VLMS, explained, 1-222
  - Client
    - DOS, 1-42
    - explained, 1-22
    - Windows, 1-227
  - COM port, explained, 1-23
  - Command
    - DOS notation, v
    - syntax, iv
    - typographic conventions, v
  - Command format, explained, 1-23
  - Common name, explained, 1-23
  - Communication
    - explained, 1-23
    - protocol, 1-24
    - protocols, listed, 1-24
  - Compare right, explained, 1-24
  - Compressed attribute, explained, 1-25
  - Compression, file, 1-56
  - Computer object, explained, 1-25
  - Configuration
    - ethernet, 1-49
    - explained, 1-25
    - NET.CFG syntax and notation, vi
  - Configuring
    - NET.CFG syntax and notation, vi
  - Connection
    - number, 1-26
    - remote, 1-169
  - Connection restriction, for managing user objects, 1-219
  - Connectivity, explained, 1-26
  - Console
    - explained, 1-26
    - server, 1-193
  - Container
    - login script, 1-80
-

---

## Index

---

- object, 1-26, 1-130
- Context
  - explained, 1-26, 1-109
  - illustrated, 1-27
  - in a Directory tree, example, 1-27
  - name, 1-100
  - object, 1-137
- Copying, recursive, explained, 1-168
- Country object, explained, 1-28
- Create right, explained, 1-28
- Custom configuration, time servers, 1-208
- D**
- Daemon
  - explained, 1-29
  - NetWare, 1-102
  - NetWare Protocol stack, 1-118
  - NPS, 1-126, 1-195
  - SAP, 1-182
- Daemons, NetWare Directory Service, 1-113
- Data
  - directory, 1-39
- Database, NetWare. See NetWare Directory database
- Default
  - Directory partition, example, 1-111
  - drive, 1-29
  - server, explained, 1-29
- Definition, printer, 1-157
- Delete Inhibit attribute, explained, 1-29
- Delete rights, explained, 1-30
- Delete Self right, explained, 1-5
- Delimiter, explained, 1-30
- Device
  - print, 1-151
  - sharing, 1-30
- Directory, 1-31
  - containing applications, example, 1-38
  - data, 1-39
  - home, explained, 1-39
  - home. See also Home directory
  - management request, 1-31
  - management request, examples, 1-31
  - NetWare, 1-34
  - NetWare, illustrated, 1-34
  - parent, 1-146
  - PUBLIC, 1-163
  - rights, 1-30, 1-173, 1-174
  - root, 1-178
  - structures, illustrated, 1-36
  - SYS:ETC, 1-35
  - SYS:LOGIN, 1-35
  - SYS:MAIL, 1-35
  - SYS:PUBLIC, 1-35
  - SYS:SYSTEM, 1-35
  - username, 1-39
- Directory database compared to file system directory, 1-30
- Directory Map object, explained, 1-31, 1-45
- Directory partition
  - defaults, example, 1-111
  - explained, 1-110
  - replicas, 1-171
- Directory path, 1-35
  - conventions, 1-35
  - conventions, illustrated, 1-35
  - example, 1-35
  - explained, 1-32
- Directory rights
  - described, 1-174
  - explained, 1-187
- Directory Service daemons. See NetWare Directory Service daemons
- Directory Services Object. See Object Directory Services. See NetWare Directory Services
- Directory structure
  - file system, 1-33
  - NetWare Directory Services, 1-40
- Directory tree
  - configuration, 1-107, 1-134
  - container object, 1-135
  - example, 1-40, 1-135
  - explained, 1-40, 1-106
  - illustrated, 1-106
  - name context, 1-100
  - names, example, 1-136
  - object, 1-134
- Directory Trustee. See Trustee
- Directory. See NetWare Directory Services
- Disk
  - explained, 1-41
  - format, 1-41
- Disk partition replicas, explained, 1-112
- Diskette. See Disk; Floppy disk
- Don't Compress attribute, explained, 1-41
- Don't Migrate attribute, explained, 1-41
- DOS
  - boot record, 1-42
  - client, 1-42
  - command notation, v
  - filename, 1-42
  - requester. See NetWare DOS Requester
  - setup routine, 1-43
  - software version number, 1-43
- DOS Requester VLMs, explained, 1-114
- DOS Requester. See NetWare DOS Requester
- DOS workstation brand name, explained, 1-93, 1-196
- Drive
  - default, 1-29
  - explained, 1-43
  - mapping, 1-44
- Dynamic memory, explained, 1-45

---

## Index

---

- E**  
EAs. See extended attribute  
Effective rights  
  explained, 1-47, 1-189  
  illustrated, 1-48  
Engine, explained, 1-49  
Erase rights, explained, 1-49  
Ethernet configuration, 1-49  
Execute Only attribute, explained, 1-53  
Extended attribute, explained, 1-53  
Extension, filename, explained, 1-59  
External network number, explained, 1-73, 1-123
- F**  
Fake root, explained, 1-55  
FAT, explained, 1-55  
File  
  caching, 1-56  
  compression, 1-56  
  explained, 1-35  
  handle. See Handle  
  illustrated, 1-34  
  locking, 1-56  
  NETX, 1-124  
  public, 1-163  
  rights, 1-30, 1-56, 1-174, 1-175  
  salvageable, 1-182  
  scan rights, 1-57  
  sharing, explained, 1-57  
File allocation table, explained, 1-55  
File Scan rights, explained, 1-57  
File system  
  NetWare for UNIX 4.02, 1-57  
  NPFS, 1-57  
  NPFx, 1-57  
  root, 1-178  
  UNIX, 1-59  
File system directory compared to NetWare Directory database, 1-30
- File Transfer Protocol (FTP), explained, 1-59  
Filename  
  DOS, 1-42  
  extension, 1-59  
  long, 1-93  
Finding. See Browsing  
Flag. See Attribute  
Floppy disk, explained, 1-41  
Form  
  explained, 1-60  
  printer, 1-158  
Frame, explained, 1-61  
FTP. See File Transfer Protocol
- G**  
Gateway, explained, 1-62  
Group membership, for managing user objects, 1-217  
Group object, explained, 1-62
- H**  
Handle, explained, 1-63  
Hard disk, explained, 1-41, 1-63  
Hardware port, 1-150  
Hardware Specific Module (HSM), explained, 1-141  
HCSS. See High Capacity Storage System  
Hexadecimal, explained, 1-63  
Hidden attribute, explained, 1-63  
High Capacity Storage System (HCSS), explained, 1-64  
High Performance File System, explained, 1-64  
Home directory  
  example, 1-39  
  for managing user objects, 1-217  
  NetWare, 1-64  
Hop count, explained, 1-64  
HPFS. See High Performance File System
- HSM. See Hardware Specific Module  
Hybrid user, explained, 1-65
- I**  
Identifier variables  
  example, 1-85  
  explained, 1-67  
Immediate Compress attribute, explained, 1-67  
Indexed attribute, explained, 1-67  
Inheritance  
  of trustee assignment, example, 1-68, 1-70  
  related to security, 1-188  
Inherited Rights Filter  
  file system, 1-67  
  illustrated, 1-69  
  NDS object, 1-68  
  related to security, 1-188  
Inherited rights, explained, 1-188  
Inode, NetWare, 1-116  
Internal network number, explained, 1-70, 1-123  
Internet Protocol (IP), explained, 1-70  
Internetwork address, IPX, 1-74  
Internetwork Packet Exchange Open Data-Link Interface. See IPXODI  
Internetwork Packet Exchange. See IPX  
Internetwork, explained, 1-71  
Interoperability, explained, 1-72  
Interprocess communication (IPC), explained, 1-72  
IP. See Internet Protocol  
IPC. See Interprocess communication  
IPX  
  explained, 1-73  
  external network number, 1-73, 1-123  
  internal network number, 1-73, 1-123  
  internetwork address, 1-74

---

## Index

---

- IPXODI, explained, 1-74
- J**
- Jumper block, explained, 1-75
- K**
- Keystroke notation, vi
- L**
- LAN driver, client, explained, 1-76
- LAN. See Local Area Network
- Large Internet Packet (LIP), explained, 1-76
- Leaf object
- examples, 1-132
  - explained, 1-77, 1-131
  - types, 1-132
- Link Support Layer (LSL), explained, 1-77, 1-140
- LIP. See Large Internet Packet
- Loadable module. See NetWare Loadable Module; Virtual Loadable Module
- Local Area Network, explained, 1-78
- Local drive mapping, explained, 1-44
- Lock manager. See Synchronization services
- Locking
- file, 1-56
  - record, 1-167
- Logical drive, explained, 1-43
- Logical memory, explained, 1-78
- Login
- explained, 1-78
  - NetWare 4, 1-185
  - restrictions, 1-79
  - UNIX, 1-190
- LOGIN directory, explained, 1-79
- Login name, for managing user objects, 1-216
- Login script
- commands, example, 1-84
  - container, 1-80
  - container, examples, 1-87
  - example, 1-82
  - explained, 1-80
  - identifier variables, listed, 1-85
  - illustrated, 1-81, 1-82
  - profile, 1-80, 1-160
  - profile, examples, 1-90
  - system, 1-87, 1-201
  - user, 1-80, 1-91, 1-216
- Logout, explained, 1-93
- Long filename, explained, 1-93
- Long machine type, explained, 1-93
- LPT1 port, explained. See Parallel port
- LSL. See Link Support Layer
- M**
- MAIL directory, explained, 1-95
- Managed node, explained, 1-117
- Management agents, NetWare, 1-117
- Map, explained, 1-95
- Media Support Module (MSM), explained, 1-140
- Memory
- allocation, 1-96
  - board, 1-96
  - dynamic, 1-45
  - logical, 1-78
  - management (DOS), 1-95
  - protected mode, 1-162
- Message
- packet, explained, 1-96
  - system, 1-96
- Migrated attribute, explained, 1-97
- MLID. See Multiple Link Interface Driver
- Modify bit, explained, 1-97
- Modify rights, explained, 1-97
- MSM. See Media Support Module
- Multiple Link Interface Driver, explained, 1-97, 1-139
- Multiple name space support, explained, 1-98
- Multiple-byte character, explained, 1-97
- Multiplexors, VLM, explained, 1-222
- Multiserver network, explained, 1-99
- N**
- Name context, explained, 1-100
- Name space, explained, 1-98, 1-100
- NCP packet signature, explained, 1-101
- NCP service protocols, explained, 1-103
- NCP, explained, 1-100
- ncp\_engine, explained, 1-100
- NDS. See NetWare Directory Services
- NEMUX, explained, 1-115
- NET.CFG file
- syntax and notation, vi
- NET.CFG file, explained, 1-102
- NETBIOS.EXE file, explained, 1-101
- NetWare
- daemon, 1-102
  - managed node, 1-117
  - network, 1-120
- NetWare 4 Login, 1-185
- NetWare Core Protocol (NCP), explained, 1-103
- NetWare Directory database
- compared with bindery, 1-14
  - explained, 1-104
- NetWare Directory Services
- default partition, 1-147
  - partition management, 1-148
  - rights. See Rights
- NetWare Directory Services (NDS)
- context, explained, 1-109
  - explained, 1-104
  - object, 1-105
  - object contexts, 1-137
  - object properties, 1-137

---

## Index

---

- properties, 1-105
  - request, explained, 1-33
  - NetWare Directory Services daemon, explained, 1-113
  - NetWare DOS Requester
    - explained, 1-114
    - illustrated, 1-115
  - NetWare inode, explained, 1-116
  - NetWare Loadable Module (NLM), explained, 1-117
  - NetWare management agents, explained, 1-117
  - NetWare operating system, explained, 1-118
  - NetWare Portable File System (NPFS)
    - volumes, 1-58
  - NetWare Protocol stack daemon, explained, 1-118
  - NetWare server
    - default, explained, 1-29
    - object, 1-119
    - print, 1-155
    - value-added, 1-222
  - NetWare volume. See Volume
  - NetWare volumes and UNIX partitions, compared, 1-58
  - Network
    - address, 1-120
    - board, explained, 1-121
    - communication, 1-121
    - explained, 1-120
    - multiserver, 1-99
    - node, explained, 1-123
    - number, 1-123, 1-124
    - printer, 1-124
    - supervisor, 1-124
    - topology (see Topology)
  - Network address restrictions, for managing user objects, 1-219
  - Network backbone
    - explained, 1-120
    - illustrated, 1-121
  - Network drive mapping, explained, 1-44
  - Network File System. See NFS
  - Network management, explained, 1-122
  - Network numbering
    - example, 1-124
    - explained, 1-123
  - Network printing, illustrated, 1-159
  - Network search drive mapping, explained, 1-44
  - NETX file, explained, 1-124
  - NFS, explained, 1-124
  - NIC. See Network board
  - NLM. See NetWare Loadable Module
  - Node
    - address, 1-125
    - managed, 1-117
    - number, 1-125
  - Non-network printing, illustrated, 1-159
  - Normal attribute, explained, 1-125
  - Novell Virtual Terminal. See NVT2
  - NPS daemon, explained, 1-126, 1-195
  - NPS. See NetWare Protocol Stack daemon
  - NSE. See Network Support Encyclopedia Professional Volume
  - Number
    - internal network, explained, 1-70
    - IPX external network, 1-73
    - IPX internal network, 1-73, 1-123
    - node, 1-125
  - Numbering, network, 1-123
  - NVT2, explained, 1-126
  - nwcm, explained, 1-127
  - nwconfig, explained, 1-127
  - NWDIAG, explained, 1-128
  - NWU daemon, and NEMUX, 1-100
  - NWU daemon, and the service engine, 1-100
- O**
- Object
    - common name, 1-135
    - complete name, 1-135
    - context, 1-109
    - Directory tree, 1-134
    - explained, 1-105, 1-129
    - in a Directory tree, example, 1-130
    - leaf, 1-77, 1-131
    - leaf, types, 1-132
    - name, 1-107, 1-108, 1-135, 1-136
    - print server, 1-156
    - printer, 1-158
    - profile, 1-160
    - rights, 1-138
    - root, 1-178
  - Object management, related to international languages. See Unicode
  - Object properties, explained, 1-137, 1-161
  - Object rights
    - described, 1-175
    - example, 1-175
    - explained, 1-175, 1-187
  - ODI architecture, illustrated, 1-139
  - ODI. See Open Data-link Interface
  - ODINSUP
    - explained, 1-138
    - illustrated, 1-142
    - protocol stack, 1-138
  - Open Data-link Interface and Ethernet, 1-52
    - example, 1-139
    - explained, 1-138
  - Operating system
    - NetWare, 1-118
    - UNIX, 1-215
  - Organization object, explained, 1-143
  - Organizational Role object, explained, 1-143
  - Organizational Unit object, explained, 1-143

---

## Index

---

### P

Packet  
  explained, 1-144  
  message, explained, 1-96  
  signature, 1-101

Packet Burst protocol  
  explained, 1-144

Parallel port, explained, 1-146

Parent directory  
  explained, 1-146  
  for applications, example, 1-38

Parent object, explained, 1-130, 1-146

Parity, explained, 1-146

Partition  
  directory, 1-110  
  NetWare Directory Services, 1-147

Password  
  explained, 1-149  
  related to security, 1-185

Password restriction, for managing user objects, 1-219

Path, directory  
  example, 1-35  
  explained, 1-35, 1-149

Permanent drive mapping, explained, 1-44

Permissions, UNIX, 1-149, 1-190

Physical drive, explained, 1-43

Polling time, queue, 1-165

Port  
  parallel, 1-146  
  serial, 1-193  
  software, 1-150

Port, hardware, 1-150

Power conditioning, explained, 1-150

Primary time server  
  explained, 1-151  
  illustrated, 1-205

Primary time server, explained, 1-204

Print  
  device, 1-151  
  mode, 1-152

Print job  
  configuration, explained, 1-153  
  configuration, for managing user objects, 1-218  
  explained, 1-152

Print queue  
  explained, 1-153  
  operator, 1-155  
  polling time, 1-155

Print server  
  explained, 1-155  
  operator, 1-156

Print Server object, explained, 1-156

Printer  
  definition, 1-157  
  explained, 1-156  
  form, 1-158  
  object, 1-158

Printing  
  explained, 1-158  
  network, illustrated, 1-159  
  non-network, illustrated, 1-159

Process, explained, 1-159

Profile login script, explained, 1-80, 1-90, 1-160

Profile object, explained, 1-160

Prompt, explained, 1-160

Property  
  explained, 1-161  
  object, 1-137  
  security equivalence, 1-190

Property rights  
  described, 1-176  
  example, 1-176  
  explained, 1-162, 1-176, 1-187

Protected mode, explained, 1-162

Protocol  
  communication, 1-24  
  communication, listed, 1-24  
  explained, 1-162  
  NCP service, 1-103  
  Packet Burst, 1-144  
  stack, 1-163

Protocol stack, explained, 1-163

PUBLIC directory, explained, 1-163

Public file, explained, 1-163

Public trustee, explained, 1-163

Purge attribute, explained, 1-164

### Q

Queue  
  explained, 1-165  
  polling time, 1-165

### R

RAM, explained, 1-166

Random Access Memory. See RAM

Read Only attribute, explained, 1-166

Read rights, explained, 1-166

Read-ahead cache, explained, 1-166

Real mode, explained, 1-166

Record locking, explained, 1-167

Recovering lost files, explained, 1-182

Recursive copying, explained, 1-168

Reference time server  
  explained, 1-168, 1-205  
  illustrated, 1-206

Registered resources, explained, 1-168

Remote  
  administration, 1-169  
  boot, 1-169  
  connection, 1-169  
  reset, explained, 1-169  
  workstation, 1-170

Rename Inhibit attribute, explained, 1-170

Rename rights, explained, 1-171

Replica  
  explained, 1-112, 1-171  
  synchronization, 1-172

Reset, remote, explained, 1-169

Resources  
  network, 1-172  
  registered, 1-168



---

## Index

---

- Restoring data, explained, 1-173
- Right  
write, 1-227
- Rights  
directory, 1-30, 1-32, 1-173, 1-174  
effective, 1-47  
effective, illustrated, 1-48  
erase, 1-49  
explained, 1-173, 1-186  
file, 1-30, 1-174, 1-175  
filter, 1-188  
inherited, 1-188  
modify, 1-97  
object, 1-138, 1-175  
object, described, 1-175  
property, 1-162, 1-176  
property, described, 1-176  
read, 1-166  
related to security, 1-186  
rename, 1-171  
supervisor, 1-199
- RIP. See Router Information Protocol
- Root  
directory, 1-178  
explained, 1-179  
fake, 1-55  
file system, 1-178  
object, 1-178
- Router  
compared to bridge, 1-20  
explained, 1-179  
illustrated, 1-179  
local versus remote, 1-180  
NetWare, versus traditional bridge, 1-179  
remote versus local, 1-180
- Router Information Protocol (RIP), explained, 1-180
- S**
- Salvageable files, explained, 1-182
- SAP daemon, explained, 1-182
- SAP. See Service Advertising Protocol (SAP)
- Script. See Login script
- Scripts, login, explained, 1-80
- SCSI bus, explained, 1-182
- Search drive  
explained, 1-182  
mapping, 1-44
- Search modes  
example, 1-183  
explained, 1-183
- Secondary time server, explained, 1-184, 1-207
- Security  
attributes, 1-189  
effective rights, 1-189  
explained, 1-184  
inheritance, 1-188  
login, 1-185  
NCP packet signature, 1-101  
rights, 1-186  
trustee, 1-186
- Security equivalence  
explained, 1-217  
for managing user objects, 1-217  
property, 1-190
- Semaphore, explained, 1-192
- Sequenced Packet Exchange. See SPXII.
- Serial port, explained, 1-193
- Server  
console, 1-193  
object. See NetWare Server object protocol, 1-194
- Server protocol. See NetWare daemon
- Service Advertising Protocol (SAP) explained, 1-194  
related to time servers, 1-207
- Service engine and NCP requests, 1-100
- Setup routine, DOS, explained, 1-43
- Shareable attribute, explained, 1-195
- Shared memory, explained, 1-196
- Short machine type, explained, 1-196
- Single reference time server  
explained, 1-196, 1-203  
illustrated, 1-204
- Socket  
explained, 1-197  
Novell-reserved, 1-197
- Software port, explained, 1-150
- SPX. See SPX II
- SPX2. See SPXII
- SPXII, explained, 1-198
- Stack, protocol, 1-163
- Station address, explained, 1-198
- Station, explained, 1-198
- STREAMS, explained, 1-198
- Subdirectory, explained, 1-34, 1-198
- Supervisor right, explained, 1-188
- Supervisor rights, explained, 1-199
- Supervisor, network, explained, 1-124
- Switch block, explained, 1-199
- Synchronization  
replica, 1-172  
time, 1-113, 1-202
- Synchronization services, explained, 1-199
- Synchronization, explained, 1-199
- Syntax. See Command format
- SYS:MAIL directory. See MAIL directory
- SYS:SYSTEM. See SYSTEM directory
- System administrator. See Network, supervisor
- System attribute, explained, 1-201
- SYSTEM directory, explained, 1-201
- System login script  
example, 1-87  
explained, 1-201
- T**
- Tape backup unit, explained, 1-202
-

---

## Index

---

- TCP, explained, 1-209  
TCP/IP, explained, 1-202  
Temporary drive mapping, explained, 1-44  
Terminal emulation software, explained, 1-202  
Time restrictions, for managing user objects, 1-219  
Time server  
  custom configuration, 1-207, 1-208  
  explained, 1-203  
  primary, 1-151, 1-204  
  primary, illustrated, 1-205  
  reference, illustrated, 1-206  
  reference, 1-168, 1-205  
  secondary, 1-184, 1-207  
  single reference, 1-196, 1-203  
  single reference, illustrated, 1-204  
Time synchronization, explained, 1-113, 1-202  
Topology  
  bus network, 1-208  
  explained, 1-208  
  ring network, 1-208  
  star network, 1-208  
Topology Specific Module (TSM), explained, 1-140  
Transaction Tracking System (TTS), explained, 1-209  
Transactional attribute, explained, 1-209  
Transmission Control Protocol. See TCP  
Transmission Control Protocol/Internet Protocol. See TCP/IP  
Trustee  
  database, 1-212  
  explained, 1-209  
  public, 1-163  
  related to security, 1-186  
Trustee right, for managing user objects, 1-217  
Trustee, explained, 1-186  
TSM. See Topology Specific Module  
TTS. See Transaction Tracking System  
**U**  
Unbinding  
  explained, 1-18, 1-213  
  protocols from boards and drivers, 1-19  
  See also Binding  
Unicode  
  code pages, 1-213  
  explained, 1-213  
UNIX  
  client, 1-214  
  file system, 1-59  
  host locking, 1-214  
  login, 1-190  
  operating system, 1-215  
  permissions, 1-149, 1-190  
UNIX partitions and NetWare volumes, compared, 1-58  
Unknown object, explained, 1-213  
User account restrictions, for managing user objects, 1-218  
User login script  
  example, 1-91  
  explained, 1-80, 1-216  
  for managing user objects, 1-218  
User object  
  explained, 1-216  
  managing, 1-216  
User template, explained, 1-220  
Utilities  
  explained, 1-220  
  NLM, 1-221  
  server, 1-221  
  workstation, 1-221  
**V**  
Value-added server, explained, 1-222  
Variables, identifier  
  example, 1-85  
  explained, 1-67  
Virtual Loadable Module (VLM)  
  child, 1-222  
  DOS Requester, 1-114  
  explained, 1-222  
  multiplexors, 1-222  
VLM. See Virtual Loadable Module  
voltab, explained, 1-223  
Volume  
  configuration, 1-224  
  explained, 1-34, 1-223  
  illustrated, 1-34  
  names, 1-223  
Volume object, explained, 1-224  
**W**  
Wait state, explained, 1-226  
WAN. See Wide Area Network  
Watchdog, explained, 1-226  
Wide Area Network (WAN), explained, 1-226  
Workstation  
  default drive, 1-29  
  explained, 1-227  
  remote, 1-170  
  utilities, 1-221  
Write right, explained, 1-227